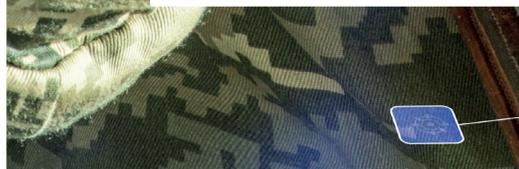
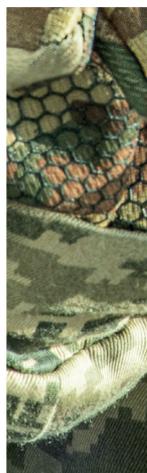
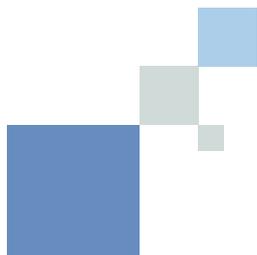


SERIE INGENIERÍA DE SISTEMAS

Cuadernos de Isdefe



1

Introducción a la

INGENIERÍA DE SISTEMAS

en el Siglo XXI



Isdefe

Introducción a la Ingeniería de Sistemas en el Siglo XXI

Volumen 1

Serie Cuadernos de Ingeniería de Sistemas
Cuadernos de Isdefe



AUTORES

Dr. Alejandro Salado / Adolfo Sánchez Domínguez / Dr. Dinesh Verma

Thomas Allen McDermott / Víctor Ramos del Pozo

Dr. Ronald Giachetti / Juan Carlos Larios Monje

David Long / Belinda Misiego Tejada / Dr. Kaitlin Henderson

Christopher Delp / L. Miguel Aparicio Ortega / Dr. Joe Gregory

Dr. Kaitlynn Castelle / Miguel Ángel Coll Matamalas



Título original: Introducción a la ingeniería de sistemas en el Siglo XXI

© Isdefe

C/ Beatriz de Bobadilla, 3 -28040 Madrid

www.isdefe.es

Primera edición: Junio 2024

ISBN: 978-84-09-62257-3

No comercial

Depósito legal: M-14808-2024

Editor: Isdefe

Coordinador: Juan Manuel García Montaña

Coordinador técnico: Alejandro Salado

Equipo de edición y revisión de estilo: Juan Manuel García Montaña, M^a del Rocío Manjón Pérez

Diseño y maquetación: Iliana Aguilar Jiménez

Grabaciones y edición vídeos: Favorit Comunicación

Impreso por Byprint Madrid

Impreso en España – *Printed in Spain*

Las opiniones contenidas en este libro son de exclusiva responsabilidad de los autores firmantes. No pretenden reflejar las opiniones ni el punto de vista de Isdefe como empresa.

El equipo de edición ha hecho todos los esfuerzos posibles para obtener los permisos pertinentes de todo el material reproducido en este libro. Si se hubiera producido alguna omisión, pedimos que nos hagan llegar por escrito la solicitud correspondiente para subsanar el error.

Todos los derechos reservados. Queda prohibida la reproducción total o parcial sin autorización del editor ©.



“[Un sistema es] cualquier porción del universo material que se decide separar del resto del universo con el propósito de considerar y discutir los diversos cambios que pueden ocurrir dentro de él bajo diversas condiciones.”

J.W. Gibbs

PRÓLOGO

Hace alrededor de 30 años, Ingeniería de Sistemas para la Defensa de España S.A. S.M.E. M.P. (Isdefe) publicó los “libros azules” de ingeniería de sistemas. Los “libros azules” eran una serie de dieciséis monografías que captaban el estado del arte de la ingeniería de sistemas de entonces, de una manera práctica y concisa. La serie fue un gran hito por dos motivos principales. En primer lugar, fue probablemente la primera publicación sobre ingeniería de sistemas que se había escrito en español hasta la fecha; y casi con total seguridad, la de mayor envergadura, hasta entonces. En segundo lugar, y lo más importante, la serie introdujo formalmente la ingeniería de sistemas en España en un momento en que, seamos honestos, sólo unos pocos eran siquiera conscientes de lo que la ingeniería de sistemas representaba.



Los “libros azules” cubrían una amplia gama de temas de ingeniería de sistemas, se podría decir que todo el cuerpo de conocimiento de la época: desde la Teoría General de Sistemas hasta el Análisis de Sistemas. La serie fue editada por un comité editorial compuesto por generales ingenieros de las Fuerzas Armadas, funcionarios gubernamentales de España y empleados de Isdefe, y fue escrita por varios expertos nacionales e internacionales en los diferentes temas, entre los que se encontraban el fallecido Ben Blanchard, Donald R. Drew y Jezdimir Knezevic, entre otros. Aunque los “libros azules” fueron una fuente de conocimiento impresionante y única, no tuvieron una distribución generalizada. Las monografías sirvieron como base para introducir tanto a los empleados de Isdefe, como a sus clientes y socios la ingeniería de sistemas, y han permanecido así. (¡Me siento muy afortunado de contar con una copia de la colección completa en mi oficina, a pesar de nunca haber sido empleado de Isdefe!).

Pero el tiempo ha pasado y, aunque no han transcurrido ni 30 años desde entonces, el panorama de la ingeniería de sistemas ha cambiado bastante, tanto en el tamaño de su cuerpo de conocimiento como en el reconocimiento que tiene como disciplina. Algunos ejemplos, sin ser exhaustivos, son:

- El número de miembros activos del International Council on Systems Engineering (INCOSE) (fundado en 1990 y que se internacionalizó en 1995) ha crecido exponencialmente desde los años 2000. La tendencia del número de nuevos miembros que se unen a INCOSE es similar.
- Han surgido muchos métodos nuevos que abarcan todo el ciclo de vida del sistema. La ingeniería de sistemas ya no es únicamente un proceso o una guía para hacer buena ingeniería, sino que los ingenieros de sistemas pueden recurrir a métodos específicos a sus tareas, como la captura de requisitos, la arquitectura de sistemas y demás.
- Hoy en día, disponemos de tecnología dedicada específicamente a ayudar a los ingenieros de sistemas; ya no debemos limitarnos al papel, los procesadores de texto y las hojas de cálculo.
- Hay cada vez más oportunidades formativas para ingenieros de sistemas. Ya no tenemos que limitar nuestro aprendizaje a formarnos en el trabajo, sino que muchas universidades ofrecen programas de posgrado, varias ofrecen programas de doctorado y algunas incluso programas de grado. INCOSE ha establecido un programa formal de certificación y tanto el número de empresas de consultoría que ofrecen formación, como el número de publicaciones de ingeniería de sistemas (artículos, libros, informes, etc.) continúa creciendo cada año.
- La sociedad comienza a entender o al menos reconocer qué es la ingeniería de sistemas. En España, por ejemplo, la ingeniería de sistemas siempre ha estado asociada con la ingeniería de sistemas informáticos. En gran parte, este sigue siendo el caso, pero desde 2014 contamos con un capítulo de INCOSE en España (la Asociación Española de Ingeniería de Sistemas, AEIS), un podcast y algunos eventos que se organizan cada año. Solamente en los últimos 10 años, la cantidad de español que se escucha en una conferencia de ingeniería de sistemas ha aumentado drásticamente. ¡Y qué gusto da!

En este nuevo contexto, Isdefe vuelve a tomar las riendas como líder nacional en ingeniería de sistemas y se dispone a revisar o complementar los “libros azules” con una mirada contemporánea a la ingeniería de sistemas. El propósito de este cuaderno es proporcionar una visión del estado de la práctica de la ingeniería de sistemas, así como de temas que son incipientes en la práctica. El cuaderno está dirigido a profesionales del gobierno y la industria española involucrados en el desarrollo de sistemas de ingeniería, tanto clientes como proveedores, en los sectores de defensa, seguridad, espacio, energía y transporte (con la esperanza de llegar también a una audiencia internacional más amplia).

El cuaderno tiene la forma de colección redactada por capítulos integrados como una única pieza, donde cada capítulo lo ha escrito un experto internacional en el campo junto con un empleado de Isdefe. Cada capítulo sirve como una introducción a un tema que podría convertirse en un cuaderno completo en el futuro para abordar en detalle los temas específicos del mismo. Nuestra intención con este cuaderno es que después de leer un capítulo, el lector tenga una visión general del estado de la práctica y se quede con ganas de aprender más sobre ese tema para implementarlo en su organización.

Ha sido nuestro propósito escribir el cuaderno en un modo divulgativo, tratando al mismo tiempo de ser concisos y mantener rigor técnico. También hemos tratado de evitar en la medida de lo posible abordar el futuro de la ingeniería de sistemas, opiniones sobre el estado de la ingeniería de sistemas o vender ideas, técnicas o métodos que no están respaldados por la práctica o la investigación. Nuestro objetivo ha sido mantener el contenido lo más cercano a la realidad, sin exagerar las capacidades existentes, pero sin tampoco ignorar los avances que estamos viviendo en este momento. Es importante reconocer, sin embargo, que, dada la fluidez con que la ingeniería de sistemas está evolucionando en este momento, es posible que cada organización se encuentre en diferentes momentos de madurez con respecto al material presentado en el cuaderno; incluso me atrevo a decir que algunas ni siquiera han implementado todavía con madurez las prácticas descritas en los antiguos “libros azules”. Dichas organizaciones no deben interpretar el contenido de este cuaderno como una utopía, sino más bien como evidencia de que hay un camino factible para desarrollar y madurar sus capacidades en ingeniería de sistemas.

El cuaderno contiene seis capítulos.

El Capítulo 1 presenta el contexto actual en el que se aplica la ingeniería de sistemas en Europa, así como algunas de las competencias del ingeniero de sistemas del siglo XXI. Los temas principales incluyen la transición de la integración vertical a la especialización, la complejidad de las estructuras contractuales, la alineación de objetivos a lo largo de la cadena de suministro, los equipos internacionales, el doble rol de cliente/contratista y las restricciones de mercado y políticas.

El Capítulo 2 presenta tres aspectos de los sistemas modernos y del futuro para los que las prácticas tradicionales de la ingeniería de sistemas pueden ser ineficaces: sistemas ciberfísicos, sistemas con gobernanza distribuida (sistemas de sistemas) y sistemas basados en el aprendizaje y el trabajo en equipos humano-máquina. El capítulo hace hincapié en las prácticas actuales de dudosa eficacia para este tipo de sistemas y presenta tendencias actuales sobre cómo abordar los aspectos únicos de estos nuevos sistemas.

El Capítulo 3 presenta la necesidad de evolucionar y adaptar los modelos de desarrollo de la ingeniería de sistemas al contexto actual de los proyectos de ingeniería. En él se contrastan los enfoques tradicionales de desarrollo de sistemas dominados por la planificación, como los modelos de cascada y V, y los enfoques de desarrollo ágil que enfatizan la capacidad de respuesta y se caracterizan por ser altamente iterativos e incrementales. La mayoría de los proyectos se pueden beneficiar de ambos tipos de modelos, por lo que los autores discuten enfoques híbridos y cómo adaptar los modelos de desarrollo al contexto de proyectos particulares.

El Capítulo 4 cambia la discusión de sistemas y procesos al área de tecnología para el ingeniero de sistemas. Este capítulo introduce formalmente la ingeniería de sistemas basada en modelos (MBSE) y aborda aspectos como los efectos de formalizar la ingeniería de sistemas, la divergencia y convergencia de semánticas, y la autoría, revisión y control de configuración en entornos basados en modelos.

El Capítulo 5 presenta las nuevas capacidades que los modelos digitales y el alto poder computacional de las estaciones de trabajo actuales proporcionan para apoyar el desarrollo y la integración de sistemas. El capítulo trata temas como la generación y evaluación automatizada de arquitecturas, el diseño basado en conjuntos y la gestión inteligente de la cadena de suministro y el proceso de integración, entre otros.

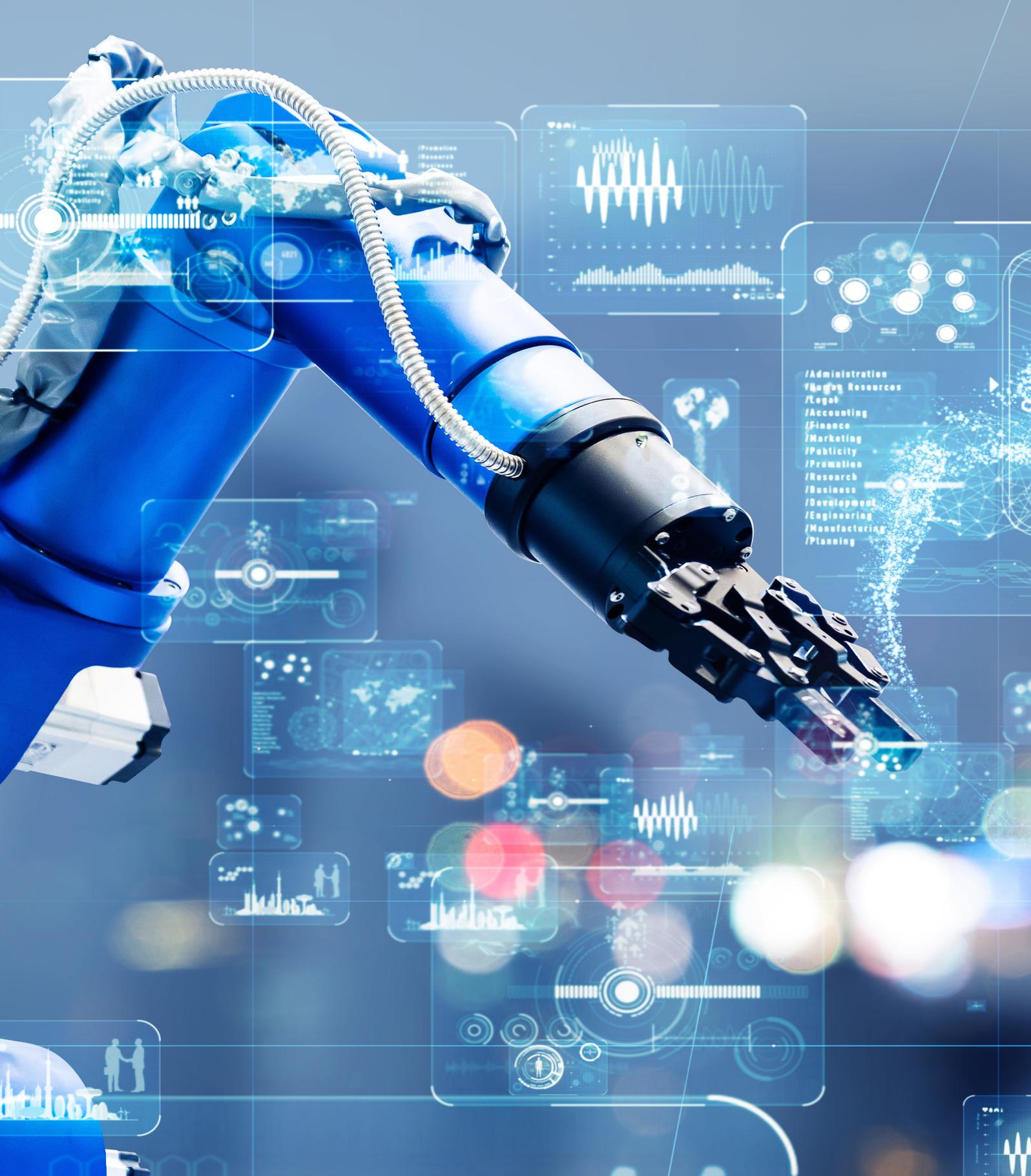
Finalmente, el Capítulo 6 aborda los mismos aspectos que el Capítulo 5 pero aplicados al despliegue, operaciones, mantenimiento y retirada de sistemas. Los temas incluyen el uso de gemelos digitales para realizar mantenimiento predictivo, entornos de realidad virtual para entrenar a los usuarios y el uso de inteligencia artificial para definir estrategias operacionales, entre otros.

Personalmente, ha sido un honor editar este primer cuaderno para la nueva serie de Isdefe y trabajar con un grupo de autores tan implicados. Me siento afortunado de que todos hayan querido embarcarse en esta aventura con nosotros. Estoy muy agradecido a Isdefe y su equipo de gestión de proyectos por confiar y comprometerse conmigo en esta iniciativa, así como por haber mostrado siempre su apoyo.

En nombre de los autores, del equipo de gestión de proyectos y de Isdefe, espero que encuentren la lectura educativa, entretenida y útil.

Dr. Alejandro Salado
Universidad de Arizona





/Promotion
/Research
/Business
/Marketing
/Planning

/Administration
/Human Resources
/Legal
/Accounting
/Finance
/Marketing
/Publicity
/Promotion
/Research
/Business
/Development
/Engineering
/Manufacturing
/Planning

INTRODUCCIÓN A LA INGENIERÍA DE SISTEMAS DEL SIGLO XXI

PRÓLOGO	7
1. LA INGENIERÍA DE SISTEMAS DEL SIGLO XXI	15
1.1. <i>Una perspectiva tradicional de la ingeniería de sistemas</i>	16
1.2. <i>El contexto europeo actual de la ingeniería de sistemas</i>	18
1.3. <i>Ingeniería de sistemas del presente-futuro</i>	24
1.4. <i>Conclusiones</i>	29
2. NUEVOS TIPOS DE SISTEMAS	35
2.1. <i>Introducción</i>	36
2.2. <i>Evaluación de la efectividad de las prácticas tradicionales de ingeniería de sistemas aplicadas a estos nuevos tipos de sistemas</i>	37
2.3. <i>Tendencias en la evolución de la ingeniería de sistemas para su aplicación eficaz a estos tipos de sistemas</i>	45
2.4. <i>Conclusiones</i>	53
3. LA EVOLUCIÓN DE LOS MODELOS DE DESARROLLO Y EJECUCIÓN DE INGENIERÍA DE SISTEMAS	59
3.1. <i>Introducción</i>	60
3.2. <i>Modelos de desarrollo en base a un plan</i>	62
3.3. <i>Modelos de desarrollo evolutivos o ágiles</i>	64
3.4. <i>Comparativa entre desarrollo en base a un plan vs. desarrollo ágil</i>	69
3.5. <i>Adaptación de modelos de desarrollo</i>	71
3.6. <i>Fusión de desarrollo y operaciones con 'DevOps'</i>	73
3.7. <i>Conclusiones</i>	74
4. INGENIERÍA DE SISTEMAS BASADA EN MODELOS	81
4.1. <i>Introducción</i>	82
4.2. <i>Elementos necesarios (e ideales) de MBSE</i>	83
4.3. <i>Modelos, más que simples dibujos</i>	87
4.4. <i>MBSE no es una solución milagrosa: una buena ingeniería de sistemas es un requisito previo para un buen modelado</i>	89
4.5. <i>Nuevas capacidades habilitadas por MBSE</i>	89
4.6. <i>Adopción de MBSE</i>	93
4.7. <i>Conclusiones</i>	96
5. TRANSFORMACIÓN DIGITAL EN EL DESARROLLO DE SISTEMAS	101
5.1. <i>Introducción</i>	102
5.2. <i>Habilitadores de la transformación digital</i>	103
5.3. <i>Gestión del ciclo de vida basado en modelos y el aseguramiento de la calidad</i>	105
5.4. <i>Más allá de la trazabilidad y de la gestión del ciclo de vida</i>	108
5.5. <i>Conclusiones</i>	113
6. TRANSFORMACIÓN DIGITAL EN LA FASE DE SERVICIO DE UN SISTEMA	119
6.1. <i>Introducción</i>	120
6.2. <i>Aplicación de tecnologías de gemelos digitales, entornos virtuales e hilos digitales en fases posteriores del ciclo de vida del sistema</i>	121
6.3. <i>Consideraciones para alcanzar la transformación digital a lo largo del ciclo de vida</i>	127
6.4. <i>Conclusiones</i>	130
EPÍLOGO	135



“Aquí es donde estamos la mayoría de nosotros hoy: formados en algún otro campo hemos aprendido ingeniería de sistemas lo mejor que hemos podido, ingenieros con formación práctica en ingeniería de sistemas.”

A.W. Wymore

Ingeniería de Sistemas en el siglo XXI

Dr. Alejandro Salado, *Universidad de Arizona* (alejandrosalado@arizona.edu)

Adolfo Sánchez Domínguez, *Isdefe* (asdominguez@isdefe.es)

Dr. Dinesh Verma, *Stevens Institute of Technology* (dverma@stevens.edu)

Resumen

Este capítulo presenta el contexto actual en el que se aplica la ingeniería de sistemas en Europa, así como las competencias ideales del ingeniero de sistemas del siglo XXI. Entre los temas principales figuran la transición de la integración vertical a la especialización, la complejidad de las estructuras contractuales, la alineación de objetivos en toda la cadena de suministro, los equipos internacionales, el doble papel de cliente/contratista y las limitaciones políticas y de mercado.

Palabras clave

Historia de la ingeniería de sistemas, evolución de la ingeniería de sistemas, habilidades y competencias.



1. UNA PERSPECTIVA TRADICIONAL DE LA INGENIERÍA DE SISTEMAS

Tradicionalmente, la ingeniería de sistemas se ha concebido como el *pegamento* que conecta las disciplinas de un proyecto de ingeniería, por un lado, y el usuario o cliente, por otro. El ingeniero de sistemas actúa como una especie de coordinador técnico, que asegura que las decisiones tomadas por los ingenieros tradicionales (por ejemplo, ingeniero eléctrico, ingeniero mecánico, etc.) estén bien equilibradas y alineadas hacia un objetivo común, que se define a nivel de sistema. Una analogía común ha sido la del director de orquesta, que se asegura de que las armonías, los tiempos y los volúmenes que toca cada instrumento se sumen adecuadamente para ofrecer una pieza musical (con suerte) hermosa [1]. Sin esa dirección, lo más probable es que la orquesta produjera una cacofonía. La necesidad de equilibrar los deseos de las disciplinas de ingeniería ha sido plasmada por varios autores mediante viñetas como la de la Figura 1, que caricaturiza el diseño de un smartphone según los *deseos* de las distintas disciplinas de ingeniería y el producto resultante una vez que se han equilibrado teniendo en cuenta al cliente.

Aunque podría decirse que la coordinación y la integración técnicas siguen siendo la representación más común de la ingeniería de sistemas, esta visión es incompleta. El ciclo de vida es el otro pilar sobre el que se asienta la perspectiva tradicional de la ingeniería de sistemas. El ingeniero de sistemas también supervisa la evolución del sistema a lo largo de su ciclo de vida (es decir, desde su concepción hasta su retirada) y, de hecho, prevé esas consideraciones para fundamentar las primeras decisiones; el ingeniero de sistemas *piensa en el final antes del comienzo*. Si la coordinación técnica o la integración son importantes para equilibrar los deseos de las disciplinas de ingeniería, ser intencional sobre las consideraciones del ciclo de vida es importante para promover la coherencia entre la necesidad real que debe satisfacerse y el producto final que entra en operaciones. Este paradigma también se ha representado a menudo en viñetas, como la de la Figura 2, al menos desde los años 60.

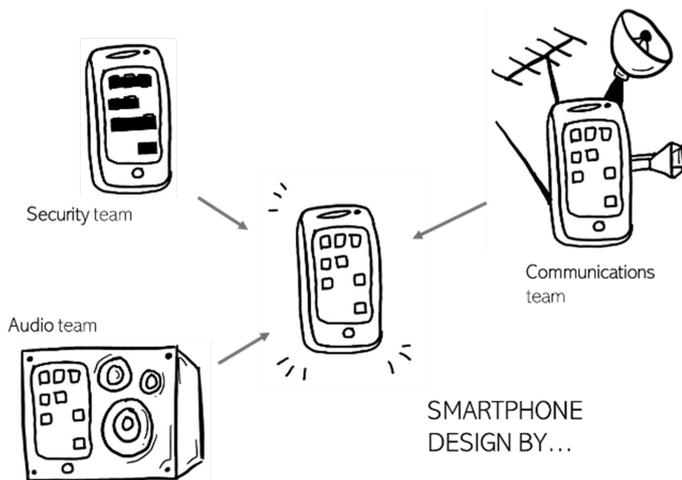


Figura 1.- Equilibrio de silos de ingeniería [2]

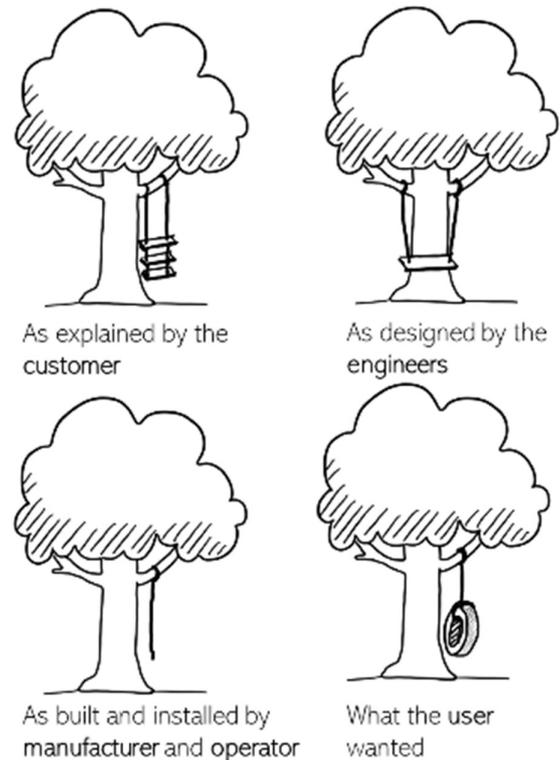


Figura 2.- Integración de las consideraciones relativas al ciclo de vida [2]

Esta perspectiva se basa en varios principios, como el pensamiento global (que a veces se equipara con la mentalidad sistémica), la previsión, la comunicación y la influencia [3]. El pensamiento global es la capacidad de identificar conexiones (o relaciones) entre partes y comprender cómo esas relaciones generan comportamientos emergentes a un nivel superior (normalmente denominado nivel de sistema). Siguiendo el paradigma del coordinador técnico, el pensamiento global se manifiesta cuando un ingeniero comprende, por ejemplo, que el consumo de energía de una parte determinada no es crítico de forma aislada, pero sí tiene efectos sobre el dimensionamiento de la fuente de alimentación, que puede afectar a la capacidad necesaria del sistema de control térmico, que a su vez puede afectar al diseño de la estructura de la carcasa, que puede afectar al rendimiento electromagnético del sistema, etc. Esto se aplica también al paradigma de las consideraciones relativas al ciclo de vida, en el que el pensamiento global se pone de manifiesto cuando un ingeniero comprende, por ejemplo, que una arquitectura determinada puede lograr un rendimiento excelente, pero al coste inasumible de un proceso de integración muy intrincado o de un mantenimiento que es imposible de realizar.

Probablemente no resulte sorprendente que la coordinación técnica en el contexto de la integración de las consideraciones del ciclo de vida se realice con el propósito de promover el éxito en el desarrollo del sistema. Es decir, se quiere evitar la desagradable sorpresa de que el sistema que se ha desarrollado sea inviable de poner en servicio y operar o incluso inadecuado para su propósito; esto es independiente de si esto es el resultado de inconsistencias técnicas o de operaciones inasumibles. La capacidad de prever y anticipar dificultades y problemas durante el desarrollo de un sistema es, por tanto, fundamental para un ingeniero de sistemas. Debido a la gran influencia que las decisiones tempranas pueden tener en el éxito del esfuerzo de desarrollo (tanto en términos de eficiencia del desarrollo como de eficacia de la solución) [4], la previsión y la anticipación permiten dirigir el trabajo de desarrollo desde el principio de forma que se eviten o mitiguen fácilmente las consecuencias de los obstáculos que puedan surgir durante el ciclo de vida del sistema.

La capacidad de comunicarse e influir en los demás es esencial para cualquiera que pretenda coordinar el trabajo de otros ingenieros, así como para cualquiera que trabaje a través de los límites del ciclo de vida. Por ejemplo, un ingeniero de sistemas puede tener que convencer a un ingeniero de antenas de que su *impresionante* antena es innecesaria y que una antena de *peores prestaciones* no sólo es aceptable, sino que es lo que realmente necesitan para obtener una solución viable a nivel de sistema. Del mismo modo, un ingeniero de sistemas puede tener que convencer a un ingeniero térmico de que su modelo térmico no necesita tanta precisión y que basta con una estimación aproximada para un proyecto concreto. Pero la coordinación no sólo se da entre disciplinas de ingeniería. Un ingeniero de sistemas puede tener que explicar a un jefe de proyecto por qué determinadas modificaciones son críticas (¡y hacerlo sin recurrir a la jerga técnica!), así como trasladar a su equipo de ingeniería algunas preocupaciones sobre la gestión del proyecto (¡y hacerlo sin parecer un gestor que sólo recorta el presupuesto!).

Hasta hace muy poco, y probablemente todavía hoy en la mayoría de las organizaciones, estas habilidades se han desarrollado predominantemente a través de la experiencia [3] y han dependido en gran medida del talento [5]. Descrito informalmente como *cicatriz*, el ingeniero de sistemas posee una base de conocimientos adquirida a base de sufrir y aprender de múltiples errores y problemas en varios proyectos. Cada incógnita que un ingeniero de sistemas encuentra a lo largo de su carrera (es decir, sucesos que desconoce y aun así se materializan), incluso antes de convertirse en ingeniero de sistemas, se convierte en una incógnita conocida que se añade a su conjunto personal de reglas o principios (es decir, el ingeniero sabe ahora que el suceso podría ocurrir), adquiriendo la capacidad de preverlos y actuar en consecuencia antes de que vuelvan a ocurrir. Esencialmente, *ha estado allí, lo ha visto, lo ha vivido*. Además, una consecuencia natural de acumular tal experiencia es el respeto o reputación que se genera entre compañeros [6]. Por lo general, esto mejora considerablemente la posición de la persona a la hora de ejercer influencia y persuadir a los demás [7] y suele ir asociado a ascensos profesionales, que ofrecen más oportunidades de participar en diversas formas de comunicación. Además, las personas con curiosidad por ir más allá de su ámbito de especialización pueden ampliar sus conocimientos intercambiando ideas con otras personas en equipos multidisciplinares o interdisciplinares [8].

Por eso, la ingeniería de sistemas se ha concebido tradicionalmente como un paso para algunos ingenieros veteranos en su progresión profesional: aquellos que eran muy buenos en sus respectivos sectores desarrollaban una sólida amplitud de conocimientos en todos los campos que componen el sistema de interés y eran capaces de comunicarse a través de silos (incluidos la dirección, los clientes y otras personas implicadas en diversas áreas del desarrollo del sistema a lo largo del ciclo de vida) y se convertirían en *el* ingeniero de sistemas de un proyecto específico [3]. La ingeniería de sistemas no sería algo que se aprendiese en la universidad, sino algo en lo que uno se transformaría y llegaría a ser.

Dada la falta de fundamentos científicos de la disciplina [9], los procesos se tornaron pronto en un vehículo eficaz para armonizar el trabajo de ingeniería de sistemas [10]. Estos llegaron a ser tan comunes en la práctica de la ingeniería de sistemas que muchos empezaron a referirse a la ingeniería de sistemas como el *proceso de ingeniería de sistemas*, en lugar de considerarla una disciplina. Actualmente, este punto de vista sigue prevaleciendo en varios ámbitos. (¡Basta con buscar el término en Internet para encontrar ese tratamiento por parte de empresas, agencias gubernamentales y universidades en el momento de escribir este capítulo!). Por lo general, este proceso se describe, de forma simplista, como una secuencia descendente (top-down) de actividades que transforman y descomponen las necesidades y los requisitos en un conjunto de componentes que posteriormente se integran para formar un sistema. Este punto de vista refuerza la idea de la ingeniería de sistemas como un paso en la progresión profesional de un ingeniero, ya que *sólo* hay que aprender y aplicar un proceso a un trabajo de ingeniería.

Dada la proliferación de diferentes procesos para apoyar un proyecto de ingeniería de sistemas, como el Desarrollo en Espiral o, más recientemente, las metodologías ágiles, ha habido una tendencia en la comunidad de ingeniería de sistemas a no considerar la ingeniería de sistemas un proceso en sí, sino más bien un enfoque metodológico: el *enfoque sistémico*. De hecho, este es el enfoque del International Council on Systems Engineering (INCOSE), que define la ingeniería de sistemas como “un enfoque transdisciplinario e integrador para permitir la realización, el uso y la retirada con éxito de sistemas de ingeniería, utilizando principios y conceptos de sistemas y métodos científicos, tecnológicos y de gestión” [11]. Aunque el paradigma del *enfoque* abarca sin duda una comprensión más amplia que el paradigma del *proceso*, sigue habiendo una implicación subyacente de que la ingeniería de sistemas no es una disciplina en sí misma, sino algo que se puede aprender transversalmente en la carrera profesional.

Esta concepción de la ingeniería de sistemas ha permanecido prácticamente inalterada desde los tiempos del programa Apolo de la NASA. De hecho, el programa Apolo ha seguido siendo el principal paradigma para la formación de ingenieros en su mayor parte. Sin duda, Apolo es probablemente uno de los proyectos de ingeniería más inspiradores y audaces que ha presenciado la humanidad. (Nosotros creemos que lo es). Por lo tanto, se podría argumentar que, si tal es el caso, ¿por qué no utilizar Apolo como epítome de cómo se debe educar, formar y desarrollar a los ingenieros? La respuesta a esta pregunta puede estar en el hecho de que también es justo reconocer que el contexto en el que se desarrolló dicho programa puede no ser representativo de los contextos a los que se enfrentan la mayoría de los ingenieros hoy en día.

2. EL CONTEXTO EUROPEO ACTUAL DE LA INGENIERÍA DE SISTEMAS

2.1. Complejidad organizacional, política e industrial

Como ya se ha dicho, el programa Apolo fue un trabajo de ingeniería excepcional. Pero también se benefició de ciertas características contextuales que no vemos a menudo, o nunca, en los programas de ingeniería actuales. Dejando a un lado el hecho de que el programa Apolo era una carrera en medio de una guerra (incluso si era la Guerra Fría), veamos cómo se comparaba la financiación de la NASA en aquella época con la situación actual (Figura 3).

Se pueden observar tres características. En primer lugar, se puede ver fácilmente el pico de financiación que recibió la NASA durante los días del Apolo, que es aproximadamente el doble de lo que ha recibido la NASA después de haber aterrizado en la Luna. En segundo lugar, hay que señalar que el presupuesto de la NASA durante el programa Apolo se empleaba principalmente en dicho programa, mientras que el presupuesto actual se reparte entre muchos proyectos en curso y futuros de la NASA. Esta asignación de recursos amplifica aún más la diferencia de financiación entre el programa Apolo y otros programas de la NASA en la actualidad (al menos en lo que se emplea al año). El tercer aspecto, el cuál es probablemente el más importante con respecto al contexto para la práctica de la ingeniería, se lo contó al primer autor del capítulo un compañero que supervisaba a un estudiante de doctorado que investigaba la asignación de fondos gubernamentales durante el programa Apolo. Todas las solicitudes de presupuesto que la NASA hizo al gobierno fueron aprobadas. Eso significa que, mientras que la financiación en los proyectos actuales refleja el presupuesto de la NASA (lo que tienen disponible para gastar), el histograma de los años del Apolo muestra lo que la NASA necesitó para completar el proyecto. Es decir, los ingenieros durante el programa Apolo no tenían limitaciones de costes.

Pero los recursos financieros no son la única diferencia contextual entre entonces y ahora. Las organizaciones suelen adoptar uno de los dos paradigmas principales para abordar el desarrollo de sistemas complejos: la integración vertical o la integración horizontal. En la integración vertical, una sola organización es propietaria de todas las fases del proceso de desarrollo del sistema, que abarcan la concepción, la ingeniería, la fabricación y la producción. Por el contrario, las organizaciones que adoptan la integración horizontal se apoyan en diferentes organizaciones externas, especializadas en distintos aspectos de la labor de ingeniería, lo que les permite aprovechar su experiencia y

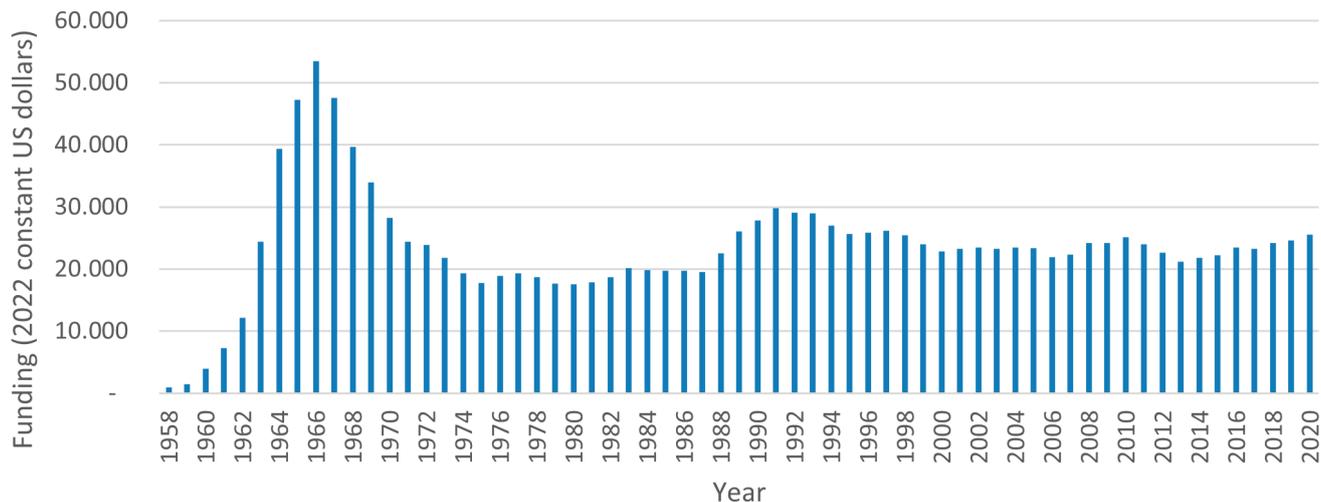


Figura 3. Presupuesto de la NASA a lo largo del tiempo

conocimientos. Por ejemplo, mientras que una organización integrada verticalmente diseñaría y fabricaría *todos* los componentes necesarios para su sistema, una organización integrada horizontalmente compraría un componente determinado (por ejemplo, una batería) a otra empresa especializada en ese tipo de tecnología (por ejemplo, una empresa que sólo diseña y construye baterías). Mientras que la integración vertical ofrece un mayor control sobre el desarrollo del sistema, la integración horizontal promete una mayor eficiencia gracias a la asignación de distintas tareas a empresas expertas en ellas. No obstante, cabe señalar que las integraciones vertical y horizontal no deben entenderse como dicotómicas, sino como una tendencia hacia una mayor integración vertical o una mayor integración horizontal.

La integración vertical era habitual en los primeros tiempos de la ingeniería de sistemas, probablemente porque las empresas existentes se limitaban a aumentar la complejidad y escala de los sistemas que desarrollaban. Sin embargo, más tarde se favoreció la integración horizontal con la esperanza de una mayor eficiencia financiera y un menor riesgo que prometía la especialización, y se convirtió en la norma para las empresas de ingeniería a gran escala. De hecho, la organización horizontal sigue siendo el enfoque más extendido hoy en día, sobre todo en Europa. Sin embargo, las organizaciones han empezado a aceptar que la especialización en estos ámbitos (es decir, sistemas a gran escala como los de los sectores de defensa y espacial) no ha cumplido sus expectativas, al menos de la forma en que se ha aplicado.

La integración horizontal introdujo la necesidad de estructuras contractuales para regir las relaciones entre las distintas organizaciones que participan en el desarrollo de un sistema (aquí surgen los papeles de contratista principal, subcontratista, etc.). La carga impuesta por las obligaciones

contractuales suele acarrear costes importantes que, en muchos casos, han puesto en peligro cualquier posible beneficio obtenido a través de la especialización¹. Además, se trata de un aspecto que, si bien se tiene en cuenta desde un punto de vista filosófico o conceptual en la ingeniería de sistemas, no está integrado en la práctica de la ingeniería de sistemas de una forma que pueda hacerse operativo para navegar de forma eficaz o combinar los aspectos contractuales y de ingeniería de un proyecto.

En Europa, esta situación se ve agravada por las limitaciones geopolíticas derivadas de la necesidad de que varios países no sólo trabajen juntos, sino que compitan y capten financiación europea. En los proyectos europeos no es infrecuente que la financiación que pueda recibir una organización para llevar a cabo una empresa de ingeniería sea proporcional o esté vinculada a la contribución financiera que el país haya hecho al proyecto a través de los canales financieros y políticos europeos. Esto se puede deber al deseo de un país de controlar la dirección de un proyecto en particular, de mantener el control o liderazgo en una tecnología en particular, o de desarrollar capacidades o tecnologías de que el país no dispone en el momento, entre otros. Para competir en este contexto, muchas empresas multinacionales replican capacidades en distintos países en lugar de aplicar plenamente la especialización, lo que da lugar a una competencia interna que se confunde con las limitaciones políticas y la pugna con otras organizaciones externas.

Hoy en día, algunas organizaciones intentan volver a la integración vertical, con la esperanza de integrar más fácilmente a sus equipos de ingenieros en busca de un

¹. Aunque los contratos existían anteriormente, el hecho de que los presupuestos fueran prácticamente ilimitados hacía que la contratación fuese sólo un vehículo para comunicar el trabajo y no una fuente de riesgos que afectara al desarrollo del proyecto.

objetivo común. SpaceX ha mostrado con sus cohetes un ejemplo notable de cómo la integración vertical en el siglo XXI ha contribuido (¡aunque no sea el único factor contribuyente!) a reducir significativamente el coste de un sistema a gran escala con respecto a la integración horizontal.

Sin embargo, las limitaciones geopolíticas persisten en el contexto europeo. Mientras que en Estados Unidos la unión política y cultural con un gobierno centralizado facilita la ejecución de programas de ingeniería de gran envergadura y alto contenido tecnológico, en el contexto europeo la fragmentación en múltiples Estados con intereses políticos y económicos diversos dificulta abordar grandes programas de ingeniería que, por su envergadura o complejidad, sería imposible acometer en solitario. Las organizaciones supranacionales (como la Agencia Espacial Europea) o los programas internacionales (como Eurofighter, A400-M o el más reciente FCAS (Future Combat Air System)) intentan paliar los efectos de esta fragmentación garantizando compromisos de financiación estables y prolongados por parte de los países miembros. (En el recuadro se presentan algunas notas históricas y ejemplos de organizaciones supranacionales en Europa dedicadas al desarrollo de sistemas de ingeniería a gran escala). Sin embargo, estas organizaciones supranacionales no ocultan los problemas inherentes a los intereses particulares de cada país miembro a la hora de repartir las cargas de trabajo, asumir responsabilidades y proteger sus industrias nacionales y sus intereses de seguridad.

Un claro ejemplo de estas dificultades lo encontramos en la puesta en marcha del programa FCAS, con Alemania, Francia y España como principales miembros contribuyentes, que tuvo que enfrentarse, por ejemplo, a la priorización de los intereses de parte de las industrias nacionales sobre las necesidades de los usuarios finales, como el hecho de que Francia (liderada por Dassault) no estuviera a favor de que Airbus fuera el coordinador nacional de los otros dos países miembros, las dudas de Alemania sobre la viabilidad del proyecto, o la apuesta española por Indra como coordinador nacional en detrimento de la más internacional Airbus, todo ello provocando en cualquier caso múltiples retrasos y desacuerdos entre los miembros y socios industriales. Es evidente que la ejecución fragmentada de este tipo de programas de ingeniería incurre en enormes costes de transacción relacionados con la negociación y la toma de decisiones entre sus miembros. Algunas de las consecuencias son la necesidad de ofrecer contrapartidas a cada uno de los socios, el diferente peso de cada uno de ellos dentro de los programas, la ubicación de la producción en plantas que a veces no son óptimas o la necesidad de transportar piezas entre ellas cuando todo el montaje podría realizarse en una única ubicación. Al final, todo ello contribuye a elevar los precios para el cliente final, a aumentar excesivamente los plazos de desarrollo y, en la mayoría de los casos, a provocar retrasos [12].

Algunos ejemplos de organizaciones supranacionales dedicadas al desarrollo de sistemas de ingeniería a gran escala en Europa

En Europa, el tratado constitutivo de la Comunidad Económica Europea (Roma, 1957) e incluso el Tratado de Ámsterdam de 1997 excluyeron el sector armamentístico y de defensa del ámbito comunitario, lo que dificultó la colaboración de los gobiernos y la adopción de un marco común de desarrollo como el que podría existir en Estados Unidos. Uno de los primeros intentos de fomentar la cooperación europea en asuntos de defensa surgió con la creación del Grupo de Programas Europeos Independientes (IEPG) en 1976. Esta iniciativa se concibió como una asociación técnica en consonancia con los principios de la Alianza Atlántica, haciendo hincapié en la preservación de las responsabilidades nacionales de cada país miembro. Las resoluciones elaboradas por el IEPG, aunque carecían de autoridad vinculante, servían principalmente como mecanismo de intercambio de información sobre los procedimientos nacionales de adquisición de armamento y material. Además, facilitaban la investigación y evaluación de posibles marcos para supervisar proyectos conjuntos.

En 1992, el IEPG se convirtió en el Grupo de Armamento de Europa Occidental (GAEO) y se integró en la Unión Europea Occidental (UEO) como organismo responsable de la cooperación en materia de armamento. A pesar de continuar con los procedimientos y relaciones del IEPG, estableció una serie de nuevos objetivos como la búsqueda de la competencia entre los diferentes mercados nacionales, el refuerzo de la base tecnológica y la cooperación en I+D de defensa. En 1996 se creó la Organización de Armamento de Europa Occidental (OAE), un nuevo órgano subsidiario de la UEO dedicado esencialmente a gestionar las actividades de investigación y tecnología.

Dada la escasa repercusión de estas iniciativas, un grupo de países comenzó a avanzar en este campo a través de diferentes acuerdos hasta que en 1996 se constituyó la Organización Conjunta de Cooperación en Materia de Armamento (OCCAR), principal organización en el ámbito de la cooperación industrial en armamento y embrión de la Agencia Europea de Defensa (EDA). La cartera de programas de la OCCAR incluye actualmente 17 importantes programas de armamento con un presupuesto operativo total en 2023 de unos 6.000 millones de euros: A400M, BOXER, COBRA, ESSOR, FREMM, FSAF-PAAMS, LSS, LWT, MALE RPAS, MAST-F, MMCM, MUSIS, NVC, PPA, REACT, TIGER y U212 NFS. La gobernanza de OCCAR y la gestión de los programas de OCCAR se basan en las normas de OCCAR.

Mientras tanto, en 1985 se creó en España la empresa estatal Isdefe (Ingeniería de Sistemas para la Defensa de España). Su fin último era servir de recurso de apoyo técnico en proyectos complejos, con especial atención a los sectores de defensa, aeronáutica y tecnologías de la información y las comunicaciones. Además, se encomendó a Isdefe la tarea de ofrecer asistencia técnica al Ministerio de Defensa en las tareas de ingeniería de sistemas, particularmente relacionadas con los programas de modernización de las Fuerzas Armadas. Desde su creación, Isdefe ha venido aplicando activamente metodologías de ingeniería de sistemas en los procesos de desarrollo de diversos sistemas, incluidos sistemas de mando y control, control del tráfico aéreo, reingeniería de plataformas, optimización de la cadena logística, sistemas de inteligencia y guerra electrónica, así como sistemas de vigilancia y control de fronteras. En particular, Isdefe ha desempeñado un papel crucial en numerosos programas militares, como las fragatas F-110, el vehículo blindado de ruedas VCR 8x8, el avión de transporte militar A400M, el helicóptero Tigre, el submarino S-80 y muchos otros.

En 2004 se fundó la Agencia Europea de Defensa (AED) para ayudar a sus Estados miembros (todos países de la UE) a desarrollar sus recursos militares poniendo en común los intereses nacionales y catalizando los aspectos operativos, tecnológicos e industriales necesarios para poner en marcha programas multinacionales de sistemas de armas, delegando la gestión real de los programas en sus fases de desarrollo y producción en organizaciones como la OCCAR.

En el ámbito aeroespacial, dos iniciativas fijaron el marco europeo de colaboración en ingeniería de sistemas: la creación en 1975 de la Agencia Espacial Europea (ESA) y el establecimiento en 2000 de la empresa aeroespacial European Aeronautic Defence and Space (EADS), rebautizada en 2014 como Airbus Group.

La creación de la Agencia Espacial Europea (ESA), al igual que la de EDA, estuvo precedida por la de otros organismos como la Organización Europea de Investigación Espacial (ESRO) en 1962, la Organización Europea de Desarrollo de Transbordadores (ELDO). Derivado del programa de inversiones de la ESRO o el Centro Europeo de Investigación y Tecnología Espaciales (ESTEC), que se encargaría del desarrollo de satélites y vehículos espaciales; y el Centro Europeo de Operaciones Espaciales (ESOC), responsable del control de las operaciones de los satélites. En 1973, con el acuerdo global de todos los países miembros, se aprobaron tres proyectos (Spacelab, el Programa Ariane y Marots) y se tomó una decisión fundamental: la creación de la Agencia Espacial Europea.

La Comisión Europea ha intentado superar los problemas de fragmentación estableciendo las siguientes iniciativas:

- La Dirección General de Industria de Defensa y Espacio (DEFIS), que dirige las actividades de la Comisión Europea en estos sectores. En el ámbito de la industria de defensa, DEFIS es responsable de mantener la competitividad y la innovación de la industria de defensa europea, garantizando la evolución de una base tecnológica e industrial de defensa europea capaz. En el ámbito espacial, DEFIS es responsable de la aplicación del Programa Espacial de la UE, que consta del Programa Europeo de Observación de la Tierra (Copernicus), el Sistema Europeo de Navegación Global por Satélite (Galileo) y el Sistema Europeo de Navegación por Complemento Geoestacionario (EGNOS).
- El Plan de Acción Europeo de Defensa (PAED), de noviembre de 2016, que busca promover una Base Tecnológica e Industrial de Defensa Europea (BTID) fuerte y competitiva basada en el Fondo Europeo de Defensa (FED), principal marco de financiación del PAED. Este plan se basa en cuatro pilares a través de los cuales se implementan acciones y programas específicos en los ámbitos de la investigación y el desarrollo, la habilitación de cadenas europeas de suministro de defensa y la construcción de un mercado único europeo de defensa.
- La Cooperación Estructurada Permanente (PESCO), de diciembre de 2017, que integra a 26 países y cuyos objetivos clave son la mejora de las capacidades de defensa, la cooperación en operaciones militares y el desarrollo de capacidades militares conjuntas. Su fin último es reforzar la autonomía estratégica de la Unión en materia de defensa.

Estas iniciativas se suman al conjunto cada vez mayor de normativas impuestas por gobiernos y agencias federales que deben cumplir las organizaciones y los sistemas de ingeniería. Los organismos públicos europeos se enfrentan con frecuencia a procedimientos de licitación rigurosos y muy regulados. Aunque estos procedimientos pretenden garantizar la transparencia y la igualdad de oportunidades para los licitadores, a veces pueden resultar lentos y complejos. Esta complejidad puede provocar retrasos en la selección y contratación de organizaciones de ingeniería. Por otra parte, la naturaleza holística de la ingeniería de sistemas exige licitar contratos de diversos tipos (suministros, licencias de software, consultoría, obras, etc.) en los que participan múltiples proveedores y subcontratistas, lo que se suma al reto de la gestión y las relaciones contractuales entre las partes interesadas. Además, en el caso de los grandes consorcios transnacionales, hay que armonizar las normativas específicas de cada país en materia de contratación pública, protección de datos y propiedad intelectual.

En el contexto de España, por ejemplo, el artículo 99.1 de la Ley de Contratos del Sector Público establece que “el objeto de los contratos del sector público deberá ser determinado. El mismo podrá definirse en atención a las necesidades o funcionalidades concretas que se pretendan satisfacer, sin cerrar el objeto del contrato a una única solución.” Este requisito obliga a las organizaciones a definir claramente las necesidades funcionales de los sistemas contratados desde el principio, impidiendo que el proyecto aproveche el conocimiento de las empresas durante sus primeras fases y, en algunos casos, la adopción de modelos, procesos o enfoques de desarrollo innovadores, como la Ingeniería de Sistemas Basada en Modelos (MBSE), si éstos no son exigidos por el cliente.

El tiempo dirá si estas iniciativas tienen éxito, pero las experiencias pasadas no parecen correlacionar la adición de más órganos de gobierno y normativas con el éxito en el desarrollo de sistemas. Ya en 1969, durante el programa Apolo, se expresaron preocupaciones por la excesiva dependencia de los procesos: “Si trazo un gráfico en función del tiempo de lo que parece ser una reciente marea creciente de costes, sobrecostes, rendimiento insatisfactorio e infelicidad entre los ingenieros, tengo motivos para preocuparme. [...] Si trazo en el mismo gráfico en función del tiempo el aumento de conversaciones, directivas... veo una alta correlación entre ambos gráficos” [10], y fueron reafirmados en 2010 por el ex administrador de la NASA Mike Griffin [5]. En términos simplistas, cada nueva normativa añade una nueva restricción que la solución debe satisfacer. Y sabemos que cada restricción reduce el espacio de soluciones, y que una reducción del espacio de soluciones generalmente reduce la asequibilidad del sistema desarrollado [13].

La colaboración internacional, esencial para los grandes programas europeos, como se ha descrito, presenta otros retos que van más allá de los introducidos formalmente por las limitaciones geopolíticas. A continuación, se enumeran algunos ejemplos:

- La comunicación eficaz entre equipos multiculturales y multilingües no es sencilla. Los problemas de comunicación derivan no sólo de las barreras lingüísticas, sino también de los matices culturales en los estilos de comunicación. Por ejemplo, lo que puede considerarse una comunicación directa y clara en un entorno cultural, puede considerarse irrespetuosa o conflictiva en otro. Al mismo tiempo, los equipos multilingües pueden tener dificultades para transmitir información técnica con precisión y de forma que la entiendan todos los miembros, debido a las pérdidas de información en la traducción, así como a la disminución del rendimiento cognitivo por el aumento de la demanda de carga cognitiva al hablar una segunda lengua.

- La coordinación de actividades en las que participa personal que trabaja en distintos lugares y utiliza diferentes herramientas y tecnologías en su entorno de trabajo local puede provocar retrasos en el intercambio de información y en la coordinación de actividades. Además, existen problemas legales y de seguridad para las organizaciones que forman parte de grandes consorcios, ya que el intercambio transfronterizo de datos puede estar sujeto a restricciones, lo que puede añadir complejidad a los esfuerzos de intercambio de información y colaboración.
- Puede ser necesario superar diferencias culturales y de hábitos de trabajo para alcanzar el objetivo final de un sistema conjunto que cumpla las expectativas. La forma de tomar decisiones puede variar mucho de una cultura a otra. Algunas culturas pueden preferir un enfoque basado en el consenso, mientras que otras se basan en la toma de decisiones jerárquica. Estas diferencias pueden afectar a la eficacia de los procesos de toma de decisiones en un proyecto de colaboración.

Estas cuestiones no deben interpretarse en el sentido de que no deba buscarse la colaboración internacional. La colaboración internacional no sólo es necesaria para desarrollar algunos sistemas a gran escala, como ya se ha explicado, sino que también es preferible en muchas situaciones por las diversas ventajas que puede reportar, como el aprovechamiento de la experiencia y los conocimientos. Por ejemplo, en el caso del programa español del submarino S-80, el Ministerio de Defensa español contrató a la US Navy y a General Dynamics-Electric Boat, lo que le permitió replantearse el programa e implantar mejores procesos y metodologías de ingeniería, incluida la adopción del Manual de Ingeniería de Sistemas de la NASA. Lo que se quiere señalar en esta sección es que esto establece un nuevo contexto en el que debe aplicarse la ingeniería de sistemas y para el que puede ser necesario evolucionar los métodos o enfoques.

2.2. Ingeniería y complejidad tecnológica: nuevos sistemas, nuevos métodos

Asimismo, en los últimos años hemos visto, y continuamos viendo, la aparición de nuevos tipos de sistemas que presentan diferencias fundamentales con respecto a aquellos con los que nació la ingeniería de sistemas. Los sistemas tradicionales se caracterizaban por estar basados predominantemente en hardware (el software no era más que una pequeña parte aislada en los casos en que había software), ser monolíticos (sus capacidades no dependían de las capacidades de otros sistemas), desarrollarse a menudo en un campo verde (sin dependencia de sistemas heredados) y carecer de autonomía para las decisiones (alta previsibilidad en el comportamiento de mando y control

del sistema). Los sistemas tradicionales son cada vez menos comunes, si es que existen. Por ejemplo, cuando se desarrollaron los primeros aviones, todos los sistemas de apoyo también tenían que desarrollarse para apoyar al avión (de ahí su nombre). Hoy en día, los nuevos aviones se desarrollan dentro de las limitaciones de los sistemas de apoyo existentes; los sistemas de apoyo ya no desempeñan un papel de apoyo, sino de habilitación. En otras palabras, el desarrollo de los sistemas actuales se ve muy limitado por su necesidad de interactuar con sistemas ya existentes (es decir, sistemas heredados). Aun así, las novedades más importantes proceden del aumento de la importancia y el tamaño de los componentes de software en los sistemas contemporáneos, la transición hacia estructuras de gobierno distribuidas y la aparición de la autonomía en las decisiones o acciones que toma el sistema.

En el pasado, el software se limitaba a algunas funcionalidades muy específicas que, de otro modo, resultaban demasiado difíciles de implementar en hardware. El software se concebía como último recurso, no como solución preferente. Los códigos podían revisarse y probarse por completo. Hoy en día, el software conduce la mayor parte de la funcionalidad de los sistemas existentes o, al menos, la mayoría de las funcionalidades dependen del software. No hay más que ver cómo han evolucionado los coches: ¡hasta el freno de mano se controla por software! El aumento de la dependencia del software, junto con el aumento de su complejidad, pone en tela de juicio muchos de los supuestos bajo los que tradicionalmente se ha practicado la ingeniería de sistemas. El software puede evolucionar rápidamente, puede ponerse en servicio de forma incremental, introduce vulnerabilidades de seguridad de una diversidad y gravedad potencial sin precedentes, es difícil, si no imposible, de probar exhaustivamente, y puede desplegarse y actualizarse sobre la marcha durante las operaciones, entre otras cosas.

Los sistemas tradicionales eran monolíticos en el sentido de que por sí solos podían ofrecer las prestaciones previstas, suponiendo que existieran sistemas de apoyo. Por ejemplo, un televisor funcionaba bien siempre que se pudiera conectar a la red eléctrica y sintonizar su antena. Hoy vivimos la proliferación de los sistemas de sistemas (SoS): aquellos que están fuertemente interconectados y dependen de las capacidades de otros sistemas independientes cuyo objetivo principal no es servir de apoyo. En otras palabras, sistemas que tienen cada uno su propio propósito se organizan de algún modo para producir capacidades imprevistas. En el ejemplo de la televisión anterior, la finalidad de la red eléctrica es suministrar energía a los aparatos domésticos y la de los proveedores de contenidos es ofrecer contenidos a los consumidores. Por tanto, en ambos casos la finalidad de los sistemas es servir al televisor. Sin embargo, un smartphone ofrece a cualquier individuo la posibilidad de navegación personal

aprovechando las señales proporcionadas por el Sistema de Posicionamiento Global (GPS), aunque el propósito principal del GPS sea guiar misiles, no ayudarle a encontrar el camino a algún lugar. Reconocer que no todo conjunto de sistemas es necesariamente un SoS es esencial para no caer en la trampa de llamar y tratar todo como un SoS, ya que a menudo se abusa del término convirtiéndolo en una palabra de moda en vez de un concepto específico. Diferenciar ambos es importante porque las prácticas tradicionales de ingeniería de sistemas son probablemente ineficaces y a veces incluso inviables para abordar los aspectos únicos de los SoS [14]. De hecho, hay incluso atributos del sistema o métricas de rendimiento, como la disponibilidad, que ni siquiera sabemos cómo calcular o predecir para un SoS [15]. Ciertamente, no se puede simplemente utilizar la ingeniería de sistemas tradicional para diseñar o integrar un SoS y esperar tener éxito. La aplicación de la ingeniería de sistemas debe adaptarse (¡no hacerse a medida!) pero, sinceramente, la comunidad de ingeniería de sistemas apenas ha empezado a comprender cómo hacerlo. Como conjunto de prácticas, estamos seguros de que la independencia de gestión y gobierno de los sistemas constituyentes que forman el SoS parecen ser los factores latentes que deben informar la evolución. Como indica la norma ISO para aplicar la ingeniería de sistemas a los SoS, los enfoques basados en supuestos de mando y control, disponibilidad de información (por ejemplo, para apoyar la verificación), existencia de requisitos o servicios garantizados pueden resultar ineficaces [14]. Algunos han llegado a afirmar que, en realidad, los SoS no pueden diseñarse mediante ingeniería, sino sólo integrarse [16], lo que hace inaplicables los procesos relacionados con los requisitos y la arquitectura. En su lugar, se están desarrollando nuevos métodos de ingeniería de sistemas basados en la persuasión y la influencia, el uso de incentivos, las federaciones oportunistas o el desarrollo y la operación combinados.

Si la gobernanza distribuida y la complejidad de una dependencia cada vez mayor del software no fueran suficientes cambios en los sistemas con los que debemos trabajar, hemos empezado a ser testigos de la incorporación de la autonomía en la toma de decisiones y acciones en los sistemas ciberfísicos, principalmente a través de la inteligencia artificial. El exhibir autonomía en las acciones que un sistema toma cambia drásticamente varios supuestos básicos que son fundamentales para la práctica de la ingeniería de sistemas. Por ejemplo, mientras que se considera que los sistemas tradicionales exhiben un comportamiento invariable, en el sentido de que el comportamiento se establece una vez construido el sistema, los sistemas inteligentes se diseñan para cambiar su comportamiento según sea necesario.

Esto pone en entredicho la eficacia de muchas prácticas tradicionales de ingeniería de sistemas, que deben

evolucionar. Por ejemplo, y ciertamente no de forma exhaustiva:

- Dado que un sistema inteligente puede cambiar su comportamiento entre el entorno de pruebas y el entorno operativo, las pruebas en el entorno de pruebas pueden dejar de ser buenas aproximaciones para predecir el comportamiento del sistema durante el funcionamiento en el entorno operativo [17].
- Si se espera que un sistema inteligente aprenda de sus *experiencias*, cada instancia de un sistema podría mostrar una variabilidad incontrolada de su comportamiento con respecto a su clase o familia. Como resultado, las líneas de productos pueden dejar de ser capaces de delimitar el comportamiento y funcionamiento de una clase de sistemas [18].
- Los sistemas inteligentes tienen una funcionalidad subyacente o de apoyo estable (es decir, optimizar una función de recompensa) y el principal impulsor del rendimiento y el comportamiento del sistema son los datos de entrenamiento. La descomposición funcional ya no es capaz de capturar *cómo* funciona el sistema, puesto que son los datos los que definen el comportamiento. ¿Cómo afecta esto a los análisis basados en funciones, como el análisis del árbol de fallos o el Análisis Crítico de Modos y Efectos de Fallo (FMECA)?

Aunque algunos de estos problemas pueden ser relevantes para los sistemas inteligentes basados en software (como una función que predice el comportamiento de compra de un usuario en una tienda en línea), sus soluciones pueden no ser directamente transferibles a los sistemas generales, debido a los efectos altamente acoplados del mundo *físico*. Por tanto, es necesaria una nueva evolución de la ingeniería de sistemas.

3. INGENIERÍA DE SISTEMAS DEL PRESENTE-FUTURO

Los paradigmas y métodos tradicionales de la ingeniería de sistemas se están volviendo ineficaces para hacer frente a los nuevos contextos -organizativos y tecnológicos- en los que se llevan a cabo las tareas de ingeniería. El papel de un único ingeniero sénior que intenta coordinar los esfuerzos técnicos gestionando la información en innumerables documentos, controlando manualmente los elementos de configuración, basándose en bocetos de estilo libre para modelar diferentes facetas del sistema e intuyendo la mayoría de las decisiones está desapareciendo poco a poco. Los métodos basados en la ciencia, la toma de

decisiones formal, el modelado formal, la digitalización de los artefactos de ingeniería y una diversidad de competencias de ingeniería de sistemas que pueden distribuirse dentro de un equipo están configurando el presente y el futuro de la ingeniería de sistemas.

En las siguientes secciones se ofrece una visión general de algunos aspectos relevantes para el contexto contemporáneo y en evolución de la ingeniería de sistemas. Nótese que estas secciones no pretenden ofrecer una visión del futuro de la ingeniería de sistemas ni una hoja de ruta para su evolución; algunas propuestas pueden encontrarse en otras publicaciones (por ejemplo, [19]).

3.1. Teorías y fundamentos emergentes

Los orígenes de la ingeniería de sistemas y su desarrollo como práctica de ingeniería en la década de 1960 estuvieron acompañados de varios esfuerzos por formalizar la disciplina. Pioneros como Wymore [20], Warfield [21], o Mesarovic [22], entre otros, intentaron desarrollar fundamentos matemáticos para apoyar la práctica de la ingeniería de sistemas, que, como ya se ha dicho, se basaba principalmente en la intuición, el talento y, finalmente, los procesos. Sin embargo, es justo afirmar que su trabajo no consiguió traspasar el ámbito académico para poner al corriente la práctica de la ingeniería de sistemas en aquel momento. Durante varios años, su trabajo no sólo no se utilizó, sino que el propio interés por descubrir los fundamentos de la ingeniería de sistemas se desvaneció frente a una investigación más aplicada y útil que se centraba en el desarrollo de métodos.

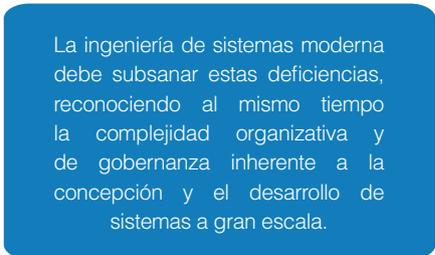
Hoy en día se reconoce cada vez más, tanto en el mundo académico como en la industria, la necesidad de fundamentos científicos que pongan al corriente la práctica de la ingeniería de sistemas. La ciencia es un elemento esencial de la ingeniería. En ausencia de principios científicos, probablemente hablaríamos más de artesanía que de ingeniería. Ciertamente, los seres humanos construyeron puentes y fabricaron productos mucho antes de que Newton formulara las primeras leyes del movimiento. La gente recurría a su intuición, reglas y experiencia y tenía éxito con ellas; ¡hasta aquí nos han hecho llegar, escribiendo el capítulo de un libro en un ordenador! Pero no diríamos que hacían ingeniería. La ingeniería de un puente o un producto es una hazaña totalmente distinta, ya que la ciencia subyacente nos permite no sólo entender cómo funcionan las cosas, sino también cómo utilizar ese conocimiento para predecir mejor los resultados de nuestras decisiones y mejorar así la eficacia y eficiencia de los productos.

Del mismo modo, probablemente deberíamos estar llamando *artesanía de sistemas* a lo que hacemos hoy en día, y no utilizar todavía el término ingeniería de sistemas. Ni siquiera hemos sido capaces de ponernos de acuerdo o encontrar una definición rigurosa de lo que es un *sistema* [23]. Por no hablar de conceptos más complejos como requisitos, necesidades, especificaciones,

arquitectura, propiedades del ciclo de vida, etc. Para que nuestra práctica de la ingeniería de sistemas, que se basa en la experiencia, la intuición, las buenas prácticas y las supuestas buenas ideas, se convierta en ingeniería *real*, en la que podamos evaluar sin lugar a dudas la bondad de un método de ingeniería de sistemas y disponer de un conjunto común y coherente de conceptos y vocabulario asociado, entre otros, necesitamos fundamentos científicos sobre los que construir.

La National Science Foundation (NSF) de Estados Unidos lleva al menos dos décadas apoyando y financiando la investigación fundamental en ingeniería de sistemas. El Departamento de Defensa (DoD) fundó el Systems Engineering Research Center (SERC) hace 15 años como una red de universidades colaboradoras con el objetivo de transformar la práctica de la ingeniería de sistemas mediante la creación de métodos, herramientas y procesos innovadores y el acercamiento entre el mundo académico y la práctica. En el momento de escribir este capítulo, INCOSE ha lanzado su iniciativa Future of Systems Engineering (FuSE), que incluye los Fundamentos de la Ingeniería de Sistemas como una de sus vías de trabajo principales. Aunque estamos lejos de disponer de un conjunto completo y maduro de principios científicos que sustenten la ingeniería de sistemas, la investigación ya ha desvelado varios de ellos, aunque la mayoría aún no se hayan trasladado a la práctica.

Hoy sabemos que la mayoría de los métodos o procesos de decisión que se utilizan en la práctica son fundamentalmente imperfectos y que no hay que fiarse de sus recomendaciones, y por qué. Sabemos que las matrices de riesgo suelen ordenar los riesgos erróneamente con respecto a su criticalidad y por qué. Sabemos que la mayoría de las categorizaciones de requisitos conducen a requisitos deficientes y por qué. Sabemos que los acuerdos de verificación entre clientes y contratistas deben basarse en discusiones programáticas, no técnicas, y por qué. Sabemos que las herramientas MBSE existentes son incapaces de modelar requisitos y que marcar modelos como requisitos conduce a espacios de solución pobres, y por qué. Sabemos que las decisiones en ingeniería no deberían ser consensuadas en general, y por qué. Sabemos que los planes de verificación no deben basarse y contratarse al principio del desarrollo del sistema, y por qué. Y, además, también sabemos cuántas de esas actividades deben realizarse, y por qué. La lista continúa, pero no pretendemos ser exhaustivos, sólo indicativos.



La ingeniería de sistemas moderna debe subsanar estas deficiencias, reconociendo al mismo tiempo la complejidad organizativa y de gobernanza inherente a la concepción y el desarrollo de sistemas a gran escala.

3.2. La ingeniería de sistemas más allá de la coordinación técnica

La conceptualización del ingeniero de sistemas como coordinador técnico o gestor técnico se ha expandido considerablemente en los últimos años. Además de esto, los ingenieros de sistemas de hoy en día pueden asumir otros roles, incluyendo pero no limitándose a ingeniero de requisitos, diseñador o arquitecto de sistemas, analista de sistemas, ingeniero de verificación y validación, ingeniero de interfaces, ingeniero de operaciones e ingeniero de información (incluyendo gestión de configuración y métricas) [24].

Últimamente, estamos viendo incluso un crecimiento significativo de las demandas de trabajo para modeladores de sistemas (o, desafortunadamente, llamados ingenieros MBSE), ingenieros que se especializan en la aplicación de MBSE y apoyan a los ingenieros de sistemas *tradicionales* o a los equipos de ingeniería de sistemas con necesidades de modelado, desde convertir sus ideas en modelos formales hasta encargarse de las tareas de gestión de modelos. En esencia, a medida que el campo madura, los ingenieros de sistemas pueden especializarse en los distintos procesos o tareas que engloba la ingeniería de sistemas.

Para ayudar al desarrollo personal del ingeniero de sistemas, INCOSE ha desarrollado un marco que captura las competencias que un ingeniero de sistemas puede adquirir y debería adquirir cuando se dirige a cierto nivel de competencia [25]. Este marco categoriza las competencias entre básicas (las que todos los ingenieros de sistemas deberían tener, como el pensamiento sistémico o el pensamiento crítico), profesionales (las relacionadas con el trabajo del ingeniero de sistemas dentro de un equipo de ingeniería, como el liderazgo técnico o la negociación), de gestión (las relacionadas con la coordinación técnica, como la gestión de riesgos) y técnicas (las relacionadas con los procesos de ingeniería de sistemas, como los requisitos o la arquitectura), y divide los conocimientos entre cinco niveles de competencia, desde la concienciación hasta el experto.

Una ventaja fundamental del marco es el reconocimiento de que un ingeniero de sistemas no necesita hacerse pasar por el que *todo lo sabe*, sino que su competencia puede desarrollarse en consonancia con la función de ingeniería de sistemas que pueda asumir. Esto permite que los equipos pasen de que cada ingeniero ejecute la ingeniería de sistemas en su dominio (para la cual probablemente no se puede desarrollar una competencia sólida en ingeniería de sistemas) a asignar roles de alta competencia dentro del equipo, compartiendo diferentes tareas de ingeniería de sistemas. En la Tabla 1 se muestra un ejemplo:

Concienciado	Practicante supervisado	Practicante	Líder practicante	Experto
Describe diferentes tipos de requisitos	Ayuda en la obtención de requisitos de las partes interesadas	Obtiene y valida los requisitos de las partes interesadas	Define y documenta políticas, procedimientos, orientaciones y mejores prácticas a nivel empresarial para los procesos de obtención y gestión de requisitos, incluyendo las herramientas asociadas	Entrena a los líderes practicantes en la obtención y gestión de requisitos
Explica por qué es necesaria la existencia de requisitos de alta calidad	Describe las características de los requisitos de buena calidad y proporciona ejemplos	Escribe requisitos de buena calidad y coherentes	Revisa y evalúa la idoneidad y la integridad del conjunto de requisitos	Asesora y arbitra en cuestiones relacionadas con requisitos complejos o sensibles

Tabla 1. Ejemplo de competencias para la definición de requisitos en diferentes niveles de competencia [extraído de [25]]

La especialización en distintas áreas de la ingeniería de sistemas es significativa. Y, aunque vamos en buena dirección, aún queda trabajo por hacer. Por ejemplo, es muy poco probable que alguien asigne a un ingeniero electrónico la responsabilidad de realizar el análisis estructural de un avión. Para esa tarea, se elegiría a un ingeniero mecánico, con varios años de experiencia en la realización de análisis estructurales de complejidad creciente y que, potencialmente, hubiera cursado estudios superiores de ingeniería estructural. Sin embargo, consideramos absolutamente sensato tomar a un ingeniero electrónico y asignarle la tarea de redactar requisitos para un sistema de varios millones de euros después de ofrecerle un seminario de 2 días sobre ingeniería de requisitos. Si bien la mayoría de las investigaciones e informes sobre las causas más frecuentes del fracaso de los proyectos no mencionan los análisis estructurales incorrectos como una de las principales causas; con frecuencia se menciona la obtención correcta de los requisitos como un problema recurrente (por ejemplo, [26-27]).

Para cubrir esta necesidad, el mundo académico está dando un paso adelante, tanto a través de la investigación (como se explica en la sección anterior) como de la educación. El primer programa de grado en ingeniería de sistemas se inició

a principios de los sesenta en la Universidad de Arizona, con un fuerte énfasis en las matemáticas, y uno de los primeros másteres (si no el primero) a finales de los sesenta en Virginia Tech, con un fuerte énfasis en la coordinación o integración técnica. Sin embargo, no fue hasta mediados de la década de 2000 cuando el Stevens Institute of Technology reinventó la educación en ingeniería de sistemas, con títulos de máster centrados en las distintas áreas de especialización y competencias que se requieren en la ingeniería de sistemas moderna. Este paradigma ha desencadenado un inmenso crecimiento de la oferta de másteres en Estados Unidos.

Esto ha sido impulsado por las demandas de la industria; es demasiado arriesgado confiar el futuro de proyectos multimillonarios a individuos que ignoran las *buenas* prácticas y métodos de la ingeniería de sistemas. La ingeniería de sistemas también se está abriendo paso lentamente en Europa, aunque quizá demasiado despacio. Los títulos de ingeniería de sistemas están bastante dispersos en diferentes países en lugar de ser ofrecidos de manera regular por la mayoría de las universidades. Por ejemplo, en el momento de escribir este capítulo, la Universidad Europea de Madrid ofrece el único programa de posgrado en ingeniería de sistemas del país, y ni siquiera es un máster tras la última reforma educativa.

Para que quede claro, lo importante no es el *título*, sino las competencias y la experiencia que puede aportar una formación formal. Esto es algo que ni siquiera se discute en las disciplinas de ingeniería tradicionales, y con suerte, será un paradigma futuro para la ingeniería de sistemas.

El crecimiento y desarrollo de la educación formal en ingeniería de sistemas a todos los niveles (grado, posgrado e incluso recientemente formación profesional) también está cambiando la forma en que la ingeniería de sistemas existe como carrera profesional. Si tradicionalmente la ingeniería de sistemas ha sido una progresión profesional, como se ha descrito anteriormente en el capítulo, empezamos a ver la ingeniería de sistemas como una elección profesional en sí misma, en la que se puede progresar desde puestos junior a puestos más senior. Por ejemplo, un ingeniero de sistemas junior podría ser responsable de derivar y gestionar los requisitos no críticos del sistema, mientras que el ingeniero de sistemas senior centraría su atención sólo en los críticos.

La aparición y adopción del MBSE también está contribuyendo a este cambio de paradigma, ya que cada vez es más fácil para una organización contratar talento MBSE directamente de la universidad que formar y *cambiar las costumbres* de su personal más veterano. Al menos en Estados Unidos, algunos centros de formación profesional han empezado a ofrecer titulaciones para modeladores de sistemas que ocuparían puestos similares a los que tenían los *delineantes* con respecto a los diagramas mecánicos (es decir, dibujo técnico).

3.3. Nuevos métodos y herramientas

El avance de la potencia computacional asequible, junto con la formalización de la ingeniería de sistemas (como se explica en la sección 3.1), han permitido el desarrollo de métodos y herramientas informáticos para apoyar las actividades de ingeniería de sistemas de una forma sin precedentes. Probablemente, la innovación de mayor impacto ha sido el desarrollo del MBSE.

En pocas palabras, el MBSE promueve el uso de modelos formales interpretables por máquinas para capturar artefactos de ingeniería de sistemas que antes se capturaban de forma narrativa o modelos informales utilizando herramientas ofimáticas generales (por ejemplo, procesadores de texto, hojas de cálculo o herramientas de dibujo libre). En principio, no hay limitación en cuanto a qué artefacto de ingeniería de sistemas se captura como un modelo (por ejemplo, requisitos, casos de uso, arquitecturas funcionales, arquitecturas físicas, planes de verificación, entre otros). Además, el MBSE también captura las relaciones entre esos artefactos (por ejemplo, asignación de requisitos a componentes, asignación de funciones a componentes,

trazabilidad entre requisitos y sus actividades de verificación, etc.). Se espera que la transición a un entorno basado en modelos proporcione varios beneficios, que incluyen, entre otros, una mejor comunicación, una mayor consistencia de la información, la reducción de errores en el desarrollo y una mejora de la eficiencia en tiempo y costos [28]. Aunque parece haber consenso entre los profesionales de la ingeniería de sistemas sobre estos beneficios, es importante mencionar que generalmente falta evidencia que demuestre que estos beneficios realmente puedan ser alcanzados [28].

En la actualidad, el MBSE está experimentando un rápido crecimiento en la disponibilidad y las capacidades de las herramientas, los lenguajes de modelado subyacentes y la adopción por parte de la industria y los gobiernos, algo que hace 10 años sólo podíamos soñar. En este sentido, la adopción de Diseño Asistido por Ordenador (CAD) en ingeniería mecánica se ha utilizado a menudo como analogía de cómo el MBSE podría transformar la ingeniería de sistemas. Varias organizaciones están siguiendo un camino de transición similar, optando por adquirir licencias de software, ofrecer una breve formación a sus empleados sobre el uso de las herramientas y dejarles *tirarse a la piscina* con MBSE. Sin embargo, el MBSE no es sólo una evolución tecnológica, como lo fue el CAD [29]. El MBSE tiene implicaciones a nivel de procesos y métodos, además de la adopción de herramientas informáticas específicas, que también deben madurar para que la adopción tecnológica tenga éxito a largo plazo. En otras palabras, el MBSE no mejora una ingeniería de sistemas deficiente. Por ejemplo, en términos de modelado, los usuarios pueden elegir actualmente entre SDL (lenguaje propietario de Vitech Corporation, Inc.), el System Modeling Language (SysML, gestionado por el Object Management Group y para el que se está desarrollando actualmente su v2), Capella (lenguaje abierto desarrollado por Thales), la Object Process Methodology (lenguaje que sustenta una norma ISO) y el Lifecycle Modeling Language (LML).

Estos lenguajes o marcos de modelado no sólo tienen puntos fuertes y débiles diferentes, sino que su estructura subyacente tiene implicaciones significativas para la práctica de la ingeniería de sistemas. Además, los lenguajes no ofrecen en la mayoría de los casos una traducción directa que garantice la compatibilidad entre ellos. No darse cuenta de que es necesario tener en cuenta estas tres facetas para adoptar la MBSE conduce al fracaso [30, 31].

La adopción del MBSE se ha centrado principalmente en las primeras actividades de desarrollo de un sistema, sobre todo en el apoyo a la gestión de requisitos, la arquitectura del sistema y la evaluación de la propagación de cambios, y es predominantemente descriptiva. Es decir, los modelos de ingeniería de sistemas se utilizan para describir aspectos del sistema (por ejemplo, cómo se interconectan dos componentes entre sí) más que para apoyar análisis cuantitativos (por ejemplo, evaluar el rendimiento a nivel de

sistema dado el rendimiento de sus componentes). Existen ciertamente excepciones a esto, que han demostrado la viabilidad de utilizar MBSE en fases posteriores del desarrollo del sistema (como, por ejemplo, para planificar actividades de prueba e integración [32, 33]), y el valor de ampliar los modelos descriptivos para realizar análisis cuantitativos [34] o simulaciones ejecutables [35]. Pero estas capacidades no están aun ampliamente extendidas.

Además, los debates sobre MBSE se están desplazando hacia un paradigma de ingeniería digital más general. En un entorno de ingeniería digital, todos los artefactos de ingeniería se capturan en modelos informáticos que están semánticamente conectados entre sí. Esto significa, por ejemplo, que el atributo de consumo de energía en un modelo electrónico está conectado con su correspondiente atributo de disipación de energía en un modelo térmico y con su correspondiente modelo de consumo de energía en un modelo de sistema de presupuesto de energía. Esto no implica que todos los modelos utilicen los mismos datos o valores para cada parámetro. Más bien, la ingeniería digital permite la formalización de fuentes autorizadas de verdad que garantizan la validez y veracidad de las fuentes de datos. La ingeniería digital no invalida ni sustituye al MBSE, sino que pretende ampliar sus ideas más allá del ámbito de la ingeniería de sistemas y a lo largo de todo el ciclo de vida, desde la formulación del problema hasta la fabricación y el soporte operativo.

A diferencia de lo ocurrido con la adopción de MBSE, que se ha facilitado mediante esfuerzos híbridos descendentes y ascendentes (es decir, dirigidos en algunos casos por equipos de ingenieros y en otros apoyados por la dirección de la empresa o el mandato del cliente) [24], la adopción de la ingeniería digital está siendo fuertemente exigida desde *arriba*, generalmente por el cliente. El Departamento de Defensa (DoD) de EE.UU., por ejemplo, ha establecido una estrategia dedicada a la ingeniería digital [36], que se ha trasladado rápidamente a sus ramas de servicio. Una acción similar ha sido llevada a cabo por el Ministerio de Defensa en España [37]. Este enfoque pone de manifiesto un conflicto entre la urgencia con la que el cliente o la dirección de la organización desean hacer avanzar su capacidad y la preparación de la comunidad de práctica para ofrecer soluciones eficaces en términos de procesos y herramientas. En la actualidad, la capacidad prevista sigue estando muy lejos de la preparación tecnológica para el apoyo operativo.

La potencia de cálculo también ha permitido acelerar tareas que, de otro modo, requerirían mucho tiempo. Tradicionalmente, la exploración de la solución durante el diseño conceptual o los esfuerzos de arquitectura del sistema se han limitado a intercambiar un puñado de alternativas en busca de la solución preferida. Sin embargo, los modelos de sistemas estructurados pueden utilizarse ahora para aumentar el tamaño del espacio de soluciones

que puede explorarse, denominado *tradespace*, y evaluar automáticamente miles, si no millones, de soluciones a la vez [28]. Lógicamente, cuanto mayor sea el espacio de soluciones que se explore, más probabilidades habrá de encontrar una solución mejor. Esto cambia el enfoque del diseño conceptual, que pasa de refinar un conjunto reducido de posibles soluciones individuales a construir modelos estructurales modulares que permitan una exploración amplia y profunda.

Asimismo, el desarrollo de sistemas se ha limitado tradicionalmente a una única solución que se elige y se perfecciona iterativamente hasta que pasa a producción y posteriormente se despliega. Esto significa que se elige un concepto de sistema o una arquitectura de sistema y, a continuación, se pasa por un diseño detallado, en el que pueden producirse iteraciones, a medida que la solución se perfecciona hasta convertirse en una solución operativa con la madurez suficiente para ser fabricada y, finalmente, poner en servicio.

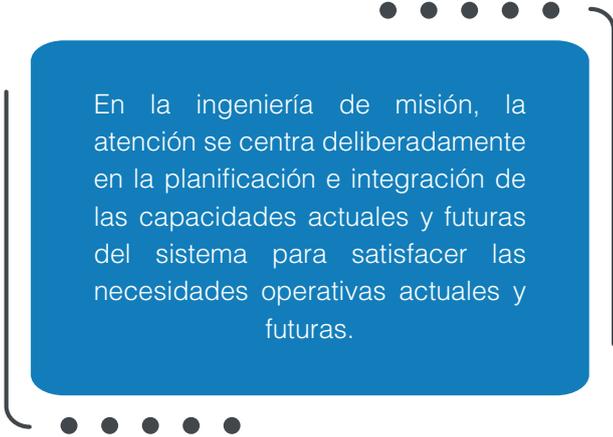
Este paradigma se ha denominado diseño basado en puntos, ya que el diseño se basa en la elección temprana de una única solución (punto) en el espacio de soluciones. Sin embargo, los modelos informáticos pueden reducir considerablemente el esfuerzo que supone mantener y perfeccionar una solución. El novedoso paradigma del diseño basado en conjuntos propone aprovechar esta ventaja para evitar anclarse a una solución específica en una fase temprana del desarrollo del sistema [39], cuando el conocimiento es limitado [4]. En el diseño basado en conjuntos, se elige un conjunto de soluciones (puntos) en el espacio de soluciones y se refinan conjuntamente. Las soluciones poco atractivas sólo se descartan del conjunto cuando se dispone de conocimientos suficientes para hacerlo con suficiente confianza. Esto proporciona flexibilidad en el proceso de desarrollo sin requerir cambios en las soluciones, ya que se conservan las soluciones aceptables.

La forma en que las experiencias pasadas y los conocimientos organizativos pueden utilizarse en la ingeniería de sistemas también está cambiando drásticamente gracias al uso de modelos legibles por máquinas y a la ayuda computacional. Los asistentes cognitivos son máquinas (normalmente sistemas de software) que ayudan a los humanos en tareas cognitivas [40]. Alexa de Amazon o Siri de Apple son buenos ejemplos de asistentes cognitivos que hemos empezado a utilizar con frecuencia, y ChatGPT está ampliando aún más las capacidades que este tipo de sistemas pueden proporcionar a los humanos. Un aspecto clave es que el ingeniero puede utilizar el lenguaje natural para desarrollar de forma dinámica e iterativa artefactos de ingeniería de sistemas con el apoyo del asistente. En ingeniería de sistemas, por ejemplo, ahora tenemos la capacidad no sólo de pedir a un asistente cognitivo que desarrolle soluciones conceptuales o arquitecturas de sistemas para un conjunto

dado de parámetros, sino también de explorar el espacio de soluciones en varias direcciones y explicar sus elecciones y recomendaciones [41-43].

La evolución de los métodos y herramientas no se ha limitado a aquellos que han permitido los avances en las capacidades informáticas. También estamos asistiendo a la aparición de nuevos procesos, modelos y paradigmas de desarrollo que han surgido de la necesidad de hacer frente a las peculiaridades del nuevo contexto y los tipos de misiones en los que y a los que se aplica ahora la ingeniería de sistemas, tal y como se ha comentado en las secciones anteriores.

Los procesos de desarrollo más iterativos, menos encorsetados y más rápidos de desplegar son cada vez más frecuentes (por ejemplo, ágiles, DevOps) y se integran con otros más tradicionales (por ejemplo, Cascada, Uve). Se ha avanzado considerablemente en el desarrollo de la noción de ingeniería de sistemas de sistemas, que últimamente se ha redefinido como *ingeniería de misión*.



En la ingeniería de misión, la atención se centra deliberadamente en la planificación e integración de las capacidades actuales y futuras del sistema para satisfacer las necesidades operativas actuales y futuras.

La principal distinción en este caso es que esas capacidades del sistema son proporcionadas por sistemas gobernados o gestionados de forma independiente.

Este debate no es exhaustivo. Hay otros avances en ingeniería de sistemas que probablemente merezcan la pena señalar, pero estos deberían proporcionar una idea de cuál es el panorama de la práctica actual de la ingeniería de sistemas (aunque todavía sea el *futuro* para algunas organizaciones).

4. CONCLUSIONES

El panorama de la ingeniería de sistemas ha cambiado en los últimos años, y se han plantado las semillas para promover nuevos cambios en los años venideros. En este capítulo se han insinuado cuáles son esos cambios y por qué han sido o son necesarios. Los recursos ya no son ilimitados, sino más bien escasos. Los esfuerzos de la ingeniería de sistemas traspasan las fronteras geopolíticas y deben equilibrar las consecuencias de las consolidaciones industriales, la rápida evolución de las tecnologías y la aparición de empresas emergentes ágiles, así como la rápida evolución de las necesidades del mercado. La naturaleza de los sistemas que debemos desarrollar y con los que debemos trabajar ahora también ha cambiado. La mayoría de los desarrollos ya no se producen en un campo verde, y a menudo debemos conciliar las limitaciones de los grandes sistemas heredados con aspectos novedosos relacionados con el aumento de la naturaleza cibernética de los sistemas, la gobernanza distribuida e incluso la inteligencia.

El ingeniero de sistemas debe pasar de ser un simple coordinador técnico a un experto en ingeniería de sistemas. Convertirse en ingeniero de sistemas en el puesto de trabajo no basta para afrontar con eficacia las tareas de ingeniería de sistemas actuales. El ingeniero de sistemas debe aprender métodos y técnicas novedosos, muchos de los cuales empiezan a sustentarse en la investigación; muchas buenas prácticas han resultado no ser tan buenas y muchos procesos maduros ya no son pertinentes. La tecnología está ocupando un lugar central en el esfuerzo de la ingeniería de sistemas, empezando por el suministro de activos digitales que pueden conectarse entre dominios y avanzando hacia el uso de asistentes cognitivos para aumentar las capacidades del ingeniero humano. Aunque todavía incipientes, estas capacidades están evolucionando rápidamente.

La ingeniería de sistemas ha llegado al siglo XXI y seguirá madurando y evolucionando. ¿Está preparada su empresa, compañía u organización para modernizar sus prácticas de ingeniería de sistemas?

REFERENCIAS

1. Ryschkewitsch, M., D. Shaible, and W.J. Larson, The art and science of systems engineering. Systems Research Forum, 2009. 03(02): p. 81-100.
2. Salado, A., Systems engineering, in The Engineering Management Handbook, B. Mesmer, et al., Editors. 2023, The American Society of Engineering Management: Huntsville, AL, USA. p. 361-370.
3. Pyster, A., N. Hutchison, and D. Henry, The Paradoxical Mindset of Systems Engineers. 2018, Hoboken, NJ, USA: John Wiley and Son, Inc.
4. Blanchard, B.S. and W.J. Fabrycky, Systems engineering and analysis. Vol. 4. 1990: Prentice Hall New Jersey;.
5. Griffin, M.D., How do we fix systems engineering?, in 61st International Astronautical Congress. 2010: Prague, Czech Republic.
6. Zinko, R., W.A. Gentry, and M.D. Laird, A development of the dimensions of personal reputation in organizations. International Journal of Organizational Analysis, 2016. 24(4): p. 634-649.
7. Manzoor, E., et al., Influence via Ethos: On the Persuasive Power of Reputation in Deliberation Online. Management Science. 0(0): p. null.
8. Delicado, B.A., A. Salado, and R. Mompó, Conceptualization of a T-Shaped engineering competency model in collaborative organizational settings: Problem and status in the Spanish aircraft industry. Systems Engineering, 2018. 21(6): p. 534-554.
9. Collopy, P.D. Systems engineering theory: What needs to be done. in Systems Conference (SysCon), 2015 9th Annual IEEE International. 2015.
10. Frosch, R.A., A new look at systems engineering. IEEE Spectrum, 1969: p. 24-28.
11. INCOSE, Systems Engineering Handbook. A Guide for System Life Cycle Processes and Activities. 5th ed. 2023, Hoboken, NJ, USA: John Wiley and Sons, Inc.
12. Villanueva, C.D., El Programa FCAS y la Industria Española de Defensa: una apuesta equivocada, in Revista Ejército. 2022.
13. Salado, A. and R. Nilchiani, A Research on Measuring and Reducing Problem Complexity to Increase System Affordability: From Theory to Practice. Procedia Computer Science, 2015. 44: p. 21-30.
14. (ISO), I.S.O., Systems and software engineering — Guidelines for the utilization of ISO/IEC/IEEE 15288 in the context of system of systems (SoS). 2019.
15. Salado, A. Abandonment: A natural consequence of autonomy and belonging in systems-of-systems. in System of Systems Engineering Conference (SoSE), 2015 10th. 2015.

16. Madni, A.M. and M. Sievers, System of Systems Integration: Key Considerations and Challenges. *Systems Engineering*, 2014. 17(3): p. 330-347.17.
17. Shadab, N., A.U. Kulkarni, and A. Salado, Shifting Paradigms in Verification and Validation of AI-Enabled Systems: A Systems-Theoretic Perspective, in *Systems Engineering and Artificial Intelligence*, W.F. Lawless, et al., Editors. 2021, Springer International Publishing: Cham. p. 363-378.
18. Shadab, N., et al. Product Herding for Intelligent Systems. in *Conference on Systems Engineering (CSER)*. 2023. Hoboken, NJ, USA.
19. INCOSE, *Systems Engineering Vision 2035*. 2023.
20. Wymore, A.W., *A mathematical theory of systems engineering: The elements*. 1967, New York: Wiley.
21. Warfield, J.N. and J.D. Hill, *A unified systems engineering concept*. Vol. Monograph 1. 1972, Columbus: Batelle Memorial Institute.
22. Mesarovic, M.D. General systems theory and its mathematical foundation. in *IEEE Systems Science and Cybernetics Conference*. 1967. Boston, MA.
23. Salado, A. and A.U. Kulkarni, An Assessment of the Adequacy of Common Definitions of the Concept of System. *INCOSE International Symposium*, 2021. 31(1): p. 510-521.
24. Sheard, S.A., *TWELVE SYSTEMS ENGINEERING ROLES*. *INCOSE International Symposium*, 1996. 6(1): p. 478-485.
25. INCOSE, *Systems Engineering Competency Framework*. 2018.
26. GAO, *DHS Annual Assessment: Major Acquisition Programs Are Generally Meeting Goals, but Cybersecurity Policy Needs Clarification*. 2023.
27. GAO, *Space Acquisitions: DOD Faces Significant Challenges as it Seeks to Accelerate Space Programs and Address Threats*. 2019.
28. Henderson, K. and A. Salado, Value and benefits of model-based systems engineering (MBSE): Evidence from the literature. *Systems Engineering*, 2021. 24(1): p. 51-66.
29. Henderson, K. and A. Salado, Is CAD A Good Paradigm for MBSE? *INCOSE International Symposium*, 2021. 31(1): p. 144-157.
30. Henderson, K., T. McDermott, and A. Salado, MBSE adoption experiences in organizations: Lessons learned. *Systems Engineering*. n/a(n/a).
31. Henderson, K. and A. Salado, The Effects of Organizational Structure on MBSE Adoption in Industry: Insights from Practitioners. *Engineering Management Journal*, 2023. In press.
32. Salado, A., 5.5.2 Efficient and Effective Systems Integration and Verification Planning Using a Model-Centric Environment. *INCOSE International Symposium*, 2013. 23(1): p. 1159-1173.
33. Gregory, J. and A. Salado, Model-Based Verification Strategies Using SysML and Bayesian Networks, in *Conference on Systems Engineering Research (CSER)*. 2023: Hoboken, NJ, USA.
34. Hecht, M. and J. Chen. Use of SysML for Quantitative System Reliability and Availability Analysis. in *2022 Annual Reliability and Maintainability Symposium (RAMS)*. 2022.
35. Karban, R., et al., Creating system engineering products with executable models in a model-based engineering environment. *SPIE Astronomical Telescopes + Instrumentation*. Vol. 9911. 2016: SPIE.
36. DOD, Department of Defense Digital Engineering Strategy. 2018, Office of the Deputy Assistant Secretary of Defense for Systems Engineering: Washington, DC, USA.
37. Defensa, M.d., *Plan de Acción del Ministerio de Defensa para la Transformación Digital*. 2020.
38. Ross, A.M. and D.E. Hastings, 11.4.3 The Tradespace Exploration Paradigm. *INCOSE International Symposium*, 2005. 15(1): p. 1706-1718.
39. Shallcross, N., et al., Set-based design: The state-of-practice and research opportunities. *Systems Engineering*, 2020. 23(5): p. 557-578.
40. Salado, A. and D. Selva, Asistentes cognitivos en ingeniería, in *UEM STEAM Essentials*. 2021.
41. Martin, A.V. and D. Selva, Explanation Approaches for the Daphne Virtual Assistant, in *AIAA Scitech 2020 Forum*. 2020.
42. Martin, A.V. and D. Selva, From Design Assistants to Design Peers: Turning Daphne into an AI Companion for Mission Designers, in *AIAA Scitech 2019 Forum*. 2020.
43. Martin, A.V.i. and D. Selva, Daphne: A Virtual Assistant for Designing Earth Observation Distributed Spacecraft Missions. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2020. 13: p. 30-48.

DR. DINESH VERMA

El Dr. Dinesh Verma es el director ejecutivo fundador del Centro de Investigación en Ingeniería de Sistemas (SERC) y catedrático de Ingeniería de Sistemas en la School of Systems and Enterprises (SSE) del Stevens Institute of Technology.



Fue decano fundador de SSE durante diez años, de 2007 a 2016. Durante sus quince años en Stevens ha propuesto con éxito programas académicos y de investigación por valor de más de 175 millones de dólares. Verma fue asesor científico del director

del Embedded Systems Institute de Eindhoven (Holanda) de 2003 a 2008. Antes de ocupar este cargo, fue Director Técnico en Lockheed Martin Undersea Systems, en Manassas, Virginia, en el área de sistemas adaptados y procesos de ingeniería de soporte, métodos y herramientas para el desarrollo de sistemas complejos.

Antes de incorporarse a Lockheed Martin, Verma trabajó como Científico Investigador en Virginia Tech y dirigió el Laboratorio de Diseño de Ingeniería de Sistemas de la Universidad. Durante su estancia en Virginia Tech y posteriormente, Verma sigue prestando sus servicios a numerosas empresas en calidad de consultor. Entre 1995 y 2000 fue profesor invitado en la Universidad de Exeter (Reino Unido).

Sus actividades profesionales y de investigación se centran en la ingeniería y el diseño de sistemas, con especial atención a la evaluación del diseño conceptual, el diseño preliminar y la arquitectura de sistemas, la toma de decisiones de diseño, el cálculo de los costes del ciclo de vida y la ingeniería de apoyo. Además de sus publicaciones, Verma ha recibido tres patentes en las áreas de cálculo de costes del ciclo de vida y técnicas de lógica difusa para evaluar conceptos de diseño.

El Dr. Verma es autor de más de 100 artículos técnicos, reseñas de libros, monografías técnicas y coautor de tres libros de texto. Obtuvo su maestría y doctorado en ingeniería industrial y de sistemas en Virginia Tech. En enero de 2007 fue nombrado Doctor Honoris Causa en Tecnología y Diseño por la Universidad de Linnaeus (Suecia) y en septiembre de 2008 obtuvo un Máster Honoris Causa en Ingeniería por el Stevens Institute of Technology. Es fellow del Consejo Internacional de Ingeniería de Sistemas (INCOSE).

DR. ALEJANDRO SALADO

El Dr. Alejandro Salado es profesor titular de ingeniería de sistemas en el Departamento de Ingeniería de Sistemas e industriales de la Universidad de Arizona y director de los programas de ingeniería de sistemas. Además, ofrece consultoría a tiempo parcial en áreas relacionadas con la transformación empresarial, el cambio cultural de los equipos técnicos, la ingeniería de sistemas y las estrategias de ingeniería.



Alejandro investiga sobre la formulación de problemas, el diseño de estrategias de verificación y validación, la ingeniería de sistemas basada en modelos y la formación en ingeniería. Antes de incorporarse al mundo académico, ocupó puestos de ingeniero de sistemas, arquitecto jefe e ingeniero jefe de sistemas en sistemas espaciales tripulados y no tripulados de hasta 1.000 millones de dólares de coste de desarrollo.

Ha publicado más de 150 artículos técnicos y sus investigaciones han recibido financiación federal de la National Science Foundation (NSF), el Naval Surface Warfare Command (NSWC), el Naval Air System Command (NAVAIR) y la Office of Naval Research (ONR), entre otros. Ha recibido el Premio CAREER de la NSF, el Premio Internacional Fulbright de Ciencia y Tecnología y varios premios a sus publicaciones.

El Dr. Salado es ingeniero de telecomunicaciones por la Universidad Politécnica de Valencia y tiene un máster en gestión de proyectos y otro en ingeniería electrónica por la Universidad Politécnica de Cataluña, el SpaceTech máster en ingeniería de sistemas espaciales por la Universidad Técnica de Delft y un doctorado en ingeniería de sistemas por el Stevens Institute of Technology.

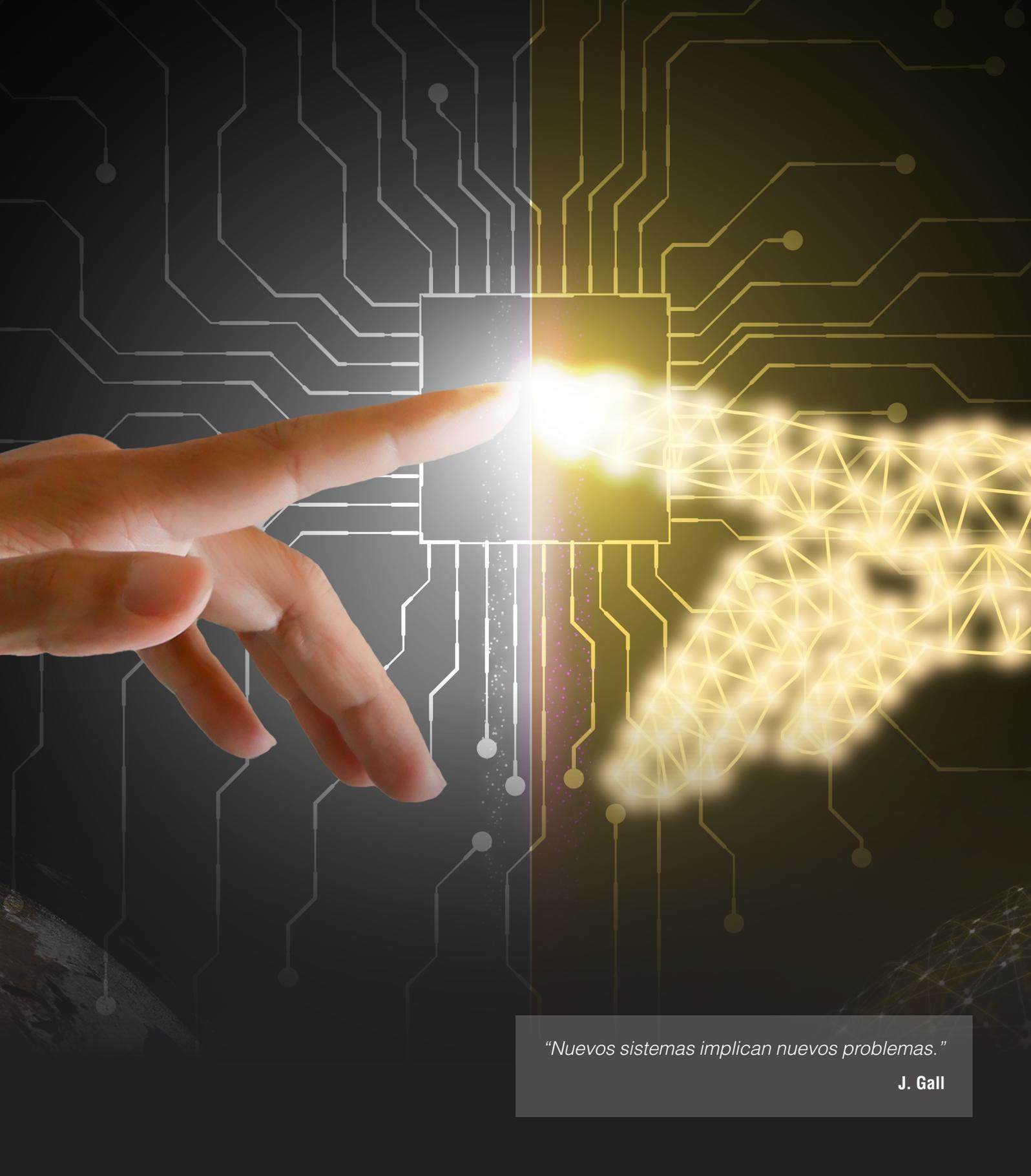
ADOLFO SÁNCHEZ

Adolfo Sánchez es ingeniero en informática por la Universidad de Zaragoza y licenciado en Historia por la UNED.

Ha participado en la definición y desarrollo de un evaluador de arquitecturas modulares de aviónica, en el mantenimiento del software del Eurofighter, en desarrollos de software de aseguramiento de la calidad realizados en programas

estratégicos de guerra electrónica y en la adquisición, despliegue e implantación del Programa CIS de la Unidad Militar de Emergencias. Actualmente es coordinador del Área CIS de Defensa en Isdefe prestando servicio a la Subdirección General de Transformación Digital del Ministerio de Defensa. Es certificado CSEP (Certified Systems Engineering Professional) por INCOSE y es formador interno de diversos cursos de ingeniería de sistemas en Isdefe.





"Nuevos sistemas implican nuevos problemas."

J. Gall

Nuevos tipos de sistemas

Tom McDermott, *Stevens Institute of Technology* (tmcdermo@stevens.edu)

Víctor Ramos, *Isdefe* (vramos@isdefe.es)

CAPÍTULO 2

Resumen

En este capítulo se exponen cuatro características que presentan los sistemas modernos y futuros que están cambiando las prácticas empleadas tradicionalmente por la ingeniería de sistemas: la adaptabilidad de los sistemas, los sistemas ciberfísicos altamente interconectados, los sistemas basados en el aprendizaje con los cuales las personas pueden formar equipo, y la gobernanza distribuida de los sistemas de sistemas. En primer lugar, se destacan las prácticas empleadas hoy en día que presentan una dudosa efectividad sobre este tipo de sistemas para, posteriormente, exponer las tendencias actuales a la hora de abordar estos aspectos únicos y propios de los nuevos sistemas.

Palabras clave

Sistemas ciberfísicos, inteligencia artificial, sistemas adaptativos, desarrollo ágil, sistemas basados en el aprendizaje, gobernanza distribuida, complejidad.



1. INTRODUCCIÓN

Existen varias tendencias que están cambiando la naturaleza de los sistemas que desarrollamos y con los que interactuamos hoy en día:

1. El aumento continuo del ritmo en la innovación tecnológica. La disponibilidad de la investigación básica a través de Internet ha propiciado que la ciencia y la tecnología sean un producto accesible y que la introducción de innovaciones técnicas tanto en las misiones como en las plataformas existentes impulsen una adaptación y un cambio continuos. Los sistemas son mucho menos estables que antes y la competencia favorece a quienes pueden adaptarse más rápidamente al cambio impulsado por la tecnología, no necesariamente a aquellos que pueden invertir más para aplicar tecnologías en una determinada misión.
2. El comportamiento de los sistemas está, y seguirá estando, definido más por el software que por el hardware. Todos los sistemas son, o serán, en gran medida digitales y estarán respaldados por infraestructuras digitales conectadas. Los sistemas hardware seguirán siendo importantes, pero tenderán a ser de propósito más general, más asequibles y distribuidos, incluso en aplicaciones críticas como las de defensa.
3. Los datos se recopilan y están cada vez más disponibles para su análisis. La capacidad de almacenamiento y procesamiento de los ordenadores continúa creciendo de forma exponencial. Los sistemas modernos y, más aún, los futuros se adaptarán a unas condiciones internas y externas cambiantes, en función de la información aportada por unos datos también cambiantes. La forma en la que desarrollamos los sistemas cambiará para mejorar la recolección y el empleo de los datos, aprovechando los datos y los modelos para mejorar la eficiencia en el desarrollo y la gestión de los programas.
4. Todo cada vez está más interconectado, abundando complejos sistemas de sistemas de naturaleza global. Cada vez más, los sistemas ciberfísicos (CPS, *Cyber-Physical Systems*) están conectados a otros sistemas para compartir datos y recursos como parte de sistemas de sistemas más grandes. Esto abre tanto nuevas oportunidades, como nuevas vulnerabilidades sobre las capacidades del sistema. El rigor de la ingeniería de sistemas es necesario para equilibrar la apertura de estos sistemas con cuestiones como la seguridad física, la seguridad de la información, la privacidad, la confianza, etc.

5. La automatización y la personalización al usuario hacen a los sistemas más eficientes, configurables y adaptables. En particular, las tareas realizadas por personas dependen mucho más de la colaboración con sistemas, total o parcialmente, automatizados. Con la inteligencia artificial (IA) y el aprendizaje automático (ML, *Machine Learning*) esta tendencia continuará avanzando hacia una nueva clase de sistemas conocida como Sistemas Basados en el Aprendizaje (LBS, *Learning Based Systems*). Un LBS es una nueva clase de sistema informático que proporciona su funcionalidad y sus prestaciones empleando técnicas de ML. En un futuro próximo, los operadores humanos y los LBS interactuarán y se adaptarán para completar misiones complejas de forma conjunta, un concepto conocido como *Human-Machine Teaming* (HMT).

Estas tendencias están dando lugar a nuevos tipos de sistemas, que hemos categorizado en cuatro tipos evolutivos e interrelacionados:

- **Sistemas Altamente Adaptables:** Los sistemas modernos hacen un uso extensivo del software por el valor añadido de su funcionalidad para el usuario, ya que es más fácil de modificar y minimiza los costes de producción. Los sistemas son cada vez más adaptables a cada usuario. Como resultado, existe una tendencia hacia sistemas que sean más fáciles de adaptar en base a unas necesidades cambiantes. Además, las prácticas de desarrollo y de soporte altamente adaptables han impregnado los modelos de negocios de los sistemas para acentuar entregas de valor más frecuente y consistente en términos de introducción de nuevas capacidades y de una mayor flexibilidad. El aprendizaje y la iteración se han convertido en elementos más valiosos que la comprensión y la estabilidad de los requisitos iniciales, incluso en sistemas críticos.
- **Sistemas Ciberfísicos Altamente Interconectados:** Los sistemas modernos están altamente interconectados y dependen totalmente de una funcionalidad definida por software. El Internet de las Cosas (IoT, *Internet of Things*), como infraestructura, está dando lugar a sistemas CPS a gran escala como ciudades inteligentes, transporte interconectado, sensores ubicuos y fabricación 4.0. Los CPS altamente interconectados generan preocupaciones sobre la seguridad física, la seguridad de la información y la confianza en dominios donde anteriormente estas cuestiones no se consideraban prioritarias. Los CPS interconectados a gran escala aumentan la complejidad y la superficie efectiva de los sistemas con los que interactuamos. Los gemelos digitales (modelos digitales de CPS conectados y que operan con CPS físicos) son ahora un dominio de la ingeniería de sistemas en rápido crecimiento.



Figura 1. Características del aumento de la complejidad de los sistemas [12]

- **Sistemas Basados en el Aprendizaje y HMT:** Los sistemas emergentes estarán cada vez compuestos en mayor medida por LBS, que son sistemas cuyos comportamientos se aprenden en lugar de programarse. Éstos actuarán de forma conjunta con personas (HMT) para aumentar la inteligencia y la *agencia*¹ humanas. El impacto de estos sistemas en las tareas realizadas por las personas y en la conciencia situacional también será una propiedad emergente de los sistemas futuros. Los LBS están ampliando la definición de confianza en los sistemas, desde preocupaciones principalmente de fiabilidad hasta preocupaciones humanas como la ética y la justicia. La gestión de la incertidumbre, así como la prueba, verificación y validación de los LBS son desafíos emergentes para la ingeniería de sistemas.
- **Gobernanza distribuida en Sistemas de Sistemas:** Las prestaciones y las capacidades de un sistema dependen de la información y de las funciones compartidas con sistemas externos que no se pueden controlar por completo. Esto lleva a la generación de comportamientos emergentes complejos y de diferentes métodos de gobernanza. Se han establecido metodologías de sistemas de sistemas (SoS, *Systems of Systems*) para ayudar a gestionar la gobernanza, pero al igual que con otras prácticas históricas de la ingeniería de sistemas, éstas aún no están a la altura de la rápida adaptación de los SoS y de la gobernanza compartida.

El tema que subyace en estos cuatro tipos de sistemas es el rápido crecimiento de su complejidad. La Figura 1 destaca las preocupaciones derivadas de la complejidad que impulsan estos nuevos tipos de sistemas [12].

1. *Agencia* en este contexto se refiere a la capacidad del ser humano de tomar acciones por sí mismo, estimuladas intrínsecamente.

La ingeniería de sistemas siempre se ha visto como una metodología para ayudar a gestionar la complejidad de los sistemas, pero tanto sus tipos como las preocupaciones asociadas a los mismos están cambiando, y se deben actualizar los métodos empleados. Éste es el tema abordado en este capítulo.

El siguiente apartado evalúa la efectividad de los métodos tradicionales empleados por la ingeniería de sistemas cuando se aplican a estos nuevos tipos de sistemas y las preocupaciones generadas en el ámbito de la ingeniería de sistemas asociada a los mismos. A continuación, analizamos algunas de las tendencias que están impulsando el cambio en la forma en la que practicamos la ingeniería de sistemas como consecuencia de la aparición de estos nuevos tipos de sistemas. El contenido expuesto no se deriva de una revisión exhaustiva de la literatura, sino de investigaciones en curso y de hojas de ruta definidas para la investigación realizadas por el Centro de Investigación en Ingeniería de Sistemas (SERC, *Systems Engineering Research Centre*) del Instituto de Tecnología Stevens en Hoboken de Nueva Jersey, EE.UU.

2. EVALUACIÓN DE LA EFECTIVIDAD DE LAS PRÁCTICAS TRADICIONALES DE INGENIERÍA DE SISTEMAS APLICADAS A ESTOS NUEVOS TIPOS DE SISTEMAS

2.1. Sistemas altamente adaptables

La adaptabilidad de los sistemas se define como la capacidad de un sistema para adaptarse de manera eficiente y rápida a circunstancias cambiantes. Un sistema adaptativo es, por tanto,

un sistema que es capaz de adaptar su comportamiento a cambios en su entorno o en elementos del propio sistema [28]. El Cuerpo de Conocimientos de Ingeniería de Sistemas (SEBOK, *Systems Engineering Body of Knowledge*) considera la adaptabilidad como la capacidad del sistema para satisfacer cambios en su misión y en sus requisitos impulsados externamente con o sin modificación del propio sistema, medido a través de algún indicador de valor como coste, tiempo o recursos [29]. Esta definición implica que el cambio impulsado por el contexto es un proceso intencionado. Esta definición no considera de forma efectiva la naturaleza de los LBS actuales, ya que éstos se actualizan continuamente en función de comportamientos aprendidos o de resultados que responden a un contexto externo cambiante. Para este tipo de sistemas no resulta ya apropiado separar al sistema de su contexto externo; ambos deben ser comprendidos y modelados conjuntamente.

Los sistemas altamente adaptables se caracterizan por un uso intensivo del software, por estar altamente conectados y por tener amplias capacidades de automatización y de adaptación a cada usuario. En el núcleo de estos sistemas están los datos que impulsan su comportamiento, el cual se define por la lógica del software, los algoritmos y las funciones de control y de gestión de los datos. El mecanismo que subyace es una conexión entre el conjunto de datos del sistema y el entorno externo que permite que los comportamientos del sistema cambien rápidamente en respuesta a cambios del contexto externo. Este puede ser un proceso intencionado de aprendizaje y de modificación basado en datos en respuesta a una misión cambiante, o puede ser altamente automatizado en respuesta a un contexto externo cambiante como ocurre con los LBS emergentes.

No obstante, la ingeniería de sistemas, fuertemente influenciada por las necesidades de los ámbitos aeroespacial y de defensa, ha seguido vinculada a la realización física de grandes sistemas complejos y de otras capacidades críticas que se pretende que persistan durante muchos años. Siempre existirá la necesidad de una rigurosa definición, análisis y prueba de estos sistemas críticos, pero los procesos del ciclo de vida que elijamos deben ser adaptados a un uso y a una vida del sistema reales. Por ejemplo, la IS para los sistemas modernos tiende a basarse más en modelos, a ser más ágil y a responder a las necesidades de los usuarios, lo que puede lograrse con procesos de ciclo de vida más adaptables y eficientes aprovechando datos y modelos. [16]. La ingeniería de sistemas de software (SSE, *Software Systems Engineering*), las tecnologías de la información, la arquitectura empresarial, el modelado y la simulación distribuidos y los sistemas de fabricación automatizados deben ser aprovechados de manera convergente para abordar la gestión del ciclo de vida de los sistemas altamente adaptables.

Los procesos técnicos y de gestión modernos de la Ingeniería de sistemas transforman los datos en vistas a través de modelos, que respaldan los análisis que conducen a la toma de decisiones. Este flujo de procesos digitales soporta “la transformación de datos en modelos y luego el análisis a través de vistas para tomar decisiones documentadas sobre artefactos digitales”. Este flujo de procesos no es nuevo, pero está evolucionando desde una naturaleza ineficiente y en gran medida manual a un flujo de procesos altamente automatizado impulsado por ciclos de adaptación rápidos, consistentes y basados en valor. En consecuencia, las prácticas de desarrollo software han evolucionado para gestionar esta automatización. La Figura 2 representa de nuevo las etapas “Definir -> Realizar -> Implementar y usar”, ampliamente recogidas en el proceso del ciclo de vida del modelo en V de la Ingeniería de sistemas, como un proceso circular para representarlo como:

- 1) Un conjunto de datos centrales, interpretados mediante transformaciones de modelos, que lleva a la toma de decisiones de diseño.
- 2) La estratificación entre disciplinas y tareas de ingeniería para producir artefactos de apoyo a la decisión.
- 3) La consideración de procesos continuamente iterativos a los que se puede entrar desde cualquier punto.

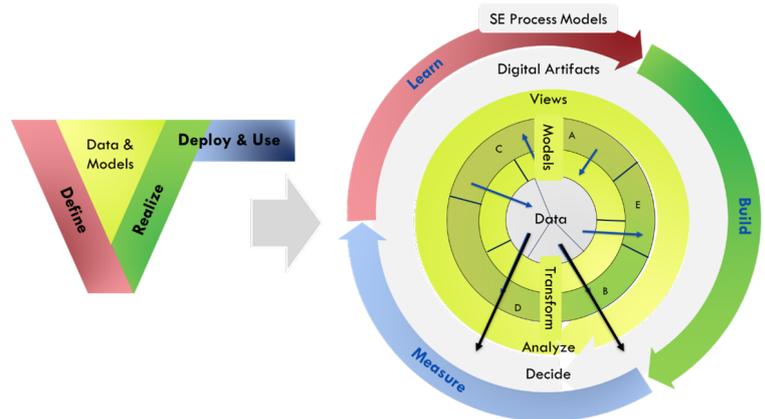


Figura 2. Procesos circulares con datos en su núcleo [16]

El desafío que presentan los sistemas altamente adaptables es mantener un nivel de rigor adecuado en la práctica de la Ingeniería de sistemas y en la definición de los procesos asociados para garantizar que estos sistemas sigan siendo escalables, resilientes, seguros (desde el punto de vista físico y de la seguridad de la información) y utilizables por operadores humanos. Los nuevos procesos de la Ingeniería de sistemas para el ciclo de vida comprenden conjuntos de datos y de modelos digitales compartidos y gestionados con autoridad durante todo el ciclo de vida del sistema, no solo un único ciclo de vida de ingeniería o de programa [16].

2.2. Sistemas Ciberfísicos Altamente Interconectados (CPS)

Los sistemas altamente adaptables están evolucionando desde sistemas exclusivamente software hasta sistemas software y hardware totalmente conectados, conocidos como CPS. La Fundación Nacional de Ciencias de Estados Unidos (NSF, *National Science Foundation*) define a los CPS como “sistemas diseñados que se construyen a partir de la perfecta integración de algoritmos y de componentes físicos, de la cual dependen” [19]. Un CPS tiene ordenadores y redes que controlan procesos físicos, a menudo caracterizados por bucles de realimentación que afectan tanto a los cálculos como a los resultados físicos generados. La Figura 3 proporciona una descripción general en capas de un marco CPS [11]. Como se muestra en la figura, el diseño de un CPS debe abordar las actividades de control sobre el dispositivo de interés, pero también sobre los sistemas humanos y de máquinas interconectados que interactúan con él, lo que puede ocurrir a largas distancias en las redes cibernéticas [10].

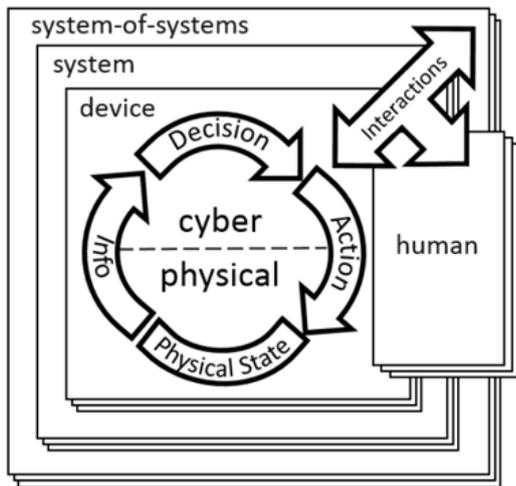


Figura 3. Marco general de un CPS (11). Reeditado por cortesía del NIST (National Institute of Standards and Technology)

Además, los CPS transforman la manera en la que las personas interactúan con los sistemas [19]. Por ejemplo, los humanos ahora pueden interactuar con sistemas diseñados a través de redes cibernéticas en lugar de hacerlo directamente (como controlar un termostato desde un teléfono móvil) o incluso a través de múltiples CPS interconectados y de sistemas software a gran escala. Estas interacciones son posibles gracias a unos mayores grados

de automatización y de autonomía [19], lo que aumenta, sin embargo, tanto la complejidad de los métodos de control del sistema, como la velocidad de evolución del mismo en respuesta a su contexto externo.

Tradicionalmente, la Ingeniería de sistemas ha diferenciado cómo se constituye un sistema internamente (es decir, su estructura) de cómo éste se manifiesta externamente (es decir, su comportamiento) y ha deducido su función a partir de su estructura. Esta idea es la base que subyace de la construcción jerárquica de sistemas con interfaces de entrada y de salida definidos en múltiples componentes modulares. Esto ha dado lugar a centrar la Ingeniería de sistemas en una representación estructural de los sistemas, apoyando el comportamiento físico en sistemas principalmente físicos. Sin embargo, los CPS altamente conectados son de naturaleza heterárquica. Se componen de numerosos elementos heterogéneos que actúan de forma independiente e interdependiente. Debido a esta complejidad, los modelos tradicionales de descomposición estructural y basados en la física resultan insuficientes. En los sistemas complejos, la forma (estructura) y la función (comportamiento) están intrínsecamente vinculadas y no son separables. No se puede ignorar la complejidad de los métodos de control del sistema (conductuales) y la evolución del entorno externo que interactúa con el sistema (adaptativo). Estos cambios son producto de las interacciones ordenador/red y del mayor uso de datos digitales en las funciones de control, junto con la naturaleza conectada del entorno externo y de los usuarios.

A modo de ejemplo, la Figura 4 muestra una visión tanto funcional como estructural de los sistemas altamente ciberfísicos que emergen en los sistemas de transporte actuales. En la práctica, estos sistemas suelen construirse a partir de hardware programable de propósito general, mientras que los comportamientos se programan mediante software. Su función en el sistema global puede evolucionar gradualmente con el tiempo, como estamos viendo hoy en día con la automatización de vehículos. Los mecanismos que determinan las cualidades de estos sistemas, como la seguridad, la adaptabilidad o la resiliencia, están basados de forma intencionada en los diseños del hardware y del software, pero también se logran mediante la inversión tanto en la estructura/función del sistema actual como en la evolución de la arquitectura a lo largo del tiempo. Este es un concepto conocido como deuda técnica, donde las decisiones tomadas durante el diseño limitan gradualmente la capacidad del sistema para desarrollar nuevas capacidades en el futuro. Por lo tanto, es fundamental que las decisiones que determinan las cualidades inmediatas y a largo plazo de los sistemas se encuentren bien fundamentadas.

La complejidad de estas redes de CPS da lugar a comportamientos emergentes que no pueden modelarse ni predecirse por completo, ni siquiera descomponerse estructuralmente de una forma tradicional. La tecnología



Figura 4. El automóvil como CPS altamente conectado [2].

Fuente: <https://telecom.economictimes.indiatimes.com/tele-talk/connected-car-opportunity-propels-multi-billion-dollar-turf-war/2971>

empleada tradicionalmente en la Ingeniería de sistemas también está evolucionando hacia la ingeniería digital y hacia prácticas de diseño basadas en simulación. El tiempo invertido en la simulación digital inicial de estos comportamientos emergentes es esencial para la toma de decisiones de diseño, tanto a corto como a largo plazo. Estas simulaciones también se pueden emplear para monitorizar los comportamientos del CPS después de la puesta en servicio para detectar de manera más rápida y precisa, durante el uso del sistema, comportamientos no deseados. Esta es la base y el concepto que subyace detrás de los gemelos digitales, que se tratará de forma amplia en los capítulos 5 y 6. En este paradigma, tanto el sistema real como el sistema virtual y los productos interconectados se planifican y gestionan de manera integral con ciclos de vida completos de ingeniería de sistemas. Como puede verse, las preocupaciones de un ingeniero de sistemas de hoy en día se expanden en este contexto desde la estructura, el comportamiento y las cualidades no funcionales impulsadas por el dominio del sistema en sí, para incluir también su interacción con un contexto y un entorno conectados más amplios, y finalmente a cómo se gestionan y evolucionan tanto el sistema real como sus gemelos digitales durante todo el ciclo de vida. Esto se ha convertido en un desafío multidisciplinar que requiere un cambio en las habilidades de los ingenieros de sistemas para integrar los principios, fundamentos y características enumerados en la Tabla 1 de una forma holística.

Las preocupaciones de la Ingeniería de sistemas de los CPS también pueden expresarse en términos de fiabilidad, relacionadas con la capacidad del CPS para resistir ante la inestabilidad, condiciones inesperadas y regresar oportunamente a un rendimiento predecible pero posiblemente degradado [11]. Esto es un verdadero problema en los CPS altamente conectados. Estas características de fiabilidad, que incluyen confianza, seguridad (física y de la información), privacidad y resiliencia, han evolucionado en su mayor parte dentro de distintos silos, tanto de disciplinas del conocimiento como educativos. Históricamente, la práctica de la ingeniería de sistemas las ha tratado como subdisciplinas dispares. Los grandes proyectos de Ingeniería de sistemas y de integración tienen a menudo líderes específicos de cada disciplina de conocimiento, que aportan puntos de vista discretos dentro del proceso de definición de soluciones de compromiso supervisado por el integrador/ingeniero jefe de sistemas. Los requisitos funcionales han provocado con frecuencia que los ingenieros y los diseñadores prioricen cada disciplina de manera diferente, aislada, según los requisitos y perspectivas específicos de cada dominio (por ejemplo, energía, fabricación, transporte, etc.). Conseguir un cierto nivel de éxito en cada una de las disciplinas suele ser vital para el éxito general del sistema. Las tendencias en la ingeniería de los CPS altamente conectados sugieren que las disciplinas de ingeniería de sistemas están convergiendo hacia una mayor interdependencia. Esto es especialmente importante en los CPS altamente conectados, en los que el pensamiento holístico basado en sistemas es fundamental para sustentar los objetivos de fiabilidad y evitar que surjan problemas con respecto a una disciplina, o que las protecciones establecidas para abordar una disciplina no comprometan otros objetivos primarios del sistema o causen efectos nocivos no deseados [17].

Principios de los CPS	Fundamentos de los CPS	Características de los CPS
Comunicaciones y uso de redes	Conceptos básicos de computación, incluyendo ingeniería software	Seguridad y privacidad
Sistemas empotrados hardware y software	Matemática continua y discreta	Matemática continua y discreta
Sistemas en tiempo real	Computación en el plano físico, incluyendo sensores, actuadores y sistema de control en tiempo real	Interoperabilidad
Computación en el plano físico, seguridad física, fiabilidad, seguridad de la información, prestaciones y gestión de riesgos	Aplicaciones transversales de sensorización, actuación, control, comunicación y computación	Fiabilidad, seguridad, estabilidad y prestaciones
Interacciones de los humanos con los CPS, incluyendo la facilidad de uso	Modelado sistemas heterogéneos y dinámicos de control, computación y comunicación	Factores humanos y usabilidad
	Desarrollo de CPS (con énfasis en los conceptos de resiliencia y seguridad física, pruebas y verificación)	Gestión de consumos y energía

Tabla 1. Principios, características y fundamentos de los CPS [18]

Por tanto, hoy nos encontramos ante un desafío multidisciplinar que asocia disciplinas fundamentales de varias escuelas, como la ingeniería, la informática y la interacción humana. También tenemos un desafío multidisciplinar en las subdisciplinas tradicionales de la ingeniería de sistemas relacionadas con (1) la fiabilidad de los sistemas informáticos (disponibilidad, fiabilidad, seguridad, integridad y mantenibilidad), (2) la ingeniería de seguridad de la información de los sistemas (confidencialidad, integridad y disponibilidad), (3) los sistemas de información (gestión, comunicaciones y privacidad), y (4) la ciberseguridad (amenazas y protección). Esto también supone un reto desde la perspectiva de la gobernanza de los sistemas, tanto en los sistemas empresariales que desarrollan CPS altamente conectados como en los comportamientos emergentes de los propios CPS [17]. Existe la necesidad de un enfoque más interdisciplinar para el diseño de los sistemas, basado en un modelado y simulación funcional del sistema rigurosos, un diseño evolutivo y una cruda evaluación. A medida que la ID y la ingeniería de sistemas basada en modelos (MBSE) se vuelven más prevalentes, existe la posibilidad de transformar los procesos tradicionales de diseño y evaluación de sistemas en formas más holísticas y más basadas en evidencias utilizando modelos (que se desarrollarán más

detalladamente en los Capítulos 4 a 6). La verificación y validación de CPS altamente conectados mediante pruebas y evaluación han sido históricamente una regla de oro, pero resultan notablemente costosas y presentan un creciente nivel de dificultad a medida que los sistemas se vuelven más complejos, más expansivos y más interdependientes de otros sistemas para proporcionar sus capacidades. Nuevamente, incorporar datos y modelos a este proceso tiene como objetivo aliviar parte del gasto y hacer que todo el proceso sea más flexible y se encuentre abierto a los cambios que pueden ocurrir a lo largo del ciclo de vida de un sistema.

2.3. Sistemas basados en el aprendizaje y HMT

Los conceptos de sistemas altamente adaptados, altamente interconectados y la gobernanza distribuida resultante están situando a la ingeniería de sistemas en medio de una transformación digital impulsada por herramientas de modelado avanzadas, la integración de datos y los gemelos digitales resultantes. Los datos, los modelos y los sistemas informáticos están convergiendo con técnicas de aprendizaje automático (ML) para dar lugar a una nueva clase de sistema informático que logra su función

y sus prestaciones mediante el uso de ML, y que se conoce como sistema basado en el aprendizaje (LBS, *Learning Based System*). Las aplicaciones de los LBS están evolucionando exponencialmente en muchos ámbitos.

Los LBS son tipos de modelos de IA que utilizan algoritmos de ML para tomar decisiones y resolver problemas sin estar programados explícitamente. Los algoritmos de aprendizaje automático detectan, analizan y aprenden patrones de forma independiente directamente a partir de los datos de entrada y modifican su comportamiento en consecuencia para predecir nuevos resultados. Los LBS se diferencian de los sistemas de IA basados en reglas por su capacidad para ser escalables, adaptables y gestionar tareas complejas. Los sistemas basados en reglas son un tipo de modelo de IA que utiliza un conjunto de reglas escritas previamente para tomar decisiones y resolver problemas. Los desarrolladores crean reglas basadas en el conocimiento humano experto que permiten al sistema procesar los datos de entrada y producir un resultado. Casi todos los sistemas del futuro, y las herramientas que utilizamos para diseñarlos, están evolucionando para integrar diferentes elementos, tanto basados en reglas como en el aprendizaje.

El número de sistemas que poseen algún nivel de capacidad de aprendizaje se encuentra actualmente experimentando un aumento exponencial. En algunas aplicaciones estos sistemas trabajan en estrecha colaboración con personas; en otras, las operaciones son en gran medida autónomas. En estos sistemas, equipos distribuidos de personas trabajan con equipos distribuidos de sistemas autónomos, con relaciones que pueden cambiar dinámicamente. Asimismo, la ingeniería de sistemas está evolucionando hacia una disciplina más tecnológica que utiliza los LBS para definir y gestionar datos digitales y modelos descriptivos que vinculan diferentes disciplinas. La transformación digital de la ingeniería de sistemas está dando lugar a avances que la están transformando al emplear tecnologías de inteligencia artificial y de aprendizaje automático para la automatización de muchas tareas de ingeniería, con la finalidad de dotar de nuevas capacidades a la inteligencia humana. Al mismo tiempo, la aplicación de la IA, el aprendizaje automático y la autonomía a muchos de los sistemas complejos y críticos impulsa la necesidad de nuevos métodos, procesos y herramientas de ingeniería de sistemas.

Un objetivo principal de la ingeniería de sistemas es garantizar que el comportamiento y las prestaciones de los sistemas complejos cumplan con los resultados esperados impulsados por las necesidades del usuario, y que la configuración del sistema se gestione durante toda su vida útil. Los avances en las aplicaciones de IA y ML permiten que los futuros componentes del sistema puedan aprender y adaptarse más rápidamente, y que el comportamiento y el rendimiento puedan ser no deterministas, con resultados menos predecibles, pero manejables. Este es el principal desafío de los LBS. La incapacidad de poder validar explícitamente los comportamientos del sistema, o el tiempo que lleva hacerlo, afecta a la confianza en estos sistemas y

está impulsando un cambio en la forma en que los ingenieros de sistemas abordan las actividades de validación [26]. La incertidumbre presente en muchos componentes de IA/ML que interactúan entre sí desafía los métodos tradicionales de descomposición utilizados por los ingenieros de sistemas, lo que requiere nuevos métodos de síntesis. Finalmente, a medida que los sistemas desarrollen medios para el coaprendizaje entre personas y máquinas, se hace necesario revisar los modelos tradicionales que separan los comportamientos humanos de los de las máquinas [14].

Por tanto, el principal desafío emergente de la ingeniería de sistemas es producir sistemas que generen valor en base a su capacidad de aprender, la cual es inherentemente no determinista, pero siendo a la vez robustos de forma apropiada, predecibles y fiables en el tipo de usos críticos y complejos en los que actualmente se aplica la ingeniería de sistemas. Esto incluye comportamientos tanto humanos como de las máquinas en entornos de decisión conjunta, que dependen en gran medida del buen diseño de los sistemas humanos y de la presentación de la información para la toma de decisiones. También incluye la adaptación de los procesos de prueba y evaluación a entornos de coaprendizaje [14].

El futuro desafío de la ingeniería de sistemas comprende a LBS que realmente se adapten y aprendan dinámicamente de sus entornos. Estos entornos pueden ser reales, simulados o una combinación de ambos gracias a la generación de datos sintéticos que, por un lado, ofrece nuevos enfoques para el entrenamiento y desarrollo de sistemas, pero, por otro, genera la necesidad de validación de los datos sintéticos. En este futuro que emerge con rapidez, la confianza de las personas en las máquinas (y tal vez también de las máquinas en las personas) será fundamental. Se espera que los sistemas aprendan a modificar, o a crear, nuevos comportamientos a medida que cambie el contexto y esto puede suceder con bastante rapidez. Los métodos para validar de nuevo las prestaciones del sistema de manera extremadamente rápida o “sobre la marcha” no forman parte del conjunto de prácticas actuales de la ingeniería de sistemas y deben desarrollarse junto con este tipo de sistemas de aprendizaje [14].

El futuro, tanto de los LBS como de la propia ingeniería de sistemas, es difícil de predecir en este momento. La ingeniería de sistemas está entrando en un período de cambios rápidos en el que intentará ponerse al día con el uso cambiante de los algoritmos de ML y los LBS resultantes. La complejidad de los futuros sistemas de ingeniería aumentará rápidamente, pero las herramientas automatizadas que utilizan los ingenieros de sistemas también evolucionarán para gestionar esa complejidad, probablemente a un ritmo más lento. Por tanto, resulta realmente urgente transformar la ingeniería de sistemas desde flujos de trabajo manuales basados en papel hacia un conjunto de prácticas totalmente digitales y basadas en modelos.

2.4. Gobernanza distribuida en sistemas de sistemas

La característica distintiva de un sistema de sistemas (SoS, *System of Systems*) es que los comportamientos del “todo” provienen de sistemas constituyentes individuales que actúan de forma independiente y autónoma [3]. Además, los SoS son generalmente sistemas socio-técnicos: sistemas impulsados por la tecnología que implican una participación humana y social significativa, donde esa participación a su vez influye en la arquitectura y el diseño del sistema técnico [13]. En otras palabras, el ser humano es parte del sistema, no un agente externo.

Para determinar unos principios de arquitectura apropiados, la literatura de ingeniería de SoS (SoSE, *Systems of Systems Engineering*) define un sistema de clasificación para los SoS vinculado al grado de control en su gestión. Las cuatro clases de SoS son: dirigido, colaborativo, virtual y reconocido. El grado de control central o gobernanza sobre los cambios del SoS es la distinción principal de cada clase [4]. En un SoS dirigido, éste se crea para cumplir un propósito específico y se gobierna de forma centralizada. Los sistemas constituyentes siguen siendo independientes, pero sus operaciones están dirigidas a través de una autoridad central. En un SoS colaborativo, los sistemas constituyentes se ofrecen voluntariamente a colaborar y no existe un agente de control central con autoridad para dirigir sus operaciones. Sin embargo, todavía existe una función de gobernanza que determina un propósito acordado para el SoS. En el SoS colaborativo, los estándares, regulaciones, normas y circunstancias impulsan el funcionamiento de los sistemas constituyentes.

La literatura sobre SoS también distingue las clasificaciones colaborativas de SoS como virtuales y reconocidas. Los SoS virtuales son colaborativos, pero se desarrollan sin un propósito o un conjunto de objetivos acordado de forma centralizada. La gobernanza se encuentra completamente distribuida. Los SoS reconocidos tienen un conjunto de objetivos reconocido y un componente de gestión designado, tienen recursos asignados para su desarrollo, pero los cambios se basan en objetivos de colaboración. La gobernanza sigue estando distribuida, pero se ajusta a políticas y resultados determinados centralmente. Un buen ejemplo de SoS colaborativo es el tráfico de vehículos en las autopistas interestatales. El hecho de que este SoS de tráfico en carretera se considere virtual o reconocido depende de la perspectiva de las partes interesadas. Los conductores lo verían como virtual (sus objetivos están determinados por su necesidad individual de ir del punto a al punto b), mientras que los funcionarios responsables del transporte por carretera lo verían como algo reconocido (sus objetivos son gestionar de forma segura y eficiente el tráfico interestatal). Los sistemas adaptables y altamente interconectados, actuales y futuros, serán más colaborativos y lucharán contra el control administrativo, y las perspectivas y comportamientos individuales afectarán a los comportamientos del sistema de maneras que no pueden predecirse fácilmente mediante la descomposición tradicionalmente empleada por la ingeniería de sistemas.

El modelo en Ola es un marco establecido para evaluar y planificar la evolución de sistemas de sistemas (ref. Figura 5). El modelo reconoce que la evolución está impulsada continuamente por aportaciones del contexto externo y, a diferencia de la ingeniería de sistemas tradicional, considera al análisis del cambio del sistema como un proceso continuo

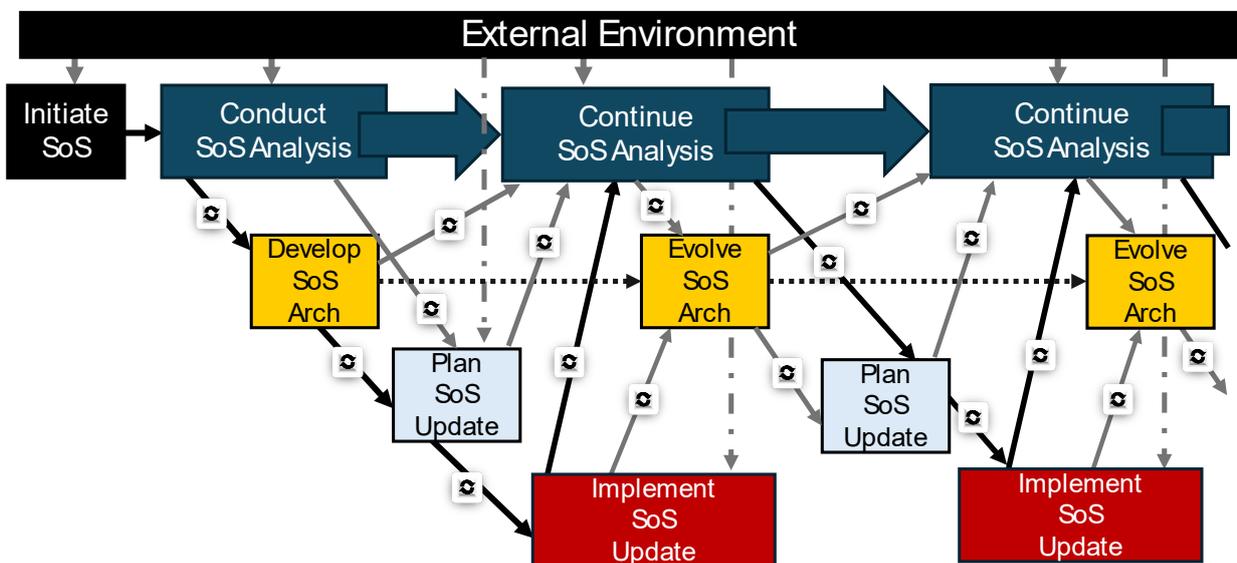


Figura 5. Modelo en Ola [7]

con múltiples incrementos superpuestos. En el modelo en Ola la evolución del sistema es un proceso prospectivo con realimentación en cada iteración, y las estrategias de control de gestión intentan agrupar múltiples cambios constituyentes en cambios de la arquitectura a nivel del SoS para generar eficiencia en el proceso de prueba y validación [7].

En la literatura sobre innovación se aporta una perspectiva diferente, llamada “gestión de la transición” [21]. La literatura sobre innovación contrarresta el modelo en Ola con una visión más ascendente de la evolución del sistema. Los modelos de sistemas de innovación reconocen la innovación como un proceso adaptativo complejo donde las innovaciones de nivel inferior en los sistemas constituyentes forman nichos de adopción, que con el tiempo producen cambios más amplios en los regímenes de los SoS establecidos, lo que finalmente resulta en una transformación del entorno (o contexto) existente. Hoy en día se puede ver el desarrollo de tecnologías de vehículos sin conductor como una evolución “en proceso”. El aspecto principal de este modelo es que la innovación progresa a través de capas sociales y puede modelarse como un fenómeno social de múltiples escalas o capas, a diferencia de la visión más mecanicista del modelo en Ola. La Figura 6 muestra este proceso tal como se refleja en la literatura sobre gestión de la transición.

Los fenómenos que compiten en estas dos visiones de la evolución de los SoS son modelos diferentes de gobernanza distribuida. La perspectiva de la IS en el modelo de ola supone que el cambio del SoS puede planificarse y gestionarse a lo largo del tiempo mediante algún mecanismo de gobernanza. La perspectiva de la innovación en el modelo de transición tecnológica señala que el cambio se producirá a partir de una evolución tecnológica de abajo hacia arriba que no puede gobernarse, pero sí fomentarse o guiarse. En los sistemas de defensa, la gobernanza distribuida del SoS y de los sistemas constituyentes comprende un conjunto de procesos de acuerdo entre organizaciones militares. En la innovación comercial, la gobernanza distribuida está determinada por factores de mercado. En este sentido, un desafío actual de la IS se deriva del mercado comercial que impulsa la innovación tecnológica y de las industrias aeroespacial y de defensa que impulsan principalmente los métodos y herramientas de IS de forma algo aislada respecto a ese mercado más grande.

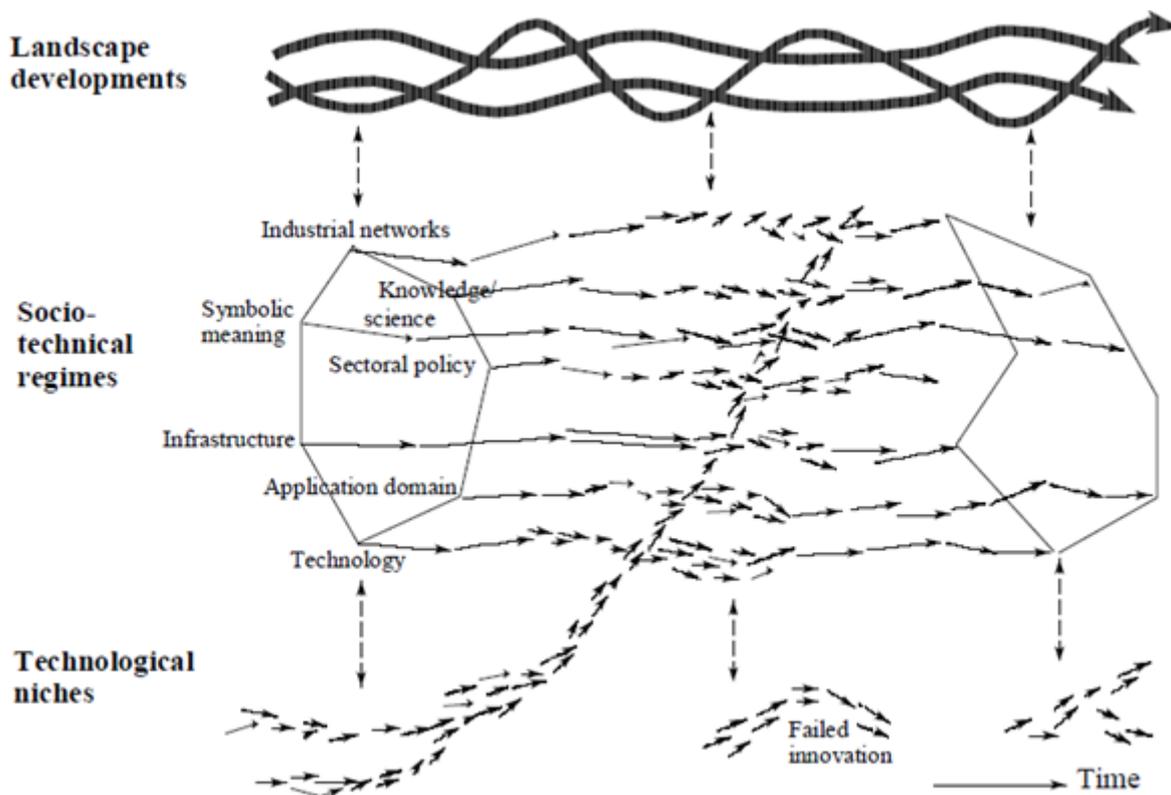
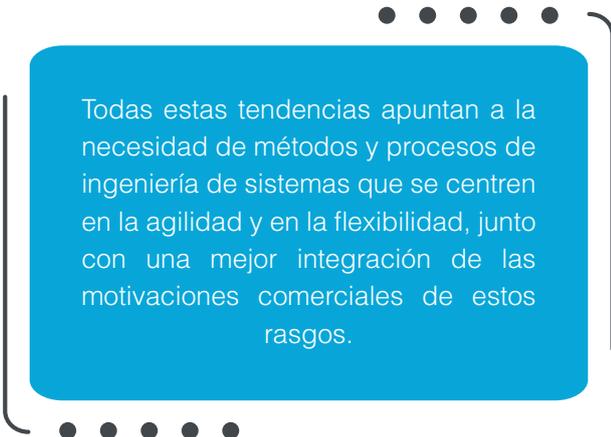


Figura 6. Perspectiva dinámica multinivel de las transiciones tecnológicas en un sistema de sistemas [8]

Esto genera la preocupación de que los métodos actuales de ingeniería de sistemas sean demasiado lentos en su cambio y no se adapten bien a la dinámica actual de gobernanza distribuida que se da en los sistemas más adaptables y altamente interconectados. Intentar predecir completamente los requisitos de un SoS emergente y planificar rígidamente y por adelantado sus actualizaciones van en contra de los mercados comerciales actuales. Las empresas comerciales apuntan a centralizar la gobernanza del valor del SoS, pero reconocen que las capacidades emergentes reales de los SoS requieren apertura y experimentación antes que una planificación rígida.

La tecnología también está impulsando el empleo de modelos de gobernanza distribuida. Tecnologías recientes como la orquestación de software y la tecnología blockchain, o tecnología de contabilidad distribuida (DLT, *Distributed Ledger Technology*), han dado lugar a la aparición de “organizaciones sin líderes” donde las reglas y los objetivos del sistema se distribuyen a través de la automatización del software y las personas pueden auto-organizar sus tareas. Es probable que estas mismas tecnologías ayuden a gestionar “datos y modelos con autoridad” en la futura ingeniería de sistemas, pero su gobernanza es una cuestión abierta hasta el momento. Ésta no es sólo una tendencia comercial; los conceptos emergentes de mando y control militar visualizan los SoS militares del futuro como una Internet de las cosas, idealmente pasando de plataformas rígidamente conectadas a distribuciones más flexibles de capacidades gestionadas por centros de mando y control distribuidos en línea con el concepto de combate colaborativo. Los mismos conceptos de sistemas altamente adaptables, de CPS altamente interconectados y de LBS/HMT emergen juntos para transformar las operaciones militares.



Todas estas tendencias apuntan a la necesidad de métodos y procesos de ingeniería de sistemas que se centren en la agilidad y en la flexibilidad, junto con una mejor integración de las motivaciones comerciales de estos rasgos.

3. TENDENCIAS EN LA EVOLUCIÓN DE LA INGENIERÍA DE SISTEMAS PARA SU APLICACIÓN EFICAZ A ESTOS TIPOS DE SISTEMAS

3.1. Digitalización

El principal cambio que impulsa la disciplina de ingeniería de sistemas hoy en día es la transición hacia la ingeniería digital, un paradigma en el que los “datos compartidos y gestionados con autoridad” se pueden transformar a través de “modelos compartidos y gestionados con autoridad”, y las herramientas asociadas, para crear artefactos digitales que pueden ser empleados por varios responsables de la toma de decisiones y otras personas que necesiten acceso a través de medios digitales al diseño y a las descripciones del sistema durante toda su vida útil [21]. Inicialmente, estos artefactos casi siempre eran documentos en papel o dibujos. Hoy en día, generalmente, se basan en tecnologías digitales, pero los flujos de trabajo todavía tienden a basarse en el empleo de documentos en lugar de sustentarse en la gestión de datos.

La vista del flujo de trabajo representada en la Figura 7 muestra conceptualmente cómo los datos compartidos y gestionados con autoridad deben transformarse en artefactos digitales en diferentes etapas del ciclo de vida en cualquier proceso de ingeniería de sistemas. Los datos, los modelos de datos federados y el almacenamiento de datos distribuidos son la infraestructura fundamental de la ingeniería de sistemas de hoy en día. Se trata de una federación distribuida de datos y modelos, gobernada entre organizaciones, con procesos autorizados para gestionar tanto los datos y como el origen y la configuración de los modelos.

La Figura 7 es especialmente relevante en la modernización de la ingeniería de sistemas, ya que la “gestión de datos” no está actualmente definida como un proceso en los estándares de la ingeniería de sistemas. Los datos, los modelos y los sistemas de almacenamiento de datos pueden concebirse y tratarse como un sistema separado que también debe desarrollarse e implementarse en apoyo del sistema principal. Éstos deben definirse y construirse junto con otros aspectos del desarrollo del sistema, y también tienen sus propios ciclos de vida. Los nuevos

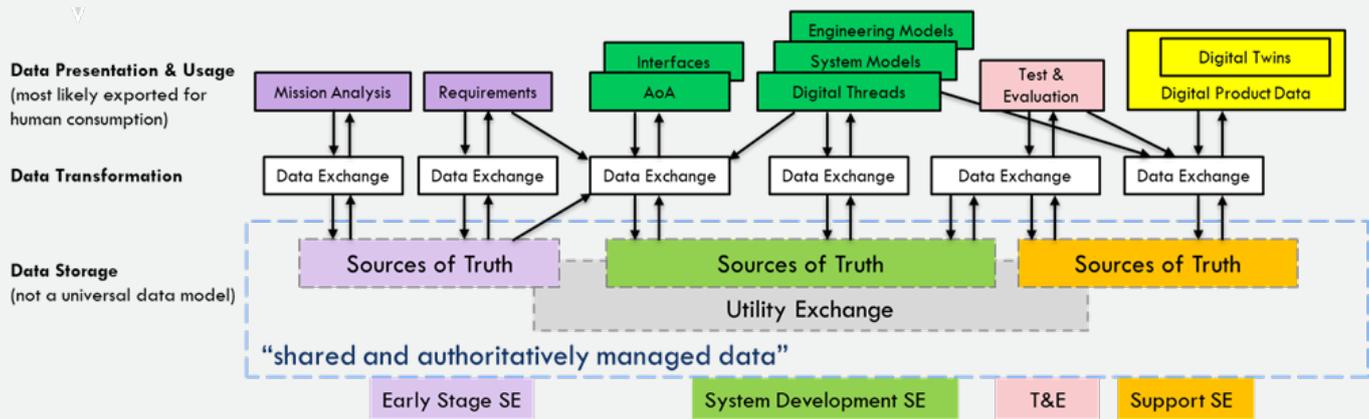


Figura 7. Transformación de datos en el ciclo de vida

procesos del ciclo de vida de la ingeniería de sistemas están evolucionando para abordar conjuntos de datos y de modelos digitales compartidos y gestionados con autoridad asociados a todo el ciclo de vida del sistema, no solo a los ciclos de vida de programas o de ingeniería individuales. Este modelo mental se ha representado como una combinación de un modelo en V convencional con un modelo en V reflejado para crear un diamante de diseño que incorpora las contrapartes digitales de un producto en todas las etapas de su ciclo de vida (ref. Figura 8) [5]. Se prevé que el sistema virtual y el sistema real evolucionen juntos a largo plazo durante todo su ciclo de vida.

Los ingenieros de sistemas han utilizado durante mucho tiempo datos digitales y diversas herramientas de modelado y de análisis para producir artefactos digitales para la toma de decisiones (como diapositivas de Microsoft PowerPoint). Sin embargo, los modelos de datos subyacentes no se han compartido de una manera “fluida” a través de herramientas y de flujos de trabajo digitales o, probablemente, no se han compartido en absoluto. Además, la autoridad sobre esos datos a menudo ha estado sustentada sobre actividades independientes generalmente organizadas por disciplinas de conocimiento.

Hoy en día, gran parte de la transformación de datos en modelos y en decisiones sigue siendo una interpretación manual de datos y un conjunto de análisis dispares. Esta interpretación manual limita la capacidad para ser iterativos y receptivos en todas las disciplinas de conocimiento y en las herramientas asociadas a las mismas. Resulta ineficiente e incompleto. La evolución de la ingeniería de sistemas es un flujo de trabajo iterativo totalmente integrado donde todo se centra en el sistema, no en el propietario de los datos o en un elemento concreto de un diseño. El principal desafío actual de la ingeniería digital no es tanto estar “basado en

modelos”, sino en comprender, crear y validar el modelo de datos subyacente a través del cual se integran los requisitos, el diseño, las pruebas, las diferentes disciplinas de conocimiento y los procesos, de forma que dicho modelo pueda ser compartible y compartido. El valor de la ingeniería de sistemas en el futuro se puede conseguir a través de una transferencia más fluida y más eficiente de los datos y de los modelos, comenzando desde las prestaciones clave subyacentes a través de los modelos, hasta las decisiones y la facilidad para poder profundizar desde las decisiones hasta los datos. Este no es actualmente el estado de la práctica de la ingeniería de sistemas, sino un objetivo a tener en cuenta a la hora de definir hojas de ruta para el desarrollo de capacidades.

3.2. El ecosistema digital

La colaboración entre disciplinas de conocimiento y entre organizaciones requiere establecer interfaces muy fiables entre múltiples aplicaciones de ingeniería, de desarrollo y de gestión en un ecosistema de ingeniería digital. El futuro ecosistema digital para la ingeniería de sistemas se concibe como una arquitectura orientada a servicios que proporciona flexibilidad y adaptabilidad entre diferentes herramientas y entre organizaciones, como se muestra en la Figura 9. El intercambio de datos entre aplicaciones puede concebirse técnicamente como un conjunto de API (*Application Program Interfaces*) colaborativas para el intercambio de datos y de información, así como de consultas y de respuestas. Es necesario madurar estándares, avanzar hacia sistemas abiertos y adquirir experiencia en la implementación de la ingeniería de sistemas digital para facilitar la adopción de la ingeniería digital en las organizaciones [16].

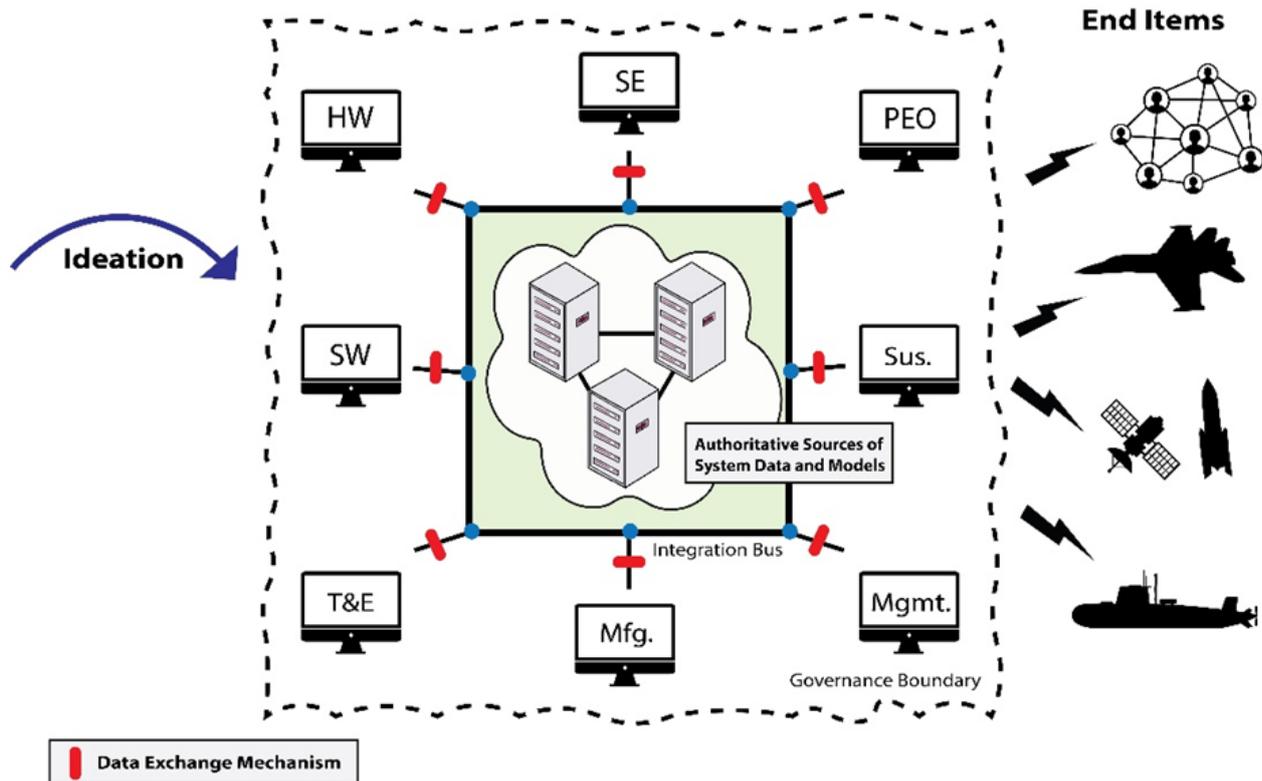


Figura 9. Vista operacional de un ecosistema digital.
 (SE: systems engineering, HW: hardware, SW: software, T&E: test & evaluation, Mfg: manufacturing, Mgmt: management, Sus: sustainment, PEO: program executives)

El uso de un modelo digital del sistema debería acabar siendo aplicado a todos los sistemas en todos los dominios, pero actualmente este concepto se encuentra aún en un estado incipiente. La mayoría de los sistemas todavía se desarrollan e implementan sin este conjunto formal, pero integral, de modelos y de vistas, lo que lleva a errores y re-trabajos continuos en las últimas etapas del ciclo de vida. A medida que los nuevos tipos de sistemas se vuelvan cada vez más adaptables, interconectados, distribuidos y basados en el aprendizaje, la importancia de MBSE debería ganar valor. Si bien aún queda camino por recorrer en términos de madurez de las capacidades aportadas por MBSE, el inicio de esa transición ya resulta factible y valioso.

3.4. Modelos de datos, ontologías y tecnología web semántica

A medida que crece la base digital de la ingeniería de sistemas, cada vez se depende más de datos definidos y estructurados (modelado de datos). Las prácticas actuales de la ingeniería de sistemas incluyen lenguajes utilizables por máquinas y el desarrollo de taxonomías, diccionarios y

ontologías de datos para permitir la interoperabilidad para el tratamiento computarizado de la síntesis, el análisis y la exploración de datos. La conectividad a lo largo de todo el modelo digital del sistema comienza en la capa de datos. El desarrollo y el mantenimiento de un modelo y de un entorno de datos empresariales se ha convertido en parte del trabajo de la ingeniería de sistemas. El modelo de datos define específicamente la estructura y las relaciones de los elementos de datos que alimentan los niveles superiores de los modelos y de los análisis del sistema. Una ontología digital es un marco reutilizable que modela datos generalizados: objetos generales que tienen propiedades comunes y entidades individuales no especificadas. Un modelo de datos que se basa en una ontología digital proporciona un mayor nivel de interoperabilidad y de reutilización de los datos.

Un informe reciente de un taller del SERC (Systems Engineering Research Center) sobre ontologías digitales resumió tanto la necesidad como el valor aportado por el desarrollo eficaz de ontologías digitales como uno de los medios para la conseguir la interoperabilidad digital [25]. Las aplicaciones tradicionales de la ingeniería de sistemas se basan principalmente en documentos para capturar e

intercambiar información. Los documentos no favorecen la interoperabilidad, ni la eficiencia, ni la automatización, dada su falta de formalismo tanto en la sintaxis como en la semántica. Los métodos basados en ontologías permiten hacer inferencias sobre datos e información para determinar nuevos hechos y descubrir lagunas y relaciones nunca antes establecidas. También permiten identificar de forma única elementos de datos para que diferentes sistemas puedan hacer referencia a los mismos conceptos sin tener que pasar por alto o duplicar estructuras de datos difíciles de manejar. Estas ontologías son un medio pragmático para describir formalmente las entidades y las relaciones relevantes en un sistema, implementar esas descripciones y relaciones y ser útiles para nuestros propósitos y nuestros objetivos específicos. Por ejemplo, las Ontologías Militares Básicas Comunes representan datos de diferentes ámbitos militares en una estructura común que permite compartir y agregar datos de cada ámbito militar utilizando representaciones de datos genéricas que pueden ser reutilizadas. La ontología común permite la interoperabilidad de datos en operaciones conjuntas. En ingeniería de sistemas, la interoperabilidad sintáctica entre los requisitos formales, el diseño y el lenguaje de pruebas es vital, pero no suficiente. La interoperabilidad de los datos es necesaria para permitir que la infraestructura digital intercambie y agregue información sobre los requisitos, el diseño y los análisis. El valor de las ontologías radica en lo que nos permiten comunicar, y la comunicación multidisciplinaria es el núcleo tanto de la ingeniería de sistemas como de la ingeniería digital.

La comunidad de ingeniería de sistemas necesita, aunque aún no tiene, una comprensión amplia de los diferentes métodos de modelado de datos ontológicos y relacionados que se requieren para garantizar el intercambio de datos y la interoperabilidad de los mismos. La comunidad de ingeniería de sistemas está preparada para ampliar su formación y su capacitación para incluir aquellos fundamentos de la arquitectura de datos, el modelado de datos y la ciencia de datos específicamente relevantes para los ingenieros de sistemas del mañana.

3.5. Agilidad

Se espera que las transformaciones de datos desde y hacia las federaciones de datos y de modelos compartidos y gestionados con autoridad descritas en el apartado anterior se produzcan de forma iterativa y continua a lo largo de toda la vida útil de un sistema. Estos datos y estos modelos pueden originarse en cualquier fase del ciclo de vida de un sistema y en cualquier función asociada con la ingeniería y con la gestión. Para hacer frente a este nuevo contexto, las prácticas de la ingeniería de sistemas se están volviendo más ágiles y más rápidas en su respuesta. Aumentar la capacidad de respuesta no significa eliminar los procesos

críticos de la ingeniería de sistemas, sino simplemente aumentar el número de iteraciones y acortar el tiempo del ciclo entre ellas. Un taller reciente sobre prácticas ágiles en sistemas con uso intensivo de hardware identificó una serie de temas que deben adoptarse mejor en la práctica de la ingeniería de sistemas [1].

La industria comercial ha adoptado prácticas ágiles en el software, en sistemas hardware/software y en servicios para abordar unas amenazas y unas oportunidades que cambian rápidamente en sus ámbitos de negocio. Un objetivo principal de las prácticas ágiles de la ingeniería de sistemas es trasladar el aprendizaje lo más pronto posible dentro del proceso de desarrollo, es decir, perseguir desde las primeras etapas de diseño de los sistemas el dar cabida a la innovación en las etapas posteriores de desarrollo. Las prácticas ágiles potenciadas por el empleo de modelos digitales, de prototipos y de infraestructuras de prueba ayudan a impulsar el aprendizaje, reducir los riesgos de integración y, como resultado, generar una mayor flexibilidad en la toma de decisiones de diseño a largo plazo. Incidir en esta estrategia a través de una elección apropiada de los sistemas, los subsistemas o los elementos del sistema ayuda a anticipar la evolución de las partes del sistema que se encuentran más expuestas al cambio o de aquellas que aporten una mayor mejora de las prestaciones, cuando se introduce en ellas un cambio innovador. Áreas críticas para la ingeniería de sistemas, como la ingeniería de sistemas software, las tecnologías de la información, la arquitectura empresarial, el modelado y la simulación distribuidos y los sistemas de fabricación automatizados son elementos que habilitan esta transición.

Las premisas subyacentes a las prácticas ágiles incluyen la colaboración directa entre los usuarios y los desarrolladores, el fomento de la simplicidad y la creación de un flujo continuo de valor. En los grandes sistemas de defensa, por ejemplo, el flujo desde la necesidad operativa hasta la generación de una capacidad pasa por muchas organizaciones y por muchos procesos antes de convertirse en un programa (de cualquier tipo). Esto acaba por introducir cambios en la interpretación de las necesidades y de los requisitos, aísla al cliente real del desarrollo de las capacidades e interrumpe el flujo de trabajo desde la necesidad a la capacidad. Por el contrario, la industria comercial ha sido pionera, en muchas formas diferentes, en determinar las necesidades y las respuestas de los clientes utilizando análisis integrados en los propios productos, acelerando la comprensión del cliente.

De manera similar, las prácticas ágiles desafían la mentalidad de lote único y la creencia de que se debe entender todo antes de la implementación. Esto implica un cambio a nivel empresarial para permitir una entrega más frecuente de sistemas operativos (o de elementos del sistema) mediante la coordinación de los ciclos de desarrollo y de entrega

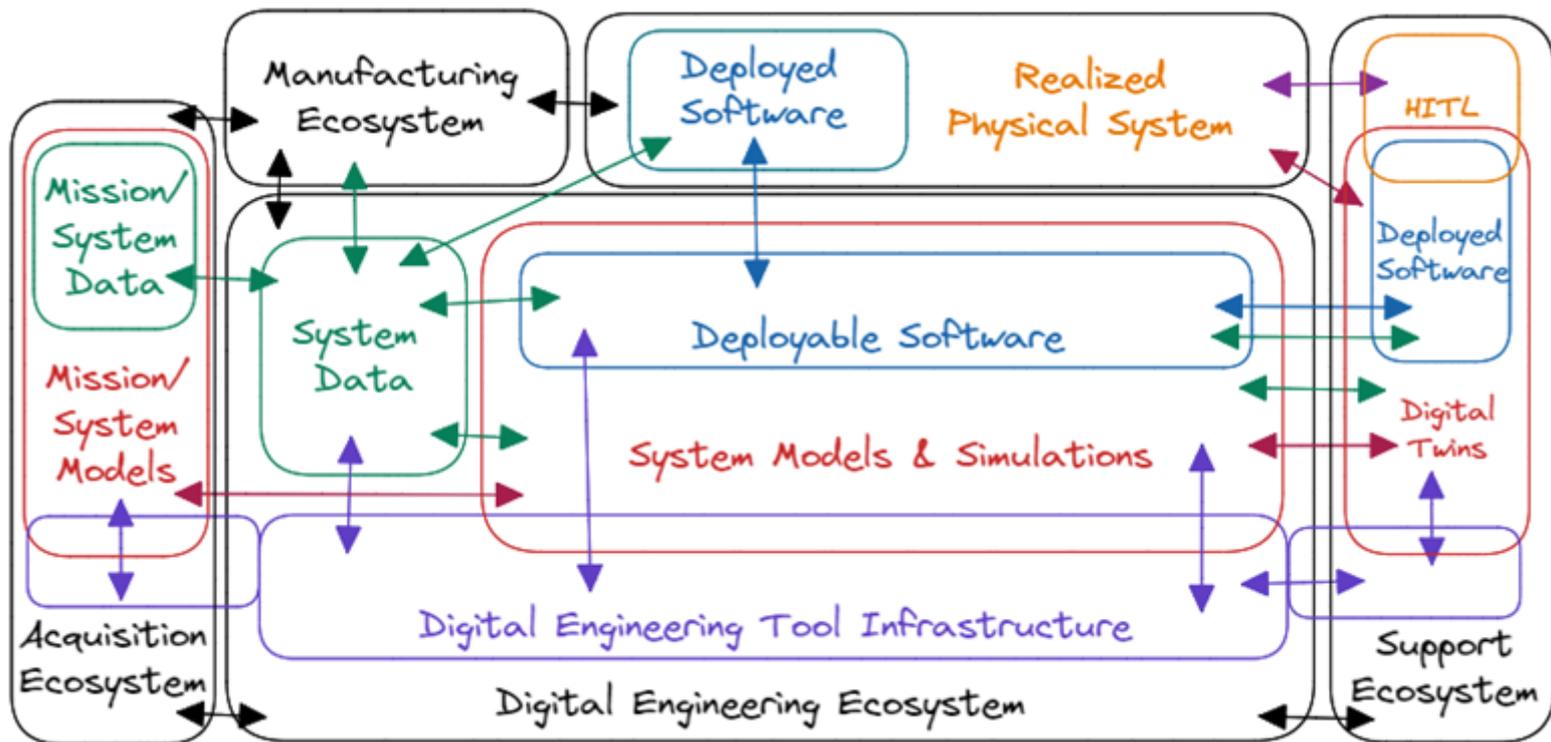


Figura 10. Los sistemas actuales deberían planificar el despliegue de software de forma frecuente en todos los ámbitos de la planificación y del desarrollo de un programa

para conseguir los mejores resultados. En lugar de agravar el efecto de los ciclos más lentos que condicionan el ritmo de entrega a nivel del sistema, una reordenación de los flujos de trabajo puede garantizar la generación de un flujo basado en lotes de trabajo más pequeños. El cumplimiento de hitos sigue siendo importante, pero debe traducirse en una reducción del riesgo en la obtención, no simplemente en el cumplimiento de unos criterios. Integrar un ritmo de trabajo constante y unos objetivos basados en hitos es fundamental para ser ágil en sistemas con uso intensivo de hardware. La significativa evolución de los prototipos desde entornos virtuales a implementaciones físicas y a un uso operativo aporta beneficios tangibles cuando el software se reutiliza de un producto a otro. Los programas deben incorporar software desplegable en sistemas de simulación y de formación, permitiendo a todos los desarrolladores y a los usuarios experimentar el uso operativo del producto, habilitando entornos híbridos de simulación (LVC, *Live Virtual Constructive*). La Figura 10 muestra este enfoque.

Lo que continúa siendo consistente con la práctica actual de la ingeniería de sistemas es una meditada descomposición y división de los elementos que constituyen un sistema. Dividir sistemas complejos en componentes más pequeños y manejables permite un aprendizaje más rápido y una mejor

comprensión de cada uno de los elementos individuales. Las prácticas ágiles aprovechan la modularidad para diseñar sistemas que puedan evolucionar con el tiempo. El control de los interfaces y de los interfaces de programación de aplicaciones (API, *Application Program Interfaces*) es fundamental, tanto para definir el trabajo sobre el sistema como las habilidades del equipo necesarias para ejecutar dicho trabajo. Los enfoques de sistemas abiertos y modulares (MOSA, *Modular Open Systems Approaches*) sientan las bases para un desarrollo ágil, tanto en sistemas software como hardware.

A modo de comentario, esta partición debe considerar la descomposición estructural de los sistemas como un área en el que innovar, ya que seguir ciegamente la descomposición tradicional en subsistemas puede limitar la agilidad.

Finalmente, la práctica ágil en sistemas con uso intensivo en hardware requiere una inversión inicial en actividades de prueba e infraestructura para reducir el riesgo del producto final. Por ejemplo, SpaceX™ considera los fallos en los lanzamientos como una inversión en la recopilación de datos. Su mentalidad de aprendizaje corporativo a partir de múltiples lanzamientos fallidos es un claro ejemplo de

la cultura y la mentalidad necesarias para la innovación y la mejora continua en la práctica ágil [9]. La inversión en herramientas de ingeniería basadas en modelos, en múltiples prototipos a nivel de sistemas y en entornos hardware-in-the-loop (HIL) es fundamental para implementar con éxito el concepto de agilidad en sistemas con uso intensivo de hardware. Esto es más difícil en programas con uso intensivo de hardware que en programas que solo incluyen software, porque las herramientas empleadas en los primeros son más diversas y están menos integradas que en los entornos de desarrollo de software actuales.

3.6. Enfoques de sistemas abiertos y modulares (MOSA)

En sistemas altamente adaptables e interconectados, como se mencionó anteriormente, se necesita perseguir, desde las primeras etapas de diseño de los sistemas, el dar cabida a la innovación en las etapas posteriores de desarrollo. Particularmente en sistemas críticos de gran envergadura, como los sistemas aeroespaciales y los de defensa, las plataformas pueden tener ciclos de vida que abarcan décadas. Incidir en esta estrategia eligiendo de forma apropiada los sistemas, los subsistemas o los elementos del sistema ayuda a anticipar la evolución de las partes del sistema que se encuentran más expuestas al cambio o de aquellas que aporten una mayor mejora de las prestaciones cuando se introduce en ellas un cambio innovador. La elección de la estrategia empresarial asociada es igualmente importante para el éxito del sistema durante su ciclo de vida. Este es el elemento impulsor detrás del Enfoque de Sistemas Abiertos y Modulares (MOSA), un enfoque de diseño que pretende mantener una base técnica y comercial que aporte en el futuro flexibilidad y capacidad de innovación sobre el sistema. La descomposición en capacidades funcionales y en componentes hardware cada vez más pequeños reduce la complejidad a nivel de componente, así como el tamaño de los equipos de trabajo asociados. Sin embargo, la modularidad debe diseñarse y planificarse cuidadosamente como estrategia en la definición de la arquitectura como parte de la planificación del ciclo de vida. Las habilidades para hacer esto con éxito son especializadas y tienen una gran demanda.

La descomposición en módulos más pequeños aumenta la agilidad, pero también aumenta el riesgo de integración. La estandarización de los interfaces, y en particular el uso de interfaces abiertos, puede ayudar a reducir ese riesgo. En el pasado, los estándares abiertos casi siempre se definían mediante documentos, lo que hacía que la interpretación, el cumplimiento y la interoperabilidad real constituyesen un verdadero desafío. Con la introducción del Protocolo de Internet (IP), de las capacidades “*plug and play*” en los productos Microsoft Windows™ y del sistema

operativo Android™, la implementación de estándares abiertos ha sido cada vez más frecuente como el software de código abierto. La adopción de la ingeniería digital y de un modelo digital del sistema facilita la definición formal de los interfaces entre módulos en una línea base digital que elimina la necesidad de interpretación y los errores en la comunicación [22]. En particular para la ingeniería de sistemas, la versión 2 de SysML (System Modeling Language) anticipa la provisión de un lenguaje de modelado central con un interfaz de programación de aplicaciones (API) digital y un conjunto de implementaciones de código abierto.

3.7. Inteligencia Artificial e Ingeniería de Sistemas (AI4SE & SE4AI)

Es difícil predecir a día de hoy exactamente cómo la IA y el ML afectarán a la IS, pero en un futuro próximo la IS experimentará cambios significativos en sus métodos, procesos, herramientas y habilidades como parte de las mega-tendencias de la IA y del ML. En un taller a principios de 2019 sobre el futuro de la IS (FuSE) organizado por INCOSE, los términos IA para IS e IS para IA se utilizaron por primera vez para describir esta transformación dual [15]. Las etiquetas “AI4SE” (*Artificial Intelligence for Systems Engineering*) y “SE4AI” (*Systems Engineering for Artificial Intelligence*) se han convertido rápidamente en metáforas de una rápida fase de evolución que se avecina en la comunidad de la ingeniería de sistemas. AI4SE puede definirse como la aplicación de técnicas de inteligencia aumentada y de aprendizaje automático para apoyar la práctica de la ingeniería de sistemas. Los objetivos de estas aplicaciones incluyen el conseguir escala en la construcción de modelos, la confianza en la exploración en el ámbito del diseño y el dar cobertura a una validación automatizada. SE4AI puede definirse como la aplicación de métodos de ingeniería de sistemas al diseño y operación de sistemas basados en el aprendizaje. Las áreas clave de aplicación de la investigación incluyen el desarrollo de principios para el diseño de sistemas basados en el aprendizaje, de modelos de evolución del ciclo de vida y de métodos de depuración de modelos [14]. Estos se han resumido de las hojas de ruta publicadas por el SERC sobre los impulsores de la IA y de la autonomía en la ingeniería de sistemas [14].

La ingeniería de sistemas y el software son campos que emplean modelos basados en lenguajes que resultan muy propicios para ser automatizados. Hoy en día existen herramientas que utilizan grandes modelos de lenguaje para crear código software. A día de hoy, están empezando a surgir herramientas basadas en inteligencia artificial que crean requisitos de ingeniería de sistemas y modelos descriptivos. Algunas de las tecnologías y de los casos de uso a tener en cuenta incluyen la construcción automatizada de

modelos a partir de características de datos semánticos, utilizados tanto en la creación de nuevos modelos como en la corrección de los modelos existentes; la búsqueda automatizada a través de datos y de modelos; la automatización de modelos basados en evidencias para garantizar la exactitud y completitud de los requisitos y el diseño del sistema; la automatización de procesos de certificación y de acreditación mediante modelos y la automatización de datos para el aseguramiento de la calidad; y, eventualmente, chatbots o asistentes cognitivos que automaticen muchas tareas rutinarias de registro de datos, investigación, cálculos de ingeniería y flujos de trabajo.

Del mismo modo, las necesidades emergentes impuestas por los LBS a la ingeniería de sistemas introducirán cambios significativos en los métodos, los procesos y las herramientas de la ingeniería de sistemas. La mayoría de ellas se reflejan en las otras tendencias recogidas en los apartados 3.1 a 3.6. Algunas de las expectativas más directas incluyen:

- Concebir nuevas arquitecturas que integren tanto arquitecturas reales como virtuales que dinámicamente se ajusten, tanto en estructura como en funcionalidad.
- Nuevas metodologías para respaldar la conciencia situacional, tanto de las personas como de las máquinas, sobre riesgos latentes y preocupaciones del comportamiento.
- Métodos, procesos y herramientas para conectar los resultados del análisis de riesgos del sistema con módulos de software de IA relacionados con esos riesgos.
- Nuevos métodos analíticos y de evaluación que calibren la confianza en los LBS, más allá de las preocupaciones tradicionales sobre la disponibilidad y la fiabilidad, hacia conceptos relacionados con la ética, la equidad, etc.
- Nuevos análisis a nivel de misión y de modelos de riesgo que surgen de la colaboración entre las personas y la IA en tareas y funciones compartidas.
- Métodos para abordar las pruebas y la evaluación de sistemas relacionados con la IA que aborden la capacidad de estos sistemas para adaptarse y aprender de unos contextos cambiantes.
- Simulación y formación asistidas por ordenador que respalden objetivos o metas cambiantes (juegos, análisis de cursos de acción), necesarios para proporcionar entornos de aprendizaje contextuales para estos sistemas.

Esto también supondrá una transformación de los ingenieros de sistemas, con una integración significativamente mayor del software y de las ciencias del comportamiento humano de vanguardia. A medida que la ingeniería digital evolucione y los modelos y las prácticas de ingeniería tradicionales dependan más de los datos subyacentes, muchas tareas de ingeniería relacionadas con la recopilación, la búsqueda, la manipulación y el análisis de datos se automatizarán. Además, el aprendizaje automático de las relaciones recogidas en los modelos y en los datos subyacentes aumentará su poder con el tiempo. Éste debería ser un cambio positivo, ya que automatizará muchas tareas cotidianas de ingeniería y llevará a una mayor concentración de los ingenieros en la resolución de problemas y en el diseño. La velocidad y la calidad de la ingeniería deberían mejorar a medida que se automaticen más actividades de prueba y validación. La idea de “asistentes cognitivos” que apoyen de forma general al ingeniero evolucionará, pero deben hacerlo de una manera que respalde la resolución de problemas y los procesos de aprendizaje asociados a la ingeniería [14, 23].

4. CONCLUSIONES

En este capítulo se han presentado cuatro nuevos tipos de sistemas que en conjunto están impulsando cambios significativos en la comunidad de ingeniería de sistemas. Estos sistemas son altamente adaptables, están altamente conectados, incluyendo CPS y otros sistemas que constituyen el núcleo tradicional de la ingeniería de sistemas, se gobiernan de forma altamente distribuida y flexible, y aprenden cada vez más nuevos comportamientos y a adaptarse por sí solos.

Se han tratado siete tendencias que cambiarán significativamente la disciplina de la ingeniería de sistemas, principalmente a partir de las actividades de investigación y de las hojas de ruta establecidas durante los últimos 5 años en el SERC. Dichas tendencias son la digitalización, los procesos ágiles, un ecosistema de ingeniería de sistemas digital en evolución, la modularidad y los modelos de negocio abiertos, los modelos digitales de sistemas, las ontologías y la tecnología semántica y, por supuesto, la IA y el ML. Muchas de estas tendencias están ocurriendo de forma independiente unas de otras, y el papel de la ingeniería de sistemas es integrarlas, tanto a ellas como a los sistemas sobre los que impactan. Algunas de ellas se desarrollarán más detalladamente en los siguientes capítulos.

Hay muchas otras tendencias que no se han incluido en este capítulo, pero que es posible que también deban incorporarse a la evolución de la práctica de la ingeniería de sistemas. Las elegidas lo han sido porque, según nuestra experiencia, son las que están teniendo un mayor impacto en la transformación de la ingeniería de sistemas en el momento de escribir este capítulo.

REFERENCES

1. Acquisition Innovation Research Center (AIRC) (2023). Agile Development of Hardware-Reliant Systems. Retrieved September 1, 2023 from <https://acqirc.org/events/agile-development-of-hardware-reliant-systems-workshop/>.
2. Bhatia H (2018). "Connected car opportunity propels multi-billion-dollar turf war." Economic Times. Retrieved September 1, 2023 from <https://telecom.economictimes.indiatimes.com/tele-talk/connected-car-opportunity-propels-multi-billion-dollar-turf-war/2971>.
3. Boardman J and Sauser B (2006). "System of systems - The meaning of 'of,'" In 2006 IEEE/SMC International Conference on System of SE. IEEE.
4. Brose C (2020). The Kill Chain: Defending America in the Future of High-Tech Warfare. Hachette Books.
5. Brunton S et al. (2020). Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning. Retrieved September 1, 2023 from https://www.researchgate.net/publication/343877323_Data-Driven_Aerospace_Engineering_Reframing_the_Industry_with_Machine_Learning.
6. Delligati L (2013). SysML Distilled: A Brief Guide to the Systems Modeling Language. Addison-Wesley.
7. Dahmann J, Rebovich G, Lane J, Lowry R, and Baldwin K (2011), "An implementers' view of SE for systems of systems," 2011 IEEE International Systems Conference, Montreal, pp. 212-217.
8. Geels F (2002). "Technological transitions as evolutionary reconfiguration processes: A multi-level perspective and a case-study," Research Policy, December 2002, 31 (8-9), 1257-1274.
9. Gorman S & Eiras A (2023). "SpaceX rocket explosion illustrates Elon Musk's 'successful failure' formula". Reuters. Retrieved September 1, 2023 from <https://www.reuters.com/lifestyle/science/spacex-rocket-explosion-illustrates-elon-musks-successful-failure-formula-2023-04-20/>.
10. Griffor E (ed) (2016). Handbook of System Safety and Security: Cyber Risk and Risk Management, Cyber Security, Adversary Modeling, Threat Analysis, Business of Safety, Functional Safety, Software Systems, and Cyber Physical Systems. Syngress.
11. Griffor, E, Greer, C, Wollman, D and Burns, M (2017), Framework for Cyber-Physical Systems: Volume 1, Overview, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], <https://doi.org/10.6028/NIST.SP.1500-201>. Retrieved September 1, 2023.
12. INCOSE (2023). SE Vision 2035: Engineering Solutions for a Better World. International Council on SE. Retrieved September 1, 2023 from <https://www.incose.org/about-systems-engineering/se-vision-2035>.

13. Maier M (1996). "Architecting Principles for Systems of Systems," Proc. of the Sixth Annual International Symposium, International Council on SE, Boston, MA, pp. 567- 574.
14. McDermott, T, Blackburn, M, & Beling, P (2021). Artificial Intelligence and Future of SE. In: Lawless, W.F., Mittu, R., Sofge, D.A., Shortell, T., McDermott, T.A. (eds) SE and Artificial Intelligence. Springer, Cham. https://doi.org/10.1007/978-3-030-77283-3_3.
15. McDermott, T, DeLaurentis, D, Beling, P, Blackburn, M & Bone, M (2020), AI4SE and SE4AI: A Research Roadmap. INSIGHT, 23: 8-14. <https://doi.org/10.1002/inst.12278>.
16. McDermott T, Alexander K, & Wallace R (2023). The Supra-System Model. INSIGHT 26 (2), 15-21.
17. McDermott and Horowitz 2017 is "McDermott T, Horowitz B (2017) Human Capital Development – Resilient Cyber Physical Systems. Systems Engineering Research Center (SERC) Technical Report SERC-2017-TR-075, September 29, 2017.
18. National Academies of Sciences, Engineering, and Medicine (2016). A 21st Century Cyber-Physical Systems Education. Washington, DC: The National Academies Press. doi:10.17226/23686.
19. National Science Foundation (NSF) (2016). Cyber Physical Systems (CPS). Retrieved September 1, 2023 from <https://new.nsf.gov/funding/opportunities/cyber-physical-systems-cps>.
20. Office of the Deputy Assistant Secretary of Defense for SE (2018). Department of Defense Digital Engineering Strategy, Washington DC.
21. Rotmans J, Kemp R, & van Asselt M (1999). "More evolution than revolution: transition management in public policy," Foresight 3 (1).
22. SAE (2023). OnQue™ Digital Standards System. Retrieved September 1, 2023 from <https://www.sae.org/onque-digital-standards>.
23. Salado A and Selva D (2021). "Asistentes Cognitivos en Ingeniería," UE Essentials, Universidad Europea de Madrid, Madrid, Spain.
24. Systems Engineering Research Center (SERC) (2017). Human Capital Development – Resilient Cyber Physical Systems. Technical Report SERC-2017-TR-075. Stevens Institute of technology, Hoboken NJ.
25. SERC (2023). Information Models and Ontologies to Enable Digital Engineering. Retrieved September 1, 2023 from <https://sercuarc.org/event/information-models-and-ontologies-to-enable-digital-engineering-research-workshop/>.
26. Shadab N, Kulkarni A, Salado A (2021). "Challenges to the Verification and Validation of AI-enabled Systems: A Systems-Theoretic Perspective," in W.F. Lawless, R. Mitty, D. Sofge, T. Shortell, T. McDermott (Eds.), Systems Engineering and Artificial Intelligence, (pp. 363-378), Cham: Springer, 2021.
27. von Bertalanffy L. (1969). General Systems Theory. New York: George Braziller.
28. Wikipedia contributors. (2021, September 29). Adaptability. In Wikipedia, The Free Encyclopedia. Retrieved September 1, 2023, from <https://en.wikipedia.org/w/index.php?title=Adaptability&oldid=1047218141>.
29. Zhu H and Arnold E, "System Adaptability." in SEBoK Editorial Board. 2023. The Guide to the SE Body of Knowledge (SEBoK), v. 2.8, R.J. Cloutier (Editor in Chief). Hoboken, NJ: The Trustees of the Stevens Institute of Technology. Accessed September 1, 2023. www.sebokwiki.org. BKCASE is managed and maintained by the Stevens Institute of Technology SE Research Center, the International Council on SE, and the Institute of Electrical and Electronics Engineers Systems Council.

BIOGRAFÍAS

THOMAS ALLEN MCDERMOTT

Tom McDermott es el Jefe de Tecnología (CTO) del Centro de Investigación de Ingeniería de Sistemas (SERC) y miembro de la facultad de la Escuela de Sistemas y Empresas en el Stevens Institute of Technology en Hoboken, Nueva Jersey. En el SERC desarrolla nuevas estrategias de investigación y lidera las actividades de investigación en el ámbito de aplicaciones de transformación digital, formación, seguridad de la información e inteligencia artificial. Anteriormente fue profesor y Director de investigación en el Georgia Tech Research Institute y Director y Gestor de Equipo Integrado de Producto en Lockheed Martin. El señor McDermott enseña arquitectura de sistemas, pensamiento sistémico y pensamiento crítico, y liderazgo en ingeniería. Proporciona servicios de consultoría a nivel ejecutivo como experto en los ámbitos de la ingeniería de sistemas y de la estrategia organizativa. Es fellow del Consejo Internacional de Ingeniería de Sistemas (INCOSE) y recientemente ha completado un periodo de 3 años como Director de Integración Estratégica en INCOSE.



VÍCTOR RAMOS DEL POZO

Víctor Ramos del Pozo es ingeniero de sistemas en Ingeniería de Sistemas para la Defensa de España (Isdefe) que actualmente presta asistencia técnica en la Oficina de Programa Española del Sistema de Armas de Siguiete Generación (NGWS) dentro del entorno de un Futuro Sistema de Combate Aéreo (FCAS), especialmente en los ámbitos de la nube de combate, sensores colaborativos y simulación. Anteriormente prestó asistencia técnica en la Dirección General de Armamento y Material en los ámbitos del análisis de la base industrial de defensa, el planeamiento de la obtención de armamento y material, y en la gestión de grandes programas de defensa.



Antes de unirse a Isdefe, Víctor trabajó como ingeniero de sistemas en EADS-CASA (actual Airbus Defence and Space) en los programas de aviones de reabastecimiento en vuelo MRTT y FSTA, así como en Indra en el departamento de sistemas de identificación amigo-enemigo (IFF).

Es miembro del Consejo Internacional de Ingeniería de Sistemas (INCOSE) y es formador interno en Isdefe en los ámbitos de la ingeniería de sistemas y de la gestión de programas.



“Tenemos desarrollos que siguen todas las reglas, pero fallan.”

J. Frosch

La evolución de los modelos de desarrollo y ejecución de ingeniería de sistemas

Ronald Giachetti, *Naval Postgraduate School (regiache@nps.edu)*
Juan Carlos Lario Monje, *Isdefe (jclario@isdefe.es)*

Resumen

Este capítulo presenta la necesidad de evolucionar y adaptar los modelos de desarrollo de ingeniería de sistemas al contexto actual de proyectos de ingeniería. El capítulo separa el debate entre aproximaciones al desarrollo de sistemas tradicionales, predominantemente *en base a un plan*, tales como los modelos “en cascada” y “en V”, y la discusión de aproximaciones al desarrollo *ágiles* que ponen el acento en la capacidad de respuesta, y caracterizadas por su carácter altamente iterativo e incremental. La mayoría de los proyectos podría beneficiarse de ambos, y por ello se exponen aproximaciones híbridas y cómo adaptar los modelos de desarrollo al contexto de proyectos particulares. El capítulo debate sobre otras metodologías emergentes incluyendo la confluencia del desarrollo con operaciones por medio de *DevOps*.

Palabras clave

Ingeniería de sistemas, modelos de desarrollo, metodologías ágiles, DevOps, utilización híbrida.



1. INTRODUCCIÓN

Un modelo de desarrollo define un enfoque para desarrollar un sistema que satisfaga las necesidades identificadas por uno o más de los actores interesados (*stakeholders*). El desarrollo de sistemas complejos y a gran escala conlleva con frecuencia la utilización de múltiples tecnologías, así como la participación de un elevado número de individuos diversos que aportan distintos conocimientos y capacidades al esfuerzo global de desarrollo. Este capítulo describe primeramente el propósito y la intención de los modelos de desarrollo de sistemas. Los modelos se organizan en dos categorías: modelos estructurados *en base a un plan* y modelos *ágiles*. Los modelos *en base a un plan* asumen que conocemos suficientemente las necesidades y la solución, de modo que podemos planificar con antelación el proyecto de desarrollo y después desarrollar el sistema de acuerdo al plan. Los modelos *ágiles* se sitúan en una posición en la que no comprendemos totalmente las necesidades o la solución y por tanto la ejecución del proyecto debe ser adaptativa o flexible, permitiendo cambios en las necesidades o en la solución. Tras introducir múltiples modelos de desarrollo representativos, el capítulo discute después cómo adaptar un modelo de desarrollo a un proyecto particular, tomando en consideración todos los factores contextuales que contribuyen a un resultado exitoso.

1.1. Contexto del desarrollo de sistemas

El desarrollo de sistemas complejos y a gran escala impone demandas múltiples de naturaleza organizacional (esto es, sobre la organización que desarrolla el sistema, también llamado sistema-realización), que un modelo de desarrollo de sistemas está destinado a abordar. En este sentido, las demandas básicas que un equipo que desarrolle un nuevo sistema habrá de acometer son la comprensión de las necesidades de los “*stakeholders*”, la creación de un concepto que dé respuesta a esas necesidades, diseñar e integrar dicho concepto con hardware, software, procesos y personas, y después implementar el sistema diseñado de modo que pueda ser puesto en funcionamiento.

El desarrollo de un sistema se encuentra casi siempre restringido por un calendario y un presupuesto, y los modelos de desarrollo de un sistema deben ser viables bajo estas condiciones. Una organización involucra el trabajo de muchas personas con diferentes conocimientos y habilidades a lo largo del desarrollo del sistema, y todas ellas deben estar organizadas de una manera eficaz para diseñar y construir el sistema en consideración. Un modelo de desarrollo proporciona un medio para comprender cómo organizar el trabajo de todos estos individuos.

Muchos sistemas incorporan tecnologías novedosas o emergentes que introducen riesgos en el desarrollo. Bajo estas condiciones un modelo de desarrollo de sistemas debe ser capaz de proporcionar medios de maduración para dichas tecnologías y de valoración de los riesgos asociados, tanto para la organización de desarrollo como para el propio sistema (ej. riesgos que pueden materializarse durante la fase de operación). Muchos sistemas, como aquellos que integran una mayor autonomía o inteligencia artificial, afrontan el reto de tener que compatibilizar el desarrollo de una tecnología en paralelo a su propio desarrollo. De manera simultánea, los denominados sistemas críticos en seguridad (“*safety-critical systems*”), tales como aeronaves o dispositivos médicos, deben ser desarrollados y operar en entornos altamente regulados. Estos aspectos generalmente imponen restricciones severas en la generación de información y datos durante el desarrollo del sistema, el cual se ve posibilitado o facilitado al adoptar un modelo de desarrollo adecuado.

Dado que el software se ha convertido en un componente esencial en la mayoría de los sistemas, el desarrollo de un sistema debe lidiar en la actualidad con la en general elevada complejidad introducida por el software, así como con potenciales vulnerabilidades de seguridad (*security*) inexistentes previamente, o con despliegues rápidos casi a golpe de *click*. En el pasado, cuando los sistemas consistían mayoritariamente en componentes hardware, las capacidades del sistema permanecían en general inmutables una vez que el sistema era desplegado. Dado que el software no es algo físico (es un conjunto de instrucciones que deben ser instaladas sobre un hardware), ello posibilita a los propietarios del sistema actualizar continuamente las capacidades del sistema durante su utilización. Más aún, los componentes software han posibilitado también la implementación de inteligencia artificial, la cual puede crear escenarios en los que los algoritmos o las decisiones adoptadas por el sistema puedan cambiar a lo largo del tiempo [26]. Esto presenta desafíos para la verificación del sistema, así como para la confianza humana depositada en el sistema en lo que a seguridad (*safety*) se refiere, o para otros aspectos. De manera más global, estas características afectan tanto al proceso de desarrollo de sistemas *intensivos en software* como a cualesquiera componentes-software parte de sistemas más grandes.

Esta sección ha descrito el contexto que rodea a proyectos modernos de desarrollo de sistemas complejos o a gran escala. Los modelos de desarrollo de sistemas han evolucionado para poder hacer frente a muchos de los aspectos antes presentados.

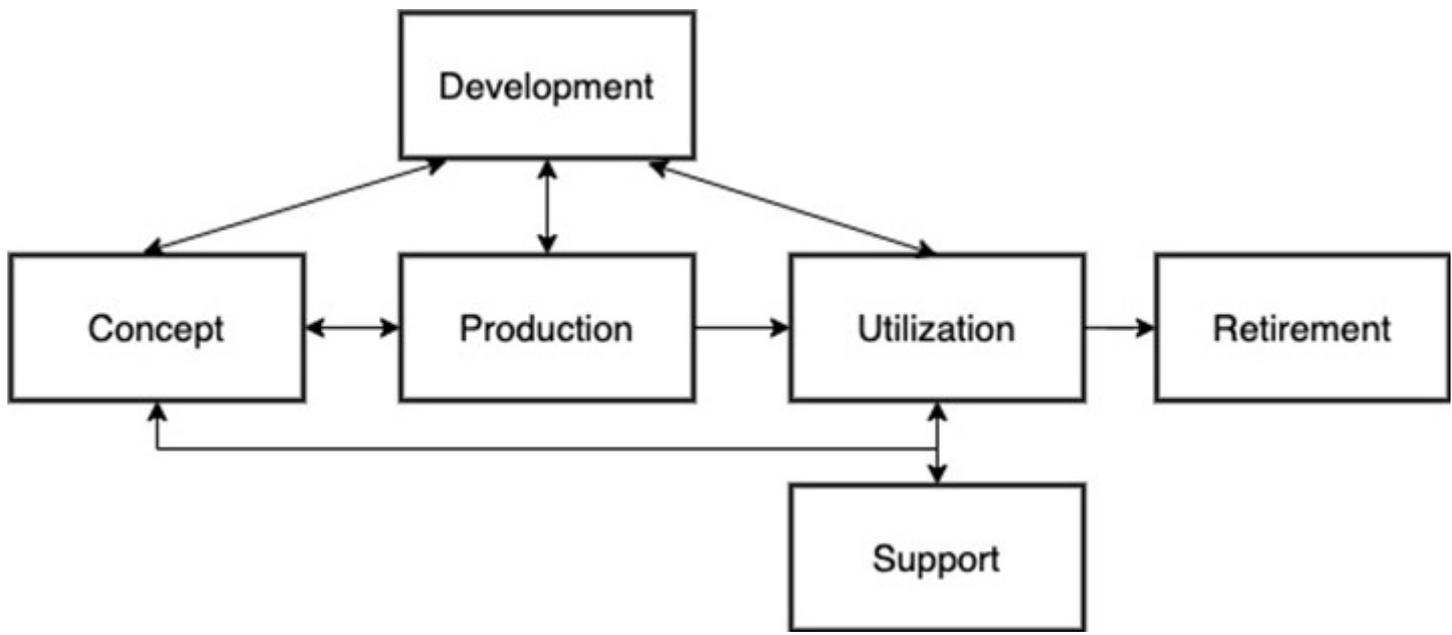


Figura 1. Ciclo de vida de un sistema [20]

1.2. Definición de un modelo de desarrollo de un sistema

Un modelo de desarrollo de un sistema proporciona un marco de trabajo estructurado y un conjunto de pautas generales para organizar las actividades de desarrollo y coordinar el desarrollo técnico de un sistema. Los modelos de desarrollo se rigen frecuentemente por una serie de principios o buenas prácticas. La utilización de un modelo de desarrollo contribuye a organizar y gestionar los esfuerzos asociados, garantizando un determinado nivel de calidad, reduciendo riesgos y mejorando la colaboración y la comunicación entre los miembros de un equipo y otras partes interesadas. Existe una correlación entre el hecho de seguir un proceso predefinido y una mayor probabilidad de entregar exitosamente un sistema con unas capacidades deseadas, dentro de las restricciones impuestas de tiempo, coste y calidad [7].

Los modelos de desarrollo de sistemas dividen, por regla general, el proceso en fases que describen el progreso o evolución del sistema a lo largo de su ciclo de vida. Los hitos *son puertas de decisión* que marcan la terminación de una fase y en los que la organización de desarrollo decide si continuar el desarrollo del sistema a la siguiente fase, esperar al completo cumplimiento de cierto número de acciones identificadas como pendientes o, en el peor de los casos, dar por finalizado el proyecto. Los criterios de entrada y salida de cada hito están en general bien definidos de antemano, de cara a minimizar incertidumbres no deseadas. En este sentido, el riesgo de continuar adelante es a menudo valorado y ponderado antes de tomar la decisión.

1.3. Modelo del ciclo de vida de un sistema

El desarrollo de un sistema debe ser entendido en el contexto de su *ciclo de vida*. El ciclo de vida de un sistema describe las fases por las que el sistema transita desde su conceptualización inicial hasta su retirada o eliminación y de las cuales el desarrollo es solo una parte. La Figura 1 muestra un ejemplo de cómo la ISO/IEC-15288 las define. Las flechas indican la existencia de múltiples rutas potenciales a través de las diferentes fases y que pueden implicar ciclos de iteración y repetición. Un sistema nuevo y sin precedentes podría atravesar las fases de manera secuencial, desde la de *Concepto* a las de *Desarrollo*, *Producción*, *Utilización* y *Soporte*, antes de procederse a su *Retirada*. Un sistema previamente existente, en cambio, podría ser actualizado y circulado nuevamente desde su fase de *Utilización* a la de *Concepto*. Algunos sistemas pueden ser diseñados y puestos en funcionamiento de forma incremental y transitarían por múltiples ciclos de iteración de las fases para aspectos diversos del sistema global.

Nótese que las actividades de ingeniería de sistemas, tales como la obtención de necesidades de las partes interesadas o la integración del sistema, no están exclusivamente asociadas con ninguna de las fases individuales del ciclo de vida del sistema. Es más, algunas actividades de ingeniería de sistemas pueden ser ejecutadas iterativa y recursivamente sobre múltiples niveles de la jerarquía del sistema. Como consecuencia, la ingeniería de sistemas abarca todas las fases del ciclo de vida de un sistema.

1.4. Modelos de desarrollo de sistemas

Dado que el conjunto del conocimiento alrededor de los modelos de proceso evoluciona, el Consejo Internacional de Ingeniería de Sistemas (INCOSE) ha revisado su clasificación de modelos de desarrollo de sistemas en varias ocasiones [20]. Describe 2 categorías:

- 1) Modelos predeterminados o *en base a un plan*.
- 2) Modelos evolutivos o *ágiles*.

La primera categoría incluye modelos tradicionales tales como el modelo “en cascada” y el modelo “en V”. La segunda categoría describe una evolución respecto a los modelos estrictamente *en base a un plan*, e incluye el modelo “en espiral”, basado en ciclos de desarrollo iterativos. La segunda categoría ha venido en ser denominada métodos *ágiles* y se inspira considerablemente en la comunidad de ingeniería del software. La categorización está basada en las características *dominantes* de los modelos correspondientes. No obstante, nótese que existe un cierto solape entre categorías, dado que la mayoría de modelos comparten características. Por ejemplo, numerosos autores presentan el modelo “en cascada” como el arquetipo de modelo *en base a un plan*, con una ejecución estrictamente secuencial de actividades. Sin embargo, el artículo original de Royce reconocía la iteración entre las actividades. De hecho, iteración y recursividad son siempre inherentes a cualquier proceso de desarrollo, dentro y entre sus diferentes fases.

2. MODELOS DE DESARROLLO EN BASE A UN PLAN

Los modelos en base a un plan describen una clase de modelos de desarrollo de sistemas que priorizan la predictibilidad disponible mediante la adhesión a un plan que configura una aproximación estructurada de desarrollo. Esta sección revisa los modelos arquetípicos de desarrollo *en base a un plan* “en cascada” y “en V”.

2.1. Modelo en cascada

El modelo “en cascada” es un modelo *en base a un plan* en el que todas las actividades ingenieriles están definidas y planificadas al inicio del proyecto. Cada actividad del modelo en cascada se supone que es ejecutada *completamente* antes de que la siguiente actividad comience. Adicionalmente, el output de la actividad precedente se convierte en input de la siguiente.

La iteración de actividades no está planeada en el modelo en cascada, pero se espera que se produzca únicamente cuando los problemas no son resolubles en una etapa y, en tal caso, el proyecto se *desplaza* a la etapa inmediatamente anterior. La Figura 2 muestra la estructura *secuencial* típica de las actividades en el modelo en cascada.

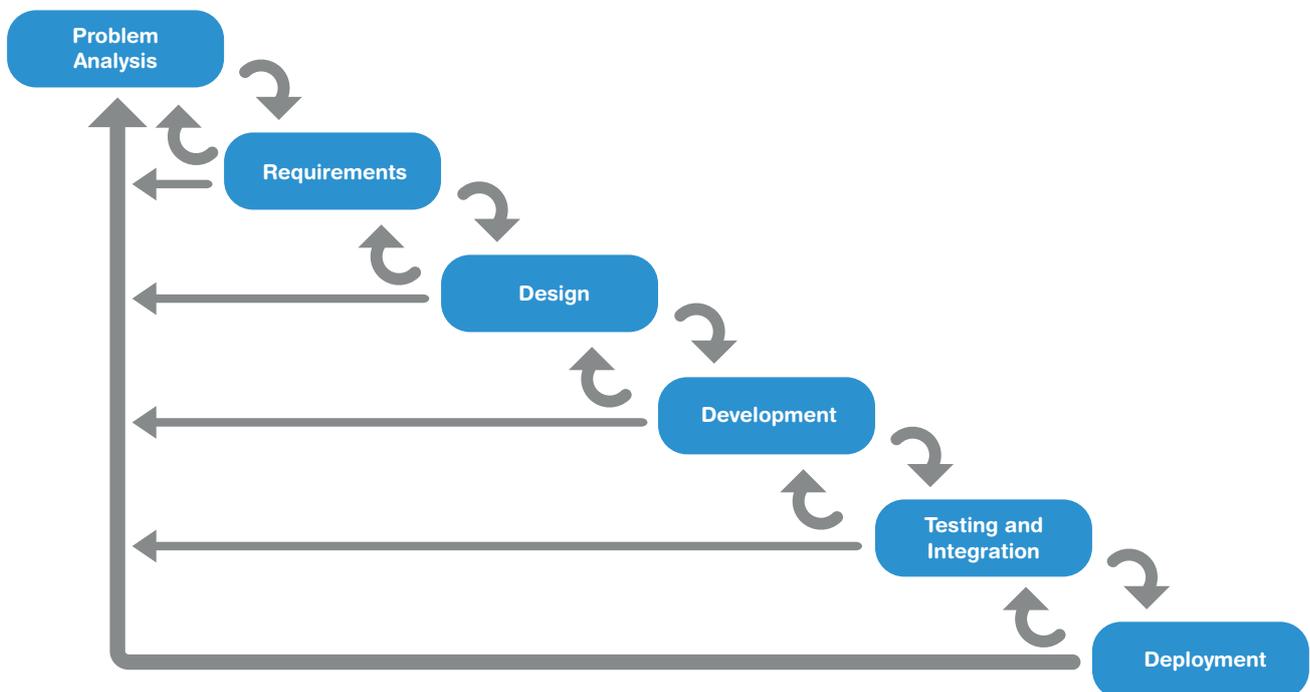


Figura 2. Modelo en cascada

Aunque en la actualidad el modelo “en cascada” es a menudo criticado y casi considerado irrelevante en los círculos de ingeniería de sistemas, todavía resulta adecuado y útil en proyectos en los que:

- sea posible conocer con seguridad los requisitos al inicio;
- dichos requisitos sean estables e inmutables;
- exista un nivel de riesgo técnico bajo;
- o haya riesgos peligrosos que requieran una cuidadosa progresión de las actividades.

Bajo estas condiciones, la aproximación secuencial y estructurada para el desarrollo de sistemas del modelo “en cascada” proporciona los beneficios de predictibilidad y de una coordinación más sencilla de los trabajos.

La implementación del modelo “en cascada” supone un desafío y puede causar extensas y costosas labores de retrabajo en escenarios en los que:

- los requisitos sean mal conocidos;
- haya probabilidades de que los requisitos evolucionen o cambien;
- o sea posible que nuevos requisitos surjan durante el desarrollo.

Cualquier cuestión descubierta durante actividades posteriores tales como ensayos o integración, puede suponer una significativa cantidad de retrabajo, dado que el equipo puede necesitar recorrer nuevamente el camino hasta los requisitos y rediseñar el sistema. Tales retrabajos a menudo provocan significativos sobrecostos sobre el presupuesto y desviaciones en la planificación. Más aún, el modelo “en cascada” no incorpora explícitamente la realimentación de los *stakeholders* durante el ciclo de vida, ya que cada actividad se encuentra encajonada temporalmente dentro de su propia etapa, lo cual limita su capacidad de validar conceptos antes de hacer demasiado trabajo, o de adaptarse a necesidades cambiantes. Además, la rigidez del proceso de desarrollo elimina cualquier oportunidad de desplegar el sistema de manera *incremental*. Estos aspectos hacen que el modelo “en cascada” resulte poco atractivo para la mayoría de los proyectos de ingeniería.

2.2. Modelo “en V”

El modelo “en V” de ingeniería de sistemas es un modelo *en base a un plan* que se vale de la naturaleza jerárquica de la mayoría de los sistemas complejos. El desarrollo se produce siguiendo una secuencia de arriba hacia abajo (*top-down*) en la que un sistema es *recursivamente* descompuesto en sistemas de más bajo nivel (a los que pueden asignarse nombres tales como subsistemas, componentes, etc.). Es de destacar que el modelo “en V” reconoce que la incertidumbre y la madurez tecnológica no son uniformes a lo largo de todos los elementos del sistema y por ello permite que elementos diferentes del sistema evolucionen a distintos ritmos. Es ésta en realidad su propiedad fundamental, que a pesar de que todas las actividades deben concluir secuencialmente (como era el caso para el modelo “en cascada”), pueden comenzar en cualquier orden. Es decir, dentro de un modelo “en V”, las actividades podrían no arrancar en el mismo orden para cada elemento del sistema, pero su finalización se produce por las restricciones del proceso, tales como que la integración no pueda completarse hasta que todos los elementos del sistema sean fabricados. De hecho, el modelo “en V” pone el acento en la *Verificación* y la *Validación* en cada nivel y entre niveles, lo cual puede rápidamente conducir a bucles de iteración mientras el equipo resuelve problemas para minimizar los impactos en coste y planificación de los retrabajos [32]. Por ejemplo, el modelo “en V” incentiva a comenzar la validación en paralelo con la captación de las necesidades de los stakeholders, iniciar la verificación en paralelo con la derivación de requisitos, comenzar la integración tan pronto como se inicie la arquitectura del sistema, etc.

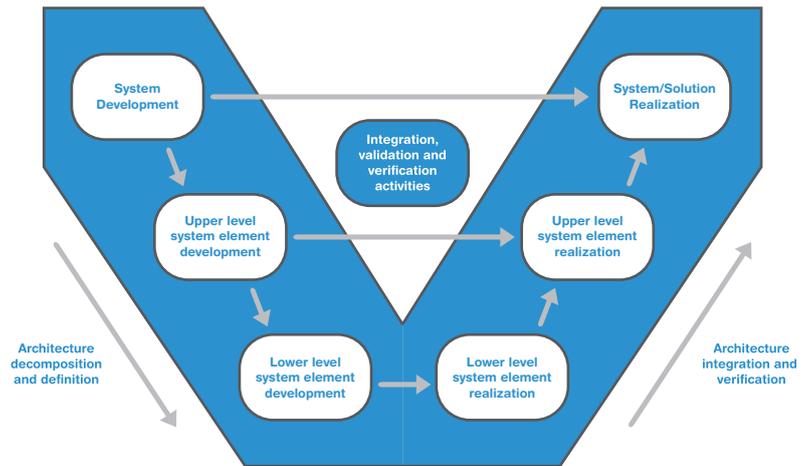


Figure 3. Vee Model

La Figura 3 muestra la forma original y canónica para el modelo “en V” [14], del cual existen en la actualidad múltiples variaciones (muchas de ellas inconsistentes con los conceptos expresados en el modelo “en V” original). La parte izquierda del mismo describe el esfuerzo de descomposición asociado al mapeado de las necesidades de los “*stakeholders*” en conceptos de diseño del sistema y, finalmente, en una especificación del sistema. En la parte inferior del modelo “en V”, los mencionados elementos del sistema se diseñan y fabrican, dando paso a la parte derecha del modelo, la cual refleja la síntesis o esfuerzo

de integración conforme dichos elementos son integrados aguas arriba de la jerarquía del sistema con verificaciones y validaciones produciéndose en cada escalón. La simetría entre las partes izquierda y derecha del modelo pone de manifiesto las interdependencias entre cada estadio de la izquierda con el correspondiente de la derecha. A modo de ejemplo, el concepto de operaciones de la parte izquierda define las *medidas de efectividad y de prestaciones* que serán la base para la posterior validación del sistema.

El modelo “en V” resulta particularmente adecuado para proyectos de desarrollo de sistemas que requieran de una aproximación sistemática y estructurada, en la que:

- los sistemas en cuestión sean de gran escala;
- involucren multitud de componentes, los cuales posean diferentes niveles de madurez y, por tanto, de riesgo;
- y presenten rigurosas restricciones de seguridad (*safety*) y de fiabilidad.

El modelo “en V”, o variaciones de carácter menor sobre el mismo, es probablemente el de uso más generalizado en industrias tales como la aeronáutica, la defensa o la automoción.

Este modelo reproduce algunas de las mismas deficiencias experimentadas con el modelo “en cascada”. El modelo adolece de una pobre capacidad de respuesta frente a cambios significativos en los requisitos o frente a premisas de partida inadecuadas que no sean descubiertas hasta más adelante en el proceso. El resultado es un costoso retrabajo con los consiguientes retrasos en la planificación.

3. MODELOS DE DESARROLLO EVOLUTIVOS O ÁGILES

Los modelos evolutivos o ágiles describen una clase de aproximaciones al desarrollo de sistemas altamente iterativas e incrementales. Los modelos *ágiles* llevan a cabo desarrollos iterativos e incrementales basados en cortos ciclos de desarrollo conocidos como *sprints* o iteraciones. Una iteración describe la repetición de las mismas actividades durante cada *sprint*, tales como el análisis, diseño, ensayo y fabricación para una pequeña porción del sistema. Es posible visualizar el carácter incremental a través de la aportación de valor o de capacidades del sistema a la conclusión de cada *sprint*. Los modelos *ágiles* subrayan el hecho de que el final de un *sprint* debería dar lugar a un resultado medible. Para sistemas software, cada *sprint* concluye típicamente con un código funcional que puede ser entregado a los *stakeholders*. Para sistemas intensivos

en hardware un *sprint* podría no derivar en un hardware real, debido a los mayores tiempos de desarrollo requeridos, pero en su lugar un *sprint* puede llevar a una reducción de un riesgo, a una especificación ingenieril o a la finalización de un análisis de ingeniería. Este tipo de modelos enfatizan la aportación de valor como un proceso *incremental*, si tal aportación al concluir cada *sprint* se traduce en alguna porción visible del sistema o en un avance del proyecto.

Al contrario que los modelos *en base a un plan*, los modelos *ágiles* llevan a cabo una planificación y un análisis de requisitos al inicio del desarrollo del sistema significativamente menores. Los modelos *ágiles* son enfoques altamente adaptativos y flexibles, dado que, conforme el equipo de proyecto planea cada nuevo *sprint*, llevarán a cabo correcciones del rumbo basadas en lo aprendido durante el *sprint* anterior. De este modo, las metodologías *ágiles* están actualizando regularmente sus planes, dirigidas por la frecuencia de los *sprints*.

Los modelos *ágiles* se basan en un conjunto de principios que conforman una actitud y que posteriormente se implementan por múltiples y diferentes métodos. Estos principios fueron descritos en el “*Agile Manifesto*” [2] y se enumeran en la Tabla 1.

1. Pronta y continua aportación de valor al cliente.
2. Aceptación de requisitos cambiantes (una notable ventaja competitiva).
3. Frecuente aportación de valor incremental a los <i>stakeholders</i> .
4. Trabajo cooperativo de usuarios y desarrolladores.
5. Foco en las personas; equipos auto-organizados.
6. Comunicación cara a cara para potenciar el trabajo en equipo.
7. Software operativo como medida primaria del progreso.
8. La velocidad en el desarrollo debiera ser sostenible en el tiempo de manera indefinida.
9. Atención continua a un buen diseño.
10. Mantenimiento de un diseño tan sencillo como sea posible, evitando trabajos innecesarios.

11. Las mejores arquitecturas surgen de equipos auto-organizados.
12. Reuniones periódicas para que los equipos reflexionen sobre las lecciones aprendidas.

Tabla 1. Principios ágiles

Un aspecto diferencial de los modelos *ágiles* es que son capaces de combinar principios sobre cómo desarrollar un sistema con otros sobre cómo organizar un equipo de desarrollo. Es este un aspecto diferencial en comparación con muchos de los modelos *en base a un plan*, que proporcionan poca guía sobre consideraciones de gestión de proyectos del desarrollo de un sistema. Los principios se ocupan de entregas continuas, aceptación de requisitos cambiantes, utilización de software operativo como medida del progreso, velocidades de desarrollo mantenidas en el tiempo y una atención permanente a un buen diseño, a la par que sencillo. Los modelos *ágiles* prescriben de manera uniforme equipos auto-organizados y multidisciplinares que trabajen en colaboración cercana con los usuarios finales (generalizable a cualesquier *stakeholders*) para generar valor de forma incremental. Más aún, asignan un gran valor a las comunicaciones cara a cara y a las reuniones periódicas, como medios para una mejor coordinación y una más rápida identificación y adopción de lecciones aprendidas, destinadas a mejorar la gestión de los equipos de desarrollo.

Las metodologías *ágiles* han sido ampliamente adoptadas y puestas en práctica de forma rutinaria en proyectos de desarrollo software de cara a mejorar la colaboración y comunicación de equipos, así como la flexibilidad en la respuesta a cambios en los requisitos del cliente o de los usuarios. La aplicación de principios *ágiles* a los procesos de ingeniería de sistemas, particularmente en el desarrollo de sistemas complejos con componentes hardware, sigue siendo un área activa de investigación y debate. Estudios diversos han mostrado los beneficios de aplicar principios *ágiles* a los procesos de ingeniería de sistemas. Dove et al. [11] analizaron la utilización de principios ágiles en varias compañías e identificaron ocho de ellos para una ingeniería de sistemas ágil consistentes con el “*Agile Manifesto*”, incluyendo una preocupación por el entorno de negocio, por el proceso de toma de decisiones, por los conceptos de operación *ágiles*, por la arquitectura de las líneas de producto, por la gestión del conocimiento compartido, por una integración y pruebas continuas, por la identificación del equipo con un objetivo común, y por un desarrollo iterativo e incremental. El Consejo Internacional de Ingeniería de Sistemas (INCOSE) posee un equipo que ha investigado cómo las metodologías ágiles pueden ser inculcadas y

adoptadas por las organizaciones de ingeniería de sistemas [38]. Sus trabajos forman parte de la iniciativa FuSE (*Future of Systems Engineering*) de INCOSE, y subrayan cómo las organizaciones de desarrollo pueden llegar a ser más *ágiles* mediante el desarrollo de procesos y de trabajadores. Por tanto, mientras que los modelos *ágiles* ya han sido contrastados y son los enfoques dominantes en software, la ingeniería de sistemas se encuentra aún examinando y tratando de comprender cuál sería la mejor manera de adoptar principios *ágiles* dadas las restricciones físicas impuestas por el hardware.

A continuación, se presenta, como ejemplo de modelo evolutivo, el modelo “en espiral”, que fue introducido en los años 80. Seguidamente se expone Scrum, porque es ampliamente utilizado en la industria del software, y el proceso iterativo y los conceptos que desde Scrum han sido considerablemente replicados por otros modelos evolutivos. Finalmente, se analizan dos modelos de desarrollo de sistemas que adoptan conceptos *ágiles* provenientes de Scrum para proyectos de mayor magnitud que incluyen tanto hardware como software: el “Marco Ágil Escalado” (SAFe, por sus siglas en inglés: *Scaled Agile Framework*) y el enfoque denominado “*Disciplined Agile Delivery*” (DAD).

3.1. Modelo “en espiral”

El modelo “en espiral” fue introducido para desarrollo software, y pone el acento en la reducción del riesgo mediante un compromiso incremental por parte de los *stakeholders* a lo largo del desarrollo del sistema [3]. Un proyecto que siga el modelo espiral realiza iteraciones para desarrollar progresivamente la definición de un sistema software y entregarlo finalmente a un cliente. Los ciclos consisten en cuatro actividades (cfr. Figura 4):

- 1) determinación de objetivos, alternativas y restricciones;
- 2) evaluación de las alternativas, identificando y solucionando riesgos;
- 3) desarrollo y verificación del producto;
- 4) planeamiento de la siguiente fase.

Cada iteración o espiral da como resultado un prototipo o versión del sistema, aunque a menudo una versión parcial. Como consecuencia, el proyecto desarrolla y despliega el sistema en incrementos.

El modelo “en espiral” resulta adecuado en contextos de desarrollo de sistemas caracterizados por requisitos cambiantes o con incertidumbre, y en proyectos en los que los riesgos sean significativos y demanden ser continuamente valorados y mitigados a lo largo del ciclo de vida del

proyecto. Sin embargo, la utilización del modelo “en espiral” normalmente solo es posible cuando el sistema puede ser desplegado en incrementos. Los sistemas software fueron uno de los campos originales para los que el modelo en espiral fue concebido [4], pero ha sido también aplicado a otros dominios. A título de ejemplo, el proceso de diseño de buques es habitualmente mostrado como un modelo en espiral, dado que involucra múltiples especialidades técnicas con un alto grado de interdependencia entre ellas. En el diseño espiral de buques, los diseñadores toman decisiones sobre la forma del casco, su tamaño, potencia, hidrostática, etc., y deben iterar estas decisiones hasta converger en un diseño aceptable [13,15]. El modelo “en espiral” ha sido también aplicado al desarrollo del vehículo aéreo no-tripulado Global Hawk, que fue capaz de poner en marcha capacidades de manera incremental en cada espiral [17].

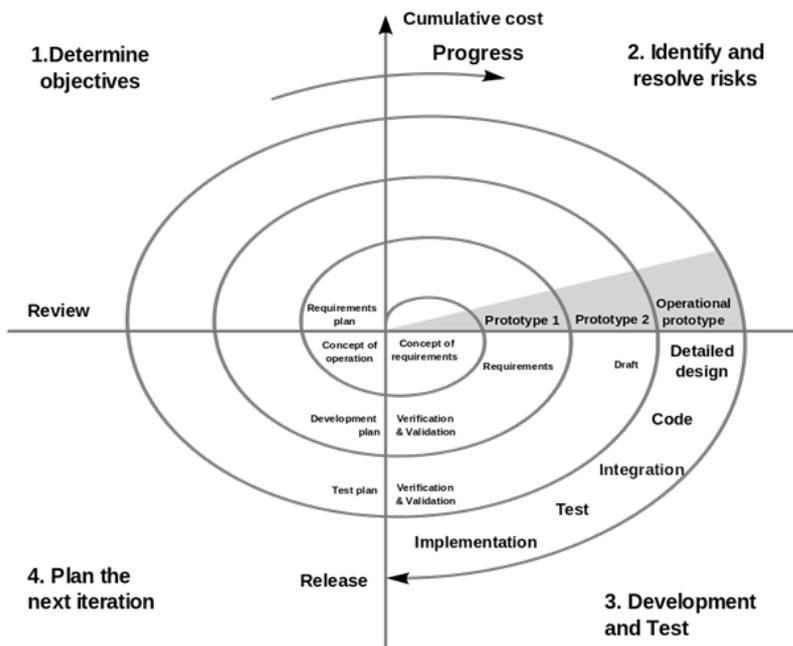


Figura 4. Modelo en espiral

3.2. Scrum

Scrum es un modelo *ágil* de desarrollo software para la gestión de equipos auto-organizados capaces de entregar código operativo tras cada finalización de cortas iteraciones denominadas *sprints* [34]. La popularidad y éxito de Scrum para el desarrollo software ha desencadenado que muchos profesionales adapten el modelo de desarrollo Scrum a sistemas que incorporan ambos elementos: software y hardware [3,10]. La Figura 5 muestra el proceso de Scrum. Los equipos trabajan sobre una acumulación de tareas identificadas en conjunción con un Propietario de Producto que representa la voz del cliente. Los elementos de trabajo no son sino requisitos de características o funciones expresados a menudo en forma de explicaciones informales (“*user stories*”). Durante la sesión de planificación del *sprint*, el equipo selecciona algunos de esos ítems sobre los que llevar a cabo análisis, diseño, desarrollo y pruebas. Un *sprint* es en realidad una iteración de desarrollo y su duración es típicamente de dos semanas en proyectos software, aunque puede dilatarse y presentar variabilidad en el caso de desarrollo hardware. El equipo tendrá una reunión puntual para comentar el progreso del *sprint*. Asimismo, se celebra una reunión final al concluir el *sprint* para que el equipo debata estrategias en las que puedan mejorar. Scrum es un proceso de aprendizaje que pone el foco en las personas y sus interacciones durante el desarrollo de un software.

Scrum funciona bien en entornos dinámicos en los que los requisitos cambien (pues dichos requisitos nuevos o modificados pueden ser incorporados al reservorio o *backlog* y ser abordados en *sprints* futuros), así como con sistemas que puedan ser fabricados y puestos en funcionamiento de forma incremental. La adopción de modelos de desarrollo Scrum requiere la implementación de la correspondiente cultura organizacional, como ocurre con otros modelos *ágiles*, debido a que es capaz de interrelacionar procesos de desarrollo con principios de gestión de equipos y de comunicación. Una organización que no se identifique con tales principios de trabajo establecidos por el equipo, probablemente no tendría éxito al llevar a cabo un proceso Scrum.

Dado que Scrum fue originariamente aplicado sobre proyectos reducidos, existían dudas iniciales acerca de su idoneidad para dirigir proyectos de mayor envergadura. Sin embargo, se propusieron variantes tales como *Large-Scale-Scrum* (LeSS) [24], para abordar esta potencial limitación. Aunque Scrum ha sido adaptado y aplicado en el desarrollo de sistemas, no está ampliamente extendido y permanecen abiertos numerosos interrogantes relativos a su aplicabilidad en diferentes contextos de desarrollo de sistemas.

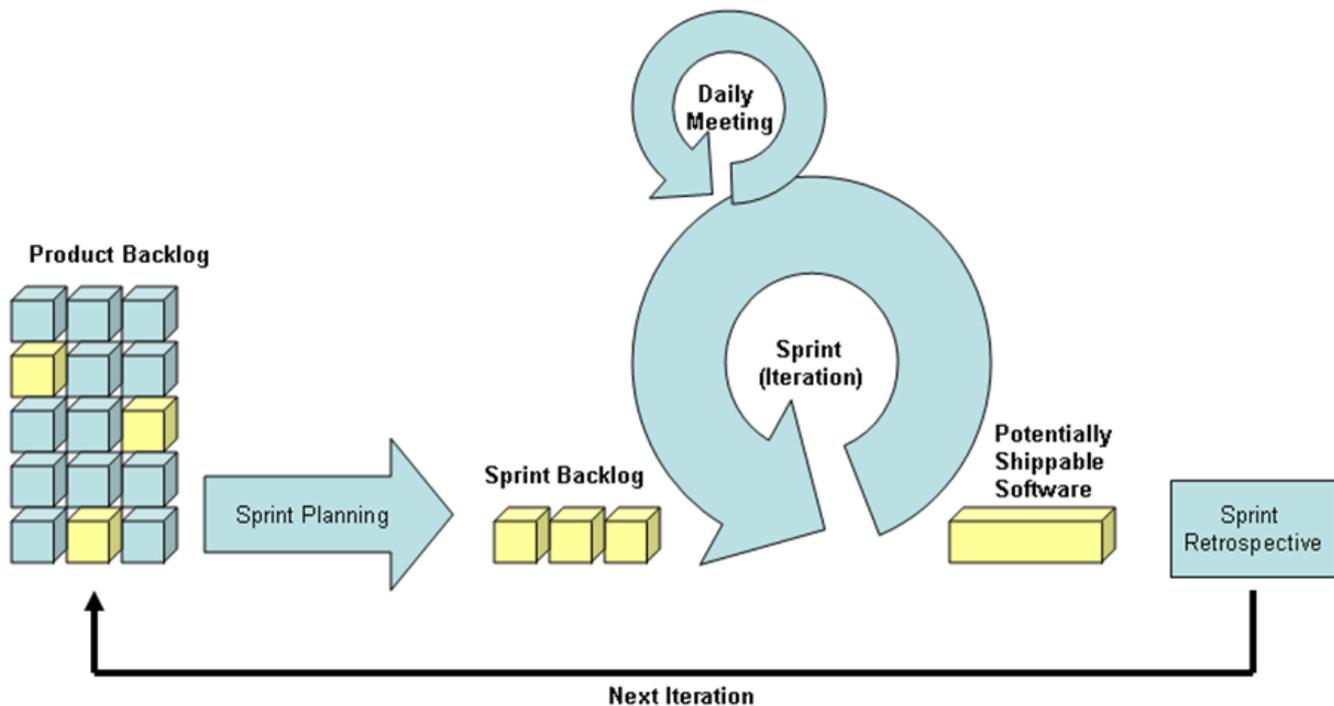


Figura 5. Proceso Scrum

3.3. Marco Ágil Escalado (SAFe)

El “Marco Ágil Escalado” (*Scale Agile Framework, SAFe*) describe una aproximación estructurada y escalable para implementar principios y prácticas de naturaleza *ágil* con equipos diversos que trabajen en proyectos complejos [25]. Los equipos están compuestos por individuos que disponen de la capacidad de llevar a cabo el análisis, el diseño, la fabricación y las pruebas de su trabajo. Operan en el desarrollo de sistemas a través de *Agile Release Trains* (ARTs), ventanas temporales definidas (típicamente de 8 a 12 semanas), durante las cuales múltiples equipos sincronizan sus cometidos con el objetivo de desarrollar y entregar una capacidad de valor a los *stakeholders*. De este modo, los ciclos ARTs se encuentran estrictamente encapsulados en el tiempo y se permite que su alcance varíe.

Los equipos planean las metas de desarrollo de un ART mediante un evento de planificación incremental conocido como *Program Increment* (PI). Las planificaciones PI reúnen a todos los equipos dentro de un ART para planificar y alinear sus trabajos para el subsiguiente PI. Durante una planificación PI, los equipos colaboran en definir objetivos, descomponer el trabajo, estimar el esfuerzo y establecer dependencias. SAFe incluye multitud de orientaciones

acerca de la gestión de un equipo de desarrollo y cómo éste debería ser organizado.

SAFe fue desarrollado para dar respuesta a las críticas acerca de la incapacidad de escalar metodologías *ágiles* como Scrum a proyectos de desarrollo de sistemas de mayor tamaño y complejidad. Lo hace mediante la aportación de guías para escalar metodologías *ágiles* desde el nivel de equipo hasta el de programa o el de portfolio. SAFe ofrece configuraciones diversas, que permiten a las organizaciones adaptar su implementación en base a sus necesidades específicas. Al proporcionar estas guías adicionales sobre cómo estructurar el desarrollo, SAFe es a veces criticado en el sentido de que haya podido sacrificar sus principios *ágiles* en pos de una mayor disciplina [31].

3.4. “Disciplined Agile Delivery” (DAD)

“*Disciplined Agile Delivery*” (DAD) es un método *ágil* que se construye sobre otros métodos como Scrum, pero que pretende cubrir el ciclo de vida completo, incluyendo por ejemplo la fase de Operaciones, a diferencia de otros enfoques *ágiles* que se focalizan prioritariamente en la de Desarrollo [1]. DAD define tres etapas: comienzo, elaboración y transición. Durante el comienzo, se identifica la visión del

proyecto, los *stakeholders* y los requisitos iniciales. En la de elaboración el proyecto desarrolla el sistema-solución sobre una base incremental. En la de transición, el proyecto sitúa el diseño en producción e involucra a los stakeholders en la validación. Durante la fase de elaboración, DAD aplica un enfoque de desarrollo iterativo e incremental, característico de los modelos *ágiles*.

DAD está principalmente orientado a proyectos de considerable envergadura haciendo consciente a la empresa, en el sentido de que DAD reconoce el contexto empresarial, y toma en consideración estándares de gobernanza, de conformidad y de carácter organizacional. DAD incide en la relevancia de validar la arquitectura en los primeros *sprints*, reduciendo en consecuencia el riesgo. DAD resulta particularmente apropiado en sistemas intensivos en software en los que el equipo de desarrollo pueda establecer y reaccionar ante bucles de realimentación entre operaciones y desarrollo (concepto *DevOps*, que será descrito más adelante).

3.5. Metodologías ágiles con Hardware

Las metodologías *ágiles*, originalmente ideadas para el desarrollo software, pueden encontrarse con ciertas dificultades a la hora de ser aplicadas al desarrollo hardware, debido a la inherente naturaleza física y a los aspectos de fabricación implicados. Agrupamos aquí estas restricciones porque aplican a todos los modelos *ágiles* previamente expuestos. Algunas de las restricciones que pueden limitar la aplicación de metodologías *ágiles* para desarrollo hardware son:

- **Plazos de ejecución mayores:** el hardware requiere a menudo plazos de ejecución más largos para el diseño, la compra de materiales y la fabricación de componentes en comparación con el desarrollo software. Esto puede representar un desafío a la hora de adherirse a los cortos ciclos iterativos típicamente asociados con los métodos *ágiles*.
- **Complejidad de fabricación:** fabricar hardware conlleva procesos complicados, utillaje especializado y medidas de control de calidad estrictas. Al contrario que el software, que puede ser *fácilmente* modificado y actualizado, hacer lo mismo con componentes hardware a menudo requiere de herramientas diversas, instalaciones, personal especializado, materiales, etc. Esto puede limitar la flexibilidad y la capacidad rápida de adaptación, que es un factor clave en las metodologías *ágiles*.
- **Iteraciones de mayor coste:** iterar sobre diseños hardware puede resultar caro, especialmente cuando implica utillaje, materiales y procesos de producción. Al contrario que el software, donde los cambios pueden ser realizados de manera relativamente sencilla y a un coste reducido, las iteraciones hardware requieren a menudo inversiones adicionales. Esto puede convertir a las numerosas iteraciones y a la experimentación en desafíos desde una perspectiva de coste y recursos.
- **Prototipos físicos:** el hardware requiere por lo general la creación de prototipos para pruebas y validación. La construcción de los mismos puede consumir mucho tiempo y ser costosa, lo cual puede impactar en la capacidad de iterar pronto y aprovechar ciclos de realimentación rápidos, como se hace normalmente en las metodologías *ágiles*.
- **Dependencias de la cadena de suministro:** el desarrollo hardware depende frecuentemente de complejas cadenas de suministro que involucran a múltiples proveedores y plazos de entrega para la compra de componentes y materiales. Esto introduce dependencias adicionales y retos en la gestión de planificaciones, coordinación y en asegurar la disponibilidad puntual de los recursos necesarios.
- **Cumplimiento de regulaciones:** los productos hardware necesitan cumplir a menudo con regulaciones y estándares industriales específicos. Asegurar este cumplimiento puede suponer amplias baterías de pruebas, procesos de certificación y documentación. Esto puede añadir complejidad y tiempo al proceso de desarrollo, impactando eventualmente en la agilidad de los ciclos de iteración.
- **Leyes de la física:** El funcionamiento y el rendimiento de los sistemas de hardware están limitados por las leyes de la física. Como resultado, generalmente existe una fuerte interconexión entre las variables de diseño y a través de las disciplinas de ingeniería. Esto hace que sea muy difícil asignar el trabajo en tareas independientes.

A pesar de estas restricciones, los principios y prácticas ágiles pueden aún ser aplicados sobre ciertos aspectos del desarrollo hardware. A modo de ejemplo, mientras que los *sprints* pueden no ser de tan corta duración como con el software, el principio de realizar frecuentes revisiones del trabajo y del alcance de las tareas puede ser útil a la hora de mantener un nivel creciente de productividad y de reaccionar rápidamente ante desalineamientos entre necesidades y soluciones. Esto ha sido demostrado en la práctica. Por ejemplo, Yang et al. [39] mostraron cómo la utilización de principios *ágiles* en el desarrollo de un producto hardware condujo a una mejora en la comunicación, a un riesgo reducido en el proyecto y a una mayor satisfacción del cliente. Análogamente, Thakurta et al. [35] evidenciaron que

el uso de estos principios en el desarrollo de un sistema embebido llevó a una colaboración de equipo mejorada, a ciclos de desarrollo más rápidos y a un menor riesgo del proyecto. Paasivara & Lassenius [29] describen la larga travesía de Ericsson en la adopción de metodologías *ágiles* para el diseño y desarrollo de sus productos.

De manera general, a pesar de que las restricciones físicas del desarrollo hardware plantean retos para la aplicación directa de enfoques *ágiles*, una adaptación cuidadosa y una aproximación a medida pueden contribuir a hacer uso de principios ágiles para mejorar la colaboración, la flexibilidad y la satisfacción del cliente en este tipo de proyectos.

4. COMPARATIVA ENTRE DESARROLLO EN BASE A UN PLAN VS. DESARROLLO ÁGIL

Esta sección compara la categoría de modelos *en base a un plan* frente a la categoría de modelos *ágiles*. Se considera primeramente cómo las dos categorías de modelos abordan la triple restricción de un proyecto: presupuesto, calendario y alcance.

Los modelos de desarrollo *en base a un plan* tienden a asumir una triada alcance-presupuesto-calendario fija (a menudo se implementan fijando tan solo el alcance), y únicamente responden de manera reactiva ante el surgimiento de problemas, debidos a la incertidumbre en los requisitos o a la aparición de riesgos técnicos. Por el contrario, el alcance, y en menor medida el presupuesto, se mantienen intencionadamente en flujo al utilizar modelos *ágiles*. Esto sucede como resultado de organizar el trabajo en *sprints* iterativos definidos en el tiempo, tal y como se explicó anteriormente, de modo que el proyecto puede adaptar sus tareas y objetivos según se considere necesario.

Conviene precisar que, si bien los modelos *ágiles* se implementan en general fijando presupuesto y calendario (permitiendo variar al alcance), y los métodos *en base a un plan* se implementan normalmente fijando el alcance (permitiendo al coste y al calendario cambiar), estas no son restricciones impuestas por ninguno de los métodos de desarrollo. De hecho, cada tipo de modelo de desarrollo puede implementar técnicas que apunten a prefijar una determinada dimensión; (ejemplo: 'Coste como Variable Independiente (CAIV)', que fija el coste a expensas de calendario y alcance [6]. La diferencia clave reside en si los cambios se producen de manera *reactiva* a los problemas o están *intencionadamente* embebidos dentro del proceso de desarrollo.

La Tabla 2 proporciona una sucinta comparación sobre cómo los modelos *en base a un plan* y los modelos *ágiles* acometen los aspectos fundamentales del entorno de negocio, el sistema a desarrollar, y la organización en la cual se aplican. La tabla no pretende ser exhaustiva, sino dar a entender en líneas generales las principales diferencias entre las dos clases de modelos. Como ejemplo, una consideración es el tipo de "Entorno de negocio" en el que la organización y el sistema operarán. Las organizaciones que trabajen en entornos regulados o que traten con sistemas críticos en seguridad necesitarán por regla general una mayor planificación, documentación y trazabilidad de requisitos. Los modelos *en base a un plan* afrontan estas demandas directamente. Sin embargo, esto no significa que los modelos *ágiles* no puedan ser utilizados en estos entornos. Por ejemplo, Hansen et al. [16] proponen una variación de Scrum, denominada Safe-Scrum, probando los principios *ágiles* en el desarrollo de sistemas críticos en seguridad tales como un sistema de aviónica, a pesar de la arraigada creencia de que el enfoque *ágil* resulta inconsistente con la necesidad de cumplir con rigurosas regulaciones de seguridad.

Factores Contingentes		Modelos en base a un plan	Modelos ágiles
Entorno de negocio	Estabilidad del Mercado y certidumbre en los requisitos.	Mejores con requisitos estables y no cambiantes.	Mejores en entornos dinámicos con requisitos cambiantes.
	Claridad y certeza en los requisitos.	Mejores cuando los requisitos son claros y bien conocidos.	Mejores para tratar con requisitos ambiguos y poco definidos.
	Entorno regulatorio o crítico en seguridad.	Mejores para cumplir con regulaciones y/o políticas estrictas; asegurando la trazabilidad mediante documentación y planes.	--
Sistema	Parte de un <i>Sistema de Sistemas</i> con interconexiones múltiples con otros sistemas diferentes.	Mejores para planificar y controlar los correspondientes interfaces.	--
	Elementos con largos plazos de entrega u obligación de planificar el uso de los recursos (ej. alcance de las pruebas).	Mejores para planificar la adquisición de tales elementos, alineando presupuesto, calendario y recursos.	Mayor riesgo de no poder disponer de tales elementos cuando sean necesarios, debido a una planificación limitada.
	Numerosos requisitos de calidad (<i>-ilities</i> : fiabilidad, mantenibilidad, ciberseguridad, etc).	Permiten identificar y definir estos requisitos pronto, lo cual resulta de utilidad dado que se les da respuesta a través de múltiples aspectos del diseño global del sistema.	Si estos requisitos se identifican durante <i>sprints</i> (especialmente si se trata de los últimos) -> riesgo de retrabajos para lograr que el sistema los cumpla, debido a que podría ser necesario cambiar múltiples aspectos del mismo.
	Riesgos tecnológicos.	Planes de maduración tecnológica previos a su inserción en el sistema.	Aprendizaje rápido y reducción de riesgos mediante iteraciones.
Organización	Equipos de trabajo dispersos geográfica u organizacionalmente.	La planificación y disciplina de estos modelos les permiten acomodarse bien a este tipo de escenarios.	Menor capacidad de gestionar estructuras organizacionales de este tipo salvo modificando algunos de sus principios (ej. dificultad de comunicaciones cara a cara). Mejores para equipos más reducidos y localizados

Tabla 2. Comparación de modelos en base a un plan y modelos ágiles

5. ADAPTACIÓN DE MODELOS DE DESARROLLO

Es conveniente tener presente que no existe un modelo *ideal* o preferido por defecto para el desarrollo de un sistema. Consideramos como mejor modelo de desarrollo de un sistema aquel que mejor conjugue:

- la organización, sus personas y su cultura;
- el sistema a desarrollar, su complejidad, su grado de conectividad con otros sistemas y el alcance de nuevas tecnologías que utiliza;
- el entorno de negocio en el que se ha desarrollado, el dinamismo del negocio y el grado de incertidumbre que rodea al desarrollo.

Por esta razón debería exigirse a menudo una adaptación (*tailoring*) de los modelos de desarrollo. La adaptación puede tomar la forma de establecer un modelo de desarrollo canónico y adaptarlo a las particularidades del proyecto, puede suponer una aproximación híbrida que aúne aspectos de diversos modelos de desarrollo o puede ser enfocado mediante la utilización de modelos de desarrollo diferentes para distintas partes del sistema (por ejemplo, adoptar un modelo “en V” para los componentes hardware y un modelo *ágil* para el software). La adaptación se sustenta en un profundo conocimiento por parte del equipo de ingeniería de sistemas de las ventajas e inconvenientes de cada modelo de desarrollo, y en la comprensión de los factores que afectan al desarrollo, como se ha mencionado antes (ej. entorno de negocio, sistema y organización).

Los enfoques *ágiles* han demostrado ser muy exitosos en la industria del software, sin embargo, existen limitaciones, como las anteriormente citadas, al aplicarlos al hardware. Por este motivo existe interés en las aproximaciones híbridas que fusionen modelos de desarrollo *ágiles* y tradicionales *en base a un plan*, para llegar a un equilibrio entre flexibilidad y necesidad de procesos de desarrollo rigurosos. Cuando se lleva a cabo una adaptación, es esencial abordar dos cuestiones [33]:

- 1) ¿Hasta qué grado se demanda agilidad por el mercado, las tecnologías implicadas y otros factores del entorno?
- 2) ¿Hasta qué nivel puede la organización ser *ágil*?

Básicamente, es importante coordinar las demandas o necesidades que el sistema debe satisfacer con la capacidad o viabilidad de la organización para adoptar un modelo de desarrollo particular [5]. Una adaptación, en este sentido, no debería verse como algo que ha de llevarse a cabo cuando ningún modelo *ideal* puede ser utilizado. Más bien, el *tailoring* debería ser concebido como un paso requerido a la hora de decidir qué tipo de modelo adoptar. Más aún, los modelos ideales deberían servir como paradigmas. En el mundo real, un modelo de desarrollo necesitará en general ser adaptado.

La Figura 6 refina la Tabla 2 al mostrar los modelos en base a un plan y los modelos *ágiles* no bajo una perspectiva dicotómica, sino como un continuo de adecuación entre los dos para diferentes variables.

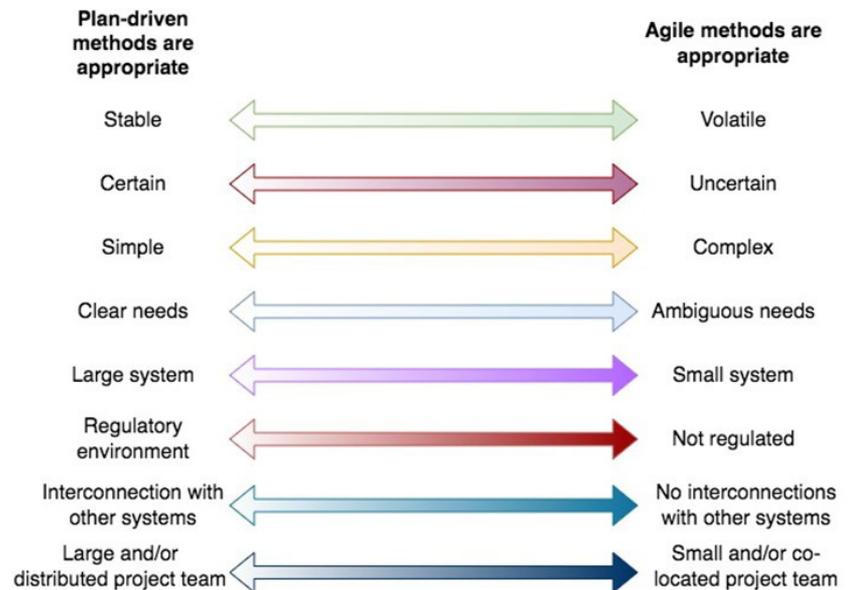


Figura 6. Factores para decidir entre un modelo en base a un plan o uno *ágil*

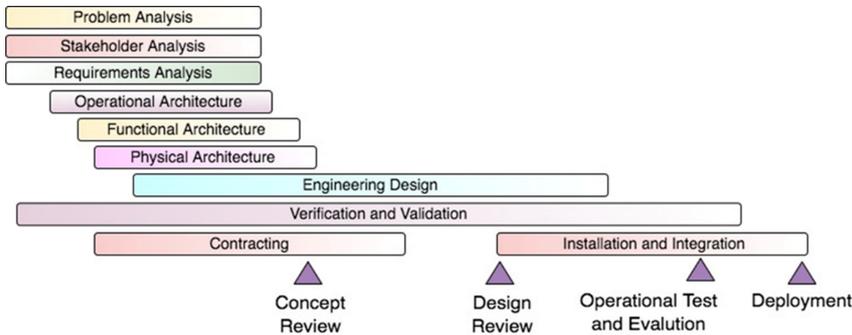


Figura 7. Planificación de alto nivel

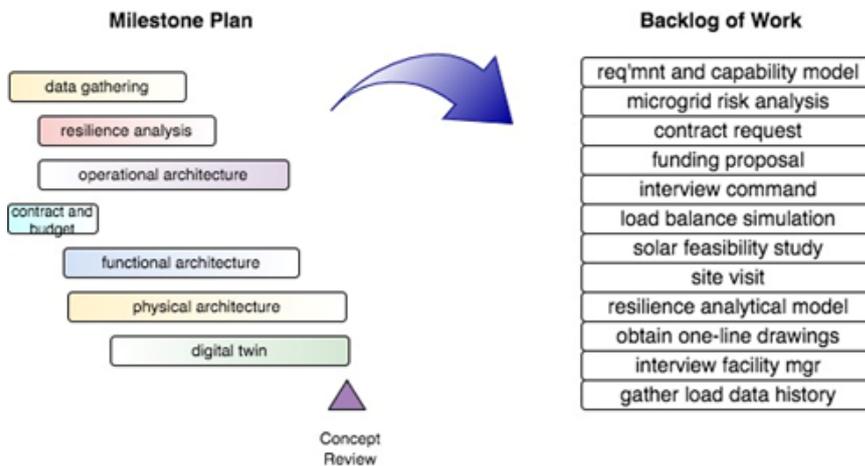


Figura 8. Planificación de un hito y reservorio (backlog)

Un medio para combinar modelos *en base a un plan* con principios ágiles es el de adoptar y adaptar conceptos ágiles relevantes a múltiples niveles de desarrollo. Un método híbrido puede evolucionar el sistema de forma iterativa e incremental de acuerdo con principios *ágiles*, al tiempo que mantiene algo de la predictibilidad disponible en los modelos *en base a un plan*.

Como ejemplo, considérense sistemas basados en hardware o intensivos en hardware. Este tipo de sistema requiere una cierta planificación porque el hardware a menudo incorpora partes con dilatados tiempos de entrega, no puede ser refactorizado, y los clientes desean conocer cuándo el sistema será abierto por primera vez. Adicionalmente, los sistemas más grandes poseen con frecuencia gran número de interfaces e interacciones con otros sistemas que deben ser planificadas y controladas. Las interacciones en este contexto incluyen también dependencias de prestaciones entre diferentes componentes. Estas problemáticas se abordan mediante macro-planificaciones del proceso de desarrollo global y utilizando una aproximación *top-down* que arranca con una arquitectura inicial del sistema. La Figura 7 refleja una vista de alto nivel de las actividades de desarrollo del sistema, y la Figura 8 muestra la planificación más detallada previa al siguiente hito. Ambas figuras ilustran que existe un amplio paralelismo de las actividades, pero no describen cómo la intensidad del esfuerzo varía con cada actividad. A modo de ejemplo, el análisis del problema podría darse durante un hito de "Revisión del concepto", pero la cantidad de horas de trabajo y esfuerzo serán superiores en etapas previas del proyecto, en comparación a justo antes de ese hito.

La Figura 7 ilustra cómo la Verificación y Validación (pruebas) ocurre *continuamente* a lo largo del proceso. Las pruebas frecuentes permiten una maduración permanente del diseño y una reducción del riesgo. Dentro de cada fase, existen iteraciones de comprensión, de diseño, de fabricación y de comprobación de ideas mediante el uso de modelos. Adicionalmente, existen bucles de realimentación e iteraciones entre fases. Por ejemplo, a medida que las capacidades son analizadas y definidas, el equipo podría repensar la manera en que plantearon el problema y revisar sus análisis asociados. Como resultado, este proceso analiza, diseña y evalúa de forma progresiva las necesidades de los *stakeholders*, los requisitos y la misión, para erigir la arquitectura de los productos.

6. FUSIÓN DE DESARROLLO Y OPERACIONES CON 'DEVSECOPS'

Los modelos de desarrollo de sistemas han sido tradicionalmente diseñados para concluir cuando se produce la transición del sistema a la fase de *Utilización* de su ciclo de vida. Aunque los sistemas pueden sin duda regresar a la fase de *Desarrollo* desde la de *Utilización*, los modelos de desarrollo tienden a asumir que, en tal caso, se especificaría un nuevo proyecto. La abrupta frontera entre *Desarrollo* y *Utilización* (también conocida como de *Operaciones*), está siendo recientemente puesta a prueba por un nuevo modelo de desarrollo denominado DevOps.

DevOps emergió de la comunidad de ingeniería software, con el objetivo de derribar las barreras encontradas a menudo entre los desarrolladores de sistema (Dev) y los operadores del sistema (Ops) [22]. DevOps implementa un ciclo continuo de desarrollar software, ensayarlo y sacarlo a operaciones, quienes lo utilizan, monitorizan sus prestaciones y proporcionan *feedback* a Desarrollo. DevOps se basa en contar con una comunicación abierta y cerrada entre desarrolladores y operadores, realimentación e integración continuas, y un flujo operacional constante.

DevOps ha sido y continúa siendo aplicado casi exclusivamente a sistemas intensivos en software. Los desafíos de adoptar DevOps al hardware son evidentes. Al contrario que el software, las nuevas características o funcionalidades no pueden ser trasladadas sobre una red y, además, los cambios a las prestaciones de un hardware conllevan habitualmente modificar el propio hardware. La implementación de DevOps ha supuesto un desafío incluso con software para sistemas embebidos [27]. Tan solo recientemente se está asistiendo a la aplicación de DevOps en dominios de sistemas que involucran tanto hardware como software, pero con una integración continua y la verificación llevándose a cabo únicamente sobre los componentes software [40].

En sistemas militares, la necesidad de *seguridad de la información* (*security*) es capital, y DevOps se ha transformado para incluir consideraciones de seguridad, creando un nuevo concepto de modelo conocido como DevSecOps [28]. Este enfoque supone integrar prácticas de seguridad de la información en cada etapa del proceso de desarrollo software, desde el diseño y generación de código hasta los ensayos y el despliegue, para asegurar que la seguridad constituye una parte del sistema desde su inicio. Al adoptar una aproximación DevSecOps, las organizaciones militares pueden contribuir a mitigar los riesgos de ciberataques, a proteger los datos y sistemas sensibles, y a asegurar que las operaciones militares no sean alteradas por vulnerabilidades o fallos del

software. De manera general, la metodología DevSecOps ha emergido como un enfoque clave para desarrollar sistemas software seguros, fiables y escalables, y es probable que su relevancia continúe incrementándose en entornos militares y otros de alto riesgo.

7. CONCLUSIONES

El capítulo ha analizado la necesidad y el rol que los modelos de desarrollo de sistemas cumplen en la ingeniería de sistemas. Se ha descrito el ciclo de vida de los sistemas como compuesto de seis fases: *Concepto, Desarrollo, Producción, Utilización, Soporte y Retirada*. El proceso de desarrollo de sistemas abarca principalmente las fases de *Concepto, Desarrollo y Producción* – aunque enfoques recientes buscan una mayor integración entre las de *Desarrollo y Utilización* (u *Operación*).

En líneas generales, los modelos de desarrollo pueden ser clasificados como estructurados *en base a un plan* o *ágiles*. Los modelos en base a un plan representan los enfoques de ingeniería de sistemas más tradicionales e incluyen los modelos “en cascada” y “en V”. Ambos se mueven predominantemente a lo largo de las actividades de desarrollo del sistema de una manera secuencial. Los proyectos que adoptan este tipo de enfoques comienzan con una planificación, y entendiendo generalmente todos los requisitos del sistema antes de avanzar a las actividades de diseño. Sin embargo, es aceptado desde hace tiempo que la implementación real de estos modelos implica extensas actividades iterativas y recursivas.

Los modelos *ágiles* han surgido mayoritariamente de la comunidad de la ingeniería del software. El software y su desarrollo difieren en varios aspectos del hardware, lo cual ayuda a explicar por qué las metodologías *ágiles* se popularizaron allí primero. Entre estas diferencias se encuentran la intangibilidad del software o que pueda ser puesto en funcionamiento sobre una red. Como consecuencia, el software es fácil y relativamente barato de modificar, incluso después de que se haya desplegado a los usuarios. Adicionalmente, el mercado software presenta un elevado dinamismo, con necesidades cambiantes y una frecuente introducción de nuevas tecnologías. Las metodologías *ágiles* explotan las características del software, facilitando cambiar para prosperar en un entorno de negocio tan dinámico. Los modelos *ágiles* están sustentados sobre principios de desarrollo iterativo e incremental, verificación y validación continuas, equipos auto-organizados, integración continua y una estrecha interacción con los usuarios. El modelo “en espiral” fue uno de los varios modelos que comenzaron a describir de una manera más formal y explícita el proceso de desarrollo de sistemas como iterativo e incremental. Otros novedosos modelos de desarrollo *ágil* que incorporan estos principios surgieron más tarde, tales como Scrum, SAFe y DAD.

El capítulo también ha proporcionado una comparativa entre los modelos *en base a un plan* y los modelos *ágiles* a lo largo de diferentes dimensiones, para resaltar sus fortalezas y debilidades. En general, los enfoques *en base a un plan* resultan adecuados para proyectos de desarrollo en los que los requisitos sean bien conocidos, la tecnología sea mayoritariamente madura y el entorno sea estable. También son útiles cuando sean necesarios registros de auditoría y documentación, como es el caso de sistemas críticos en seguridad o numerosos sistemas de defensa. Los modelos *ágiles*, por el contrario, son adecuados en entornos de gran incertidumbre y dinamismo. El capítulo señaló que muchos proyectos podrían presentar características adecuadas para modelos *en base a un plan* y otras características para modelos *ágiles*. En tales casos, se sugirió un modelo híbrido *en base a un plan* y *ágil*, intentando combinar las fortalezas de cada modelo de proceso. Tal adaptación de modelos de proceso es muy importante en la ingeniería de sistemas, puesto que no existe un modelo adecuado único para cada proyecto. Más bien, los ingenieros de sistemas informados y con conocimientos deberían adaptar o personalizar los modelos de desarrollo del sistema para adecuarlos a las necesidades de la organización y del proyecto de desarrollo.

El capítulo describió tendencias de desarrollo emergentes que afectan a la mayoría de los proyectos de desarrollo de sistemas, tales como DevOps. El concepto DevOps entrelaza el *Desarrollo* con *Operaciones*, haciendo uso de realimentación continua para la evolución del sistema a lo largo de su vida en servicio. DevOps, así como otras tendencias de desarrollo emergentes, puede ser incorporado a cualquier modelo de desarrollo de sistema y, ciertamente, los modelos de desarrollo pueden necesitar ser modificados para explotar mejor estas nuevas capacidades.

The image features a hand holding a smartphone in the lower foreground. The background is a blurred image of a person's face, overlaid with a grid of hexagonal icons. The central focus is a large white hexagon containing the text 'AGILE DEVELOPMENT'. Surrounding icons include a gear, a rocket, a lightbulb, a code editor, a briefcase, and a Wi-Fi symbol.

AGILE DEVELOPMENT

REFERENCES

1. Ambler, S. W., & Lines, M. (2012). *Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise*. IBM press.
2. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). *The agile manifesto*.
3. Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, 21(5), 61-72.
4. B. Boehm, *Spiral Development: Experience, Principles, and Refinements*, CMU/SEI-2000-SR-008, July 2000.
5. Boehm, B., and Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed*. Addison-Wesley Professional.
6. Brady, J. (2001). Systems engineering and cost as an independent variable. *Systems engineering*, 4(4), 233-241.
7. Chrissis, M. B., Konrad, M., & Shrum, S. (2011). *CMMI for development: guidelines for process integration and product improvement*. Pearson Education.
8. De Weck, O. L., Roos, D., & Magee, C. L. (2011). *Engineering systems: Meeting human needs in a complex technological world*. Mit Press.
9. Dingsøy, T., Falessi, D. and Power, K. (2019). "Agile development at scale: the next frontier". In: *IEEE software* 36.2 (2019), pages 30–38.
10. Douglass, B. P. (2015). *Agile systems engineering*. Morgan Kaufmann.
11. Dove, R., Lunney, K., Orosz, M., & Yokell, M. (2023, July). *Agile Systems Engineering—Eight Core Aspects*. In *INCOSE International Symposium* (Vol. 33, No. 1, pp. 823-837).
12. Elssamadisy, A., Abu-Tayeh, G., & Brown, T. (2018). *Scaling agile and lean development in the enterprise*. Pearson Education.
13. Evans, J. H. (1959). Basic design concepts. *Journal of the American Society for Naval Engineers*, 71(4), 671-678.
14. Forsberg and Mooz, 1998 à Forsberg, K. and Mooz, H. (1998), 7.17. *System Engineering for Faster, Cheaper, Better*. *INCOSE International Symposium*, 8: 917-927. <https://doi.org/10.1002/j.2334-5837.1998.tb00130.x>.
15. Gale, P.A., "The ship design process," in *Ship Des. Construction*, vol. 1, T. Lamb, Ed. Alexandria, VA: SNAME, 2003, ch. 5.
16. Hanssen, K. G., Stålhane, T., & Myklebust, T. (2018). *SafeScrum®—agile development of safety-critical software*. Springer Nature Switzerland AG.

17. Henning, W.A., and Walter, D.T., (2005). Spiral development in action: a case study of spiral development in the Global Hawk Unmanned Aerial Vehicle program. Master's Thesis. Naval Postgraduate School, Monterey, CA.
18. Jim Highsmith and Alistair Cockburn. "Agile Software Development: The Business of Innovation". In: Computer 34.9 (2001), pages 120–127.
19. Ibarra H., and Scoular, A. (2019). "The Leader as Coach". In: Harvard Business Review (November - December 2019).
20. INCOSE (2023). SE Vision 2035: Engineering Solutions for a Better World. International Council on SE. Retrieved September 1, 2023 from <https://www.incose.org/about-systems-engineering/se-vision-2035>.
21. ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes, 2015-05-15.
22. Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps handbook: How to create world-class agility, reliability (2nd ed.). IT Revolution Press. <https://itrevolution.com/book/the-devops-handbook/>.
23. Knaster R. and Leffingwell, D. (2017). SAFe 4.0 distilled: applying the Scaled Agile Framework for lean software and systems engineering. Addison-Wesley Professional.
24. Larman, C., & Vodde, B. (2016). Large-scale scrum: More with LeSS. Addison-Wesley Professional.
25. Leffingwell, D.: Scaling Software Agility: Best Practices for Large Enterprises. Pearson Education, Boston (2007).
26. Lockey, S., Gillespie, N., Holm, D., & Someh, I. A. (2021). A review of trust in artificial intelligence: Challenges, vulnerabilities and future directions. Proceedings of the 54th Hawaii International Conference on System Sciences.
27. Lwakatare, L. E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H. H., Bosch, J., & Oivo, M. (2016, January). Towards DevOps in the embedded systems domain: Why is it so hard? In 2016 49th Hawaii international conference on system sciences (hicss), pp. 5437-5446.
28. Miller, A. W., Giachetti, R. E., & Van Bossuyt, D. L. (2022). Challenges of Adopting DevOps for the Combat Systems Development Environment, Defense Acquisition Research Journal, 29(1).
29. Paasivaara M. and Lassenius, C. (2019). "Empower your agile organization: Community- based decision making in large-scale agile development at Ericsson". In: IEEE Software 36.2 (2019), pages 64–69.
30. Pahl G. and Beitz, W. (2013). Pahl/Beitz Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung. Springer-Verlag, 2013.
31. Putta, A., Paasivaara, M., & Lassenius, C. (2018). Benefits and challenges of adopting the scaled agile framework (SAFe): preliminary results from a multivocal literature review. In Product-Focused Software Process Improvement: 19th International Conference, PROFES 2018, Wolfsburg, Germany, November 28–30, 2018, Proceedings 19 (pp. 334-351). Springer International Publishing.
32. Scheithauer and Forsberg, 2013. à Scheithauer, D. and Forsberg, K. (2013), 4.5.3 V-Model Views. INCOSE International Symposium, 23: 502-516. <https://doi.org/10.1002/j.2334-5837.2013.tb03035.x>.
33. Stelzmann, E. (2012). Contextualizing agile systems engineering. IEEE Aerospace and Electronic Systems Magazine, 27(5), 17-22.
34. Schwaber 1997 à Schwaber, K. (1997). SCRUM Development Process. In: Sutherland, J., Casanave, C., Miller, J., Patel, P., Hollowell, G. (eds) Business Object Design and Implementation. Springer, London. https://doi.org/10.1007/978-1-4471-0947-1_11.
35. Thakurta, R., Mukhopadhyay, D., & Das, D. (2020). Applying agile practices in embedded system development: An industrial case study. Journal of Systems and Software, 168, 110660.
36. Ullman, D. G. (2019). Scrum for hardware design: supporting material for the mechanical design process.
37. Wade A Henning and Daniel T Walter. Spiral development in action: a case study of spiral development in the Global Hawk Unmanned Aerial Vehicle program. Technical report. Naval Postgraduate School, Monterey, CA, 2005.
38. Willett, K. D., Dove, R., Chudnow, A., Eckman, R., Rosser, L., Stevens, J. S., ... and Yokell, M. (2021, July). Agility in the Future of Systems Engineering (FuSE)-A Roadmap of Foundational Concepts. In INCOSE International Symposium (Vol. 31, No. 1, pp. 158-174).
39. Yang, J., Lappas, T., Egan, M., & Bagaria, N. (2019). Applying Agile methods to hardware product development. Journal of Systems and Software, 154, 1-14.
40. Zaeske, W., & Durak, U. (2022). DevOps for Airborne Software: Exploring Modern Approaches. Springer Nature.

BIOGRAFÍAS

DR. RONALD GIACHETTI

El Dr. Ronald Giachetti es catedrático de Ingeniería de Sistemas en la Escuela Naval de Postgrado (NPS) de Monterrey, California. Enseña e investiga sobre ingeniería de sistemas basada en modelos, arquitectura de sistemas e ingeniería de sistemas. Aporta 30 años de experiencia en ingeniería tanto en el ámbito académico como en el industrial. Autor de más de 50 artículos técnicos publicados, incluido un libro de texto sobre el diseño de sistemas empresariales. Ha sido conferenciante en diversos centros de Europa, América Latina y Asia. Ha ocupado distintos cargos directivos, entre ellos el de director del departamento de ingeniería de sistemas, decano de la Escuela de Postgrado de Ingeniería de la NPS y presidente del Consejo Asesor Corporativo (CAB) del Consejo Internacional de Ingeniería de Sistemas (INCOSE). Recibió el título de ingeniero por el Instituto Politécnico Rensselaer, la Universidad Politécnica de Nueva York y la Universidad Estatal de Carolina del Norte.

Fuera del ámbito de la ingeniería, es un ávido navegante que participa activamente en regatas locales y nacionales.



JUAN CARLOS LARIO MONJE

Juan Carlos Lario Monje es ingeniero aeronáutico con una amplia experiencia multidisciplinar en diversos campos de la ingeniería tales como safety, calidad, programas, offsets, contratos, definición y gestión de requisitos, integración, V&V de sistemas (verificación y validación), certificación y calificación. Con los años y la experiencia, ha adquirido una perspectiva global en programas aeronáuticos internacionales (civiles/militares), desde su fase de desarrollo hasta su posterior producción en serie y puesta en servicio final, con implicación específica en aerodinámica, guerra electrónica y sistemas electro-ópticos. A través de su participación directa en los principales programas aeronáuticos europeos: civiles (A380, A350) y militares (A400M, EF2000, Tiger-HAD, NH90), ha adquirido profundos conocimientos de las principales empresas de la industria aeronáutica española (Airbus, Indra), de los usuarios finales de los programas de defensa (Ejército de Tierra, Ejército del Aire) y de las agencias de contratación europeas (NETMA, OCCAR). Hace gala de una curiosidad inagotable y es un profesional entusiasta del aprendizaje y la enseñanza continuos. Certificado CSEP por INCOSE.





"Todos los modelos son incorrectos, algunos tienen utilidad."

G.E.P. Box.

Ingeniería de Sistemas Basada en Modelos

David Long, *Blue Holon* (david.long@incose.net)

Dr. Kaitlin Henderson, *Radiance Technologies* (kaitlin.henderson@radiancetech.com)

Belinda Misiego, *Isdefe* (bmisiego@isdefe.es)

CAPÍTULO 4

Resumen

Este capítulo presenta la Ingeniería de Sistemas Basada en Modelos (MBSE), como una evolución de la disciplina que aprovecha el potencial de los modelos digitales. El capítulo identifica los elementos principales que debe contener un modelo para tener éxito y desmitifica algunos conceptos erróneos comunes que conducen a prácticas débiles y/o deficientes en las organizaciones. Tras presentar algunas capacidades sin precedentes de la ingeniería de sistemas al adoptar enfoques basados en modelos, el capítulo proporciona algunas orientaciones para facilitar la adopción exitosa de MBSE. Los temas principales del capítulo son los efectos de la formalización de la ingeniería de sistemas, la divergencia y convergencia de la semántica y la creación, revisión y control de la configuración en entornos basados en modelos.

Palabras clave

Ingeniería de sistemas basada en modelos (MBSE), ontología, metamodelo, lenguaje de modelado de sistemas (SysML).



1. INTRODUCCIÓN

Ingeniería de Sistemas Basada en Modelos (conocida mundialmente por las siglas MBSE, procedentes de *Model-Based Systems Engineering*), ingeniería basada en modelos (MBE, *Model-Based Engineering*), gemelo digital, ingeniería digital, transformación digital... toda una explosión de conceptos y siglas que demuestran la aplicación de lo digital a la ingeniería de sistemas y al ciclo de vida de la ingeniería en general. Este capítulo intentará aclarar cuáles son las diferencias entre los distintos conceptos y cómo se interrelacionan. Irá más allá del marketing, de los mitos y las conjeturas, y tratará de avanzar hacia una comprensión práctica de lo que significa la transformación digital para la ingeniería de sistemas, los fundamentos que se deben conocer y el valor esperado que se debe lograr.

Antes de profundizar en las discusiones sobre qué es MBSE, se presenta el contexto del que surgió. En las últimas décadas, se ha visto una evolución en el desarrollo y la capacidad de los productos. Primero se pasó de productos básicos basados en tecnologías electromecánicas a productos inteligentes que aprovechaban la electrónica y el software para brindar nuevas capacidades y una experiencia de usuario mejorada. Luego se introdujeron sensores y redes para crear productos inteligentes y conectados con capacidades tales como saber exactamente cuándo llegará el próximo tren y cuántos asientos quedan disponibles. Después, estos productos necesitaron estar cada vez más conectados hasta convertirse en un sistema de productos, con todos los elementos coordinados. Hoy en día, los sistemas tradicionales e inteligentes están colaborando como sistemas de sistemas (SoS, *Systems of Systems*) para satisfacer las necesidades de la sociedad (por ejemplo, operaciones aeroportuarias en las que el sistema de información de vuelos, el sistema de transporte, el sistema de seguridad operativa y todos los demás sistemas involucrados colaboran de manera coordinada y conectada).

En la época de los productos electromecánicos, las interacciones entre piezas, componentes y sistemas eran algo limitadas. Teniendo en cuenta el bajo acoplamiento entre los componentes y los equipos de diseño que los creaban, los documentos representaron la solución más adecuada para capturar datos de arquitectura y diseño. En un mundo basado en documentos, los datos están dispersos en una multitud de documentos. Sin embargo, a medida que se introducen nuevas tecnologías y los sistemas se vuelven más grandes y complicados, el número y la complejidad de las interfaces aumentan considerablemente. Los datos dispersos entre artefactos podrían generar inconsistencias entre ellos. Además, conseguir mantener actualizados cientos de documentos para un sistema complejo puede ser una tarea ardua. Pueden ocurrir inconsistencias en el diseño debido a discrepancias interpretativas de la información

debido al uso de enfoques, lenguaje o diagramas sin una semántica común entre las partes interesadas. Descubrir inconsistencias en el diseño (y, por lo tanto, reelaboraciones significativas) puede generar posibles retrasos y costes adicionales en el desarrollo, y las inconsistencias no descubiertas pueden dar como resultado fallos en el sistema [1].

Al observar el progreso tecnológico en rápida evolución en el mundo, queda claro que las necesidades de los procesos de diseño de ingeniería han excedido lo que las capacidades de la ingeniería de sistemas tradicional pueden proporcionar. De hecho, en 2014 INCOSE publicó la “Visión de Ingeniería de Sistemas 2025” (INCOSE, 2014), con una declaración de hacia dónde tenía que moverse la industria para resolver las necesidades de la sociedad. La publicación destacó una serie de desafíos, entre ellos:

- La complejidad de la misión está creciendo más rápido que la capacidad para gestionarla.
- El diseño del sistema surge de las piezas, más que de la arquitectura.
- El conocimiento y la inversión se pierden en los límites de las fases del ciclo de vida del proyecto.
- El conocimiento y la inversión se pierden entre proyectos

Lo que funcionó para los sistemas electromecánicos en las décadas de 1950, 1960 y 1970 no es suficiente para abordar las necesidades y tecnologías actuales. Los ingenieros de sistemas comenzaron a dar el salto a enfoques basados en modelos para responder con agilidad y eficiencia al mundo complejo y cambiante.

En 2021, INCOSE publicó la “Visión de Ingeniería de Sistemas 2035” [3]. Uno de los titulares de esta publicación es “El futuro de la ingeniería de sistemas se basa predominantemente en modelos”. Se espera que en 2035 exista una familia de marcos integrados y unificados para el modelado y la simulación de sistemas.

Pero ¿qué es MBSE? El significado de MBSE es muy confuso, todo un desafío por la amplitud y ambigüedad de lo que constituye un modelo. Una definición técnica de modelo es “Una representación física, matemática o lógica de otro modo de un sistema, entidad, fenómeno o proceso” [4]. Una definición más amplia de modelo es “una representación gráfica, matemática (simbólica), física o verbal o una versión simplificada de un concepto, fenómeno, relación, estructura, sistema o un aspecto del mundo real”. Una representación de dinámica de fluidos computacional del aire que fluye sobre un ala de un avión satisface estas definiciones, pero también lo hace un dibujo del flujo de un proceso y un documento de control de interfaz

tradicional. Debido a la amplitud de lo que constituye un modelo, podría haber tantas interpretaciones del significado de modelo como profesionales. La mayoría de la ellos citarán la definición de MBSE de INCOSE, que es “la aplicación formalizada de modelado para respaldar las actividades de requisitos, diseño, análisis, verificación y validación de sistemas comenzando en la fase de diseño conceptual y continuando en la etapa de desarrollo y etapas posteriores del ciclo de vida”.

En pocas palabras, MBSE representa un nuevo método para realizar ingeniería de sistemas; uno que utiliza técnicas basadas en modelos para realizar tareas de ingeniería de sistemas en lugar de las tradicionales basadas en documentos [5]. Se trata de pasar de capturar datos en documentos en lenguaje natural a representaciones más ricas que reflejen y comuniquen los fenómenos, la arquitectura y el diseño de sistemas, idealmente en un formato legible por una máquina. El artefacto central es el modelo del sistema, del cual se derivan todos los demás artefactos. Sirve como tejido conectivo que une la gran empresa digital que permite la ingeniería. Si se pudiera retroceder en el tiempo para asignar un nuevo nombre a MBSE, lo más probable es que se denominase ingeniería de sistemas digitales.

MBSE debería hacer que los modelos analíticos y descriptivos del sistema sean explícitos, coherentes, consistentes y ejecutables, tanto por humanos como por computadoras. Debería reflejar una evolución desde representaciones de baja fidelidad en documentos hacia representaciones más ricas y de mayor fidelidad que las máquinas puedan leer e interpretar. Debería mejorar la granularidad de la captura de conocimientos para la gestión, el análisis y el aprendizaje del conocimiento. Debería permitir un modelo arquitectónico descriptivo que conecte múltiples modelos analíticos, que juntos representen el diseño con el grado de rigor requerido para el problema y la solución en cuestión.

MBSE también debería aprovechar los modelos para la comunicación y el análisis, representar “datos autorizados” para el diseño y las especificaciones del sistema, garantizar un diseño y especificaciones consistentes y proporcionar un modelo de sistema explícito a los equipos de ingeniería. En resumen, MBSE debería ser una evolución, no una revolución, en el pensamiento y el enfoque que aprovechan las tecnologías y capacidades modernas para representar mejor los datos, la información y el conocimiento. Si bien es evolutivo, este cambio ofrece resultados transformadores. Se espera que MBSE mejore la calidad del sistema, reduzca costes, acorte los tiempos de desarrollo, integre nuevas tecnologías y brinde continuidad digital con la fabricación y las operaciones.

2. ELEMENTOS NECESARIOS (E IDEALES) DE MBSE

Existe una diferencia fundamental entre los modelos en ingeniería de sistemas y la ingeniería de sistemas basada en modelos. Los ingenieros siempre han utilizado modelos para comprender y razonar. El resultado de la ingeniería siempre ha sido algún tipo de modelo que permite analizar y mejorar la comprensión de un problema. La diferencia está en que MBSE usa modernas técnicas para la captura y representación de la información fundamental necesaria para diseñar un sistema. Tradicionalmente ha estado más en el lado descriptivo y arquitectónico, aunque no se limita inherentemente a ellos, extendiéndose desde una primera expresión de necesidad a lo largo de todo el ciclo de vida del sistema. Estos modelos permiten obtener, capturar y representar un sistema para que se puedan utilizar mejor las técnicas computacionales (modelos analíticos, modelado y simulación, etc.) que siempre se han usado y que probablemente se continuarán usando.

En el mundo del diseño de ingeniería, los modelos arquitectónicos conectan la idea detrás de una solución de diseño con su implementación como un sistema real. Es la forma en la que se alinean actualmente los equipos de trabajo, ya que permite comunicar con una mayor conciencia para minimizar malentendidos, así como capturar el conocimiento a lo largo del tiempo. Estos modelos intentan representar las entidades del problema de ingeniería y su relación entre sí y conectarlas con la solución propuesta o el mecanismo existente que aborda el problema. Si se consigue caracterizar adecuadamente cada parte del sistema y sus interfaces, tanto funcional (es decir, lo que hace el sistema) como físicamente, así como las interacciones e intercambios entre ellos, entonces se podrá, con un alto grado de fidelidad y precisión, pasar a una descripción detallada de los componentes. El modelo utilizado de esta manera es la pieza central de MBSE.

Cuatro elementos son críticos para un modelo [6]:

- **Lenguaje:** El lenguaje de modelado permite la expresión y representación clara del modelo, de modo que puedan surgir la comprensión y el conocimiento. El lenguaje debe ser claro e inequívoco para representar el modelo de forma precisa y comprensible.
- **Estructura:** La estructura permite que el modelo capture el comportamiento del sistema al describir claramente la relación entre las entidades de este.
- **Argumentación:** Un modelo debe ser capaz de presentar el “argumento” crítico de que el sistema satisface las necesidades de las partes interesadas. El modelo debe representar el diseño del sistema de tal manera que el equipo de diseño pueda demostrar que el sistema logra los propósitos para los cuales fue diseñado.

- **Presentación:** un modelo debe incluir algún mecanismo para presentar el argumento de manera que pueda ser visto y comprendido por los usuarios, que pueden incluir ingenieros, clientes y otras partes interesadas.

Estos elementos se pueden definir mediante el uso de metamodelos, semántica y ontologías. Los metamodelos, la semántica y las ontologías construyen las formalidades subyacentes que hacen posible muchos de los beneficios clave de MBSE (por ejemplo, colaboración, intercambio de modelos, reutilización, reducción de ambigüedad, etc.). Un metamodelo define la sintaxis, las restricciones y los patrones que conforman el lenguaje de modelado que se utiliza al crear un modelo. Se puede considerar una ontología como un tipo de metamodelo que define un conjunto común de terminología y relaciones para un dominio determinado, incorporando una semántica precisa a los términos y relaciones.

La Figura 1 muestra un ejemplo de parte de un metamodelo para ingeniería de sistemas. Capta (y limita) los diferentes conceptos que el ingeniero de sistemas puede utilizar

para respaldar su trabajo, alineando la comprensión y la interpretabilidad que tiene cada miembro del equipo. Por ejemplo, un ingeniero no podría distinguir entre un requisito y una especificación, ya que el metamodelo no ofrece conceptos diferentes para esos dos términos. Al mismo tiempo, el metamodelo no permite que los componentes intercambien directamente entradas y salidas, sino que obliga al ingeniero a pensar en las funciones que realiza el componente para pensar en entradas y salidas. El impacto de esto no es sólo el de fomentar la comprensión y garantizar ciertas buenas prácticas al razonar sobre la información de ingeniería de sistemas, sino que permite construir modelos legibles por máquinas para permitir un mejor acceso y procesamiento de dicha información. En este ejemplo de metamodelo, se podría ver cómo un ingeniero podría consultar el modelo subyacente para identificar todos los requisitos de verificación relacionados con una función (a través de una ruta relacional a través de requisitos, componentes y finalmente funciones), o hacer cumplir reglas para garantizar que cada componente debe ejecutar una función, o cada función, debe tener al menos una entrada y una salida para ser consistente con la teoría de sistemas.

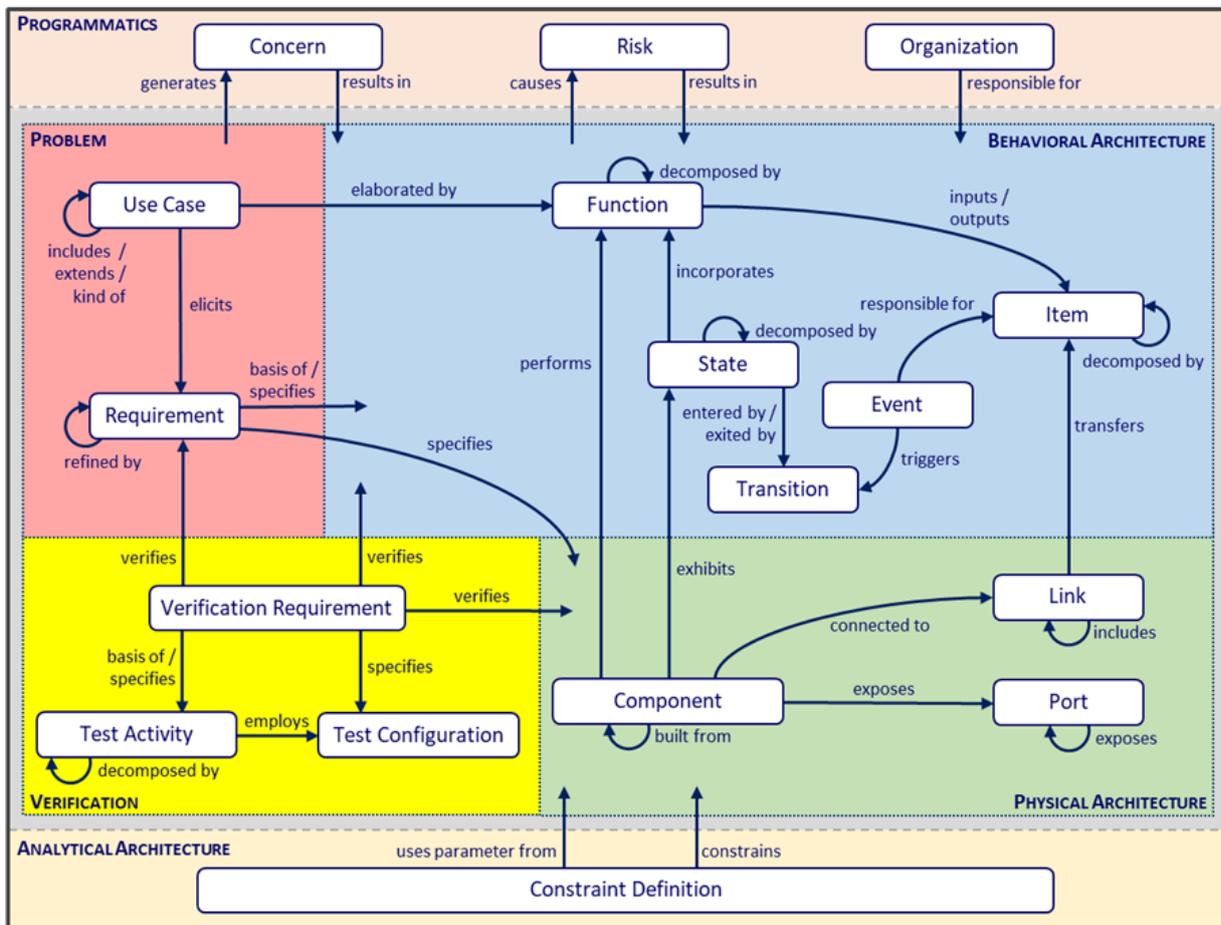


Figura 1. Ejemplo de metamodelo

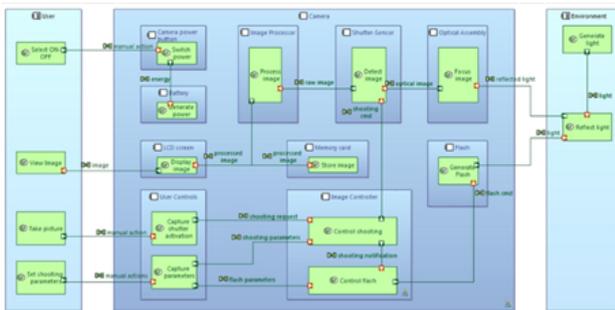
Agregar una semántica precisa a un metamodelo permite el uso de axiomas que brindan capacidades de razonamiento. Mientras que un metamodelo normal sólo puede garantizar la corrección sintáctica, un ingeniero puede aprovechar la semántica de un modelo para inferir aspectos del modelo que se derivan del significado incorporado en el modelo. Por ejemplo, mientras que un metamodelo normal sólo puede comprobar que cada “componente” ejecuta una “función”, una ontología ofrece la posibilidad de comprobar que una instancia específica de una instancia de función realmente representa una función. Así, si un ingeniero definiese la *serpiente* como un “componente” y la *mosca* como una “función”, podría crear la relación en la que la *serpiente* realiza la función de la *mosca*. Esto es sintácticamente correcto y un metamodelo normal lo consideraría válido. Sin embargo, una ontología podría identificar que el modelo no es sensato porque las serpientes no pueden volar.

No existe un diseño único para metamodelos u ontologías. Cada organización elige lo que se adapta mejor a su contexto y necesidades específicas. Sin embargo, el metamodelo y/o la ontología no deben ser arbitrarios. Los conceptos, relaciones, semántica y axiomas deben ser significativos, internamente consistentes e idealmente consistentes con la teoría y los principios de la ingeniería de sistemas, si uno quiere beneficiarse de su estructura y legibilidad por máquina (si se implementa como tal).

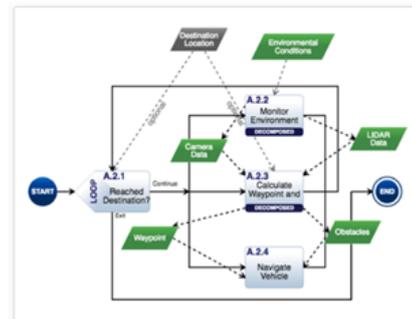
George Box dijo la famosa frase: “Todos los modelos son incorrectos, pero algunos son útiles”. La pregunta es ¿cómo de equivocado puede ser un modelo para seguir siendo útil? Siempre que se trata con modelos se debe definir y comprender el propósito detrás de ese modelo para que el modelo sea adecuado para su propósito en su tipo, alcance y nivel de fidelidad. MBSE puede ser válido en cualquier punto del ciclo de vida del sistema, siempre que la inversión de esfuerzo en el modelo esté alineada con el propósito apropiado. Por ejemplo, si el modelo se utiliza al principio del ciclo de vida, es probable que no se opte necesariamente por una representación de alta fidelidad. En lugar de precisión, es posible que se esté buscando una mejor comprensión y alineación con las partes interesadas para obtener las necesidades reales en la fase de definición de la arquitectura. Ya sea que se trate de una nueva reingeniería de diseño, un ciclo de vida temprano o un ciclo de vida tardío, si se identifica el propósito y se mantiene en primer plano, entonces se podrá ofrecer valor a través de MBSE. Si no se hace, se puede caer en la trampa de modelar por modelar, lo que frustra el propósito de MBSE. La utilidad de un modelo está claramente ligada a la comprensión del propósito previsto.

La implementación de MBSE requiere de dos pilares además del lenguaje de modelado, la herramienta que permite la creación y visualización de los modelos y el método con el que se implementa MBSE [5].

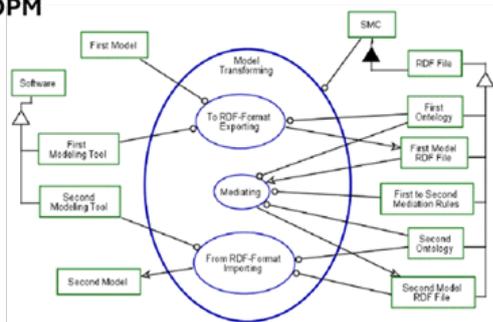
ARCADIA



LML



OPM



SysML/SDL

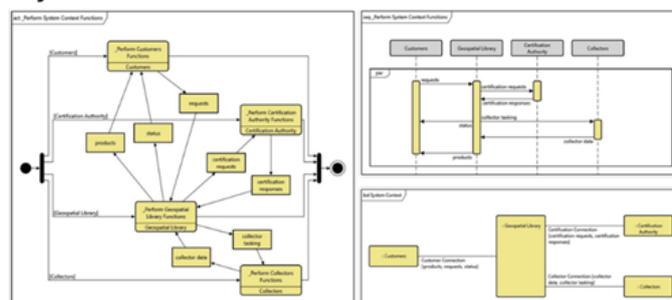


Figura 2. Ejemplos de modelos de sistemas creados en diferentes lenguajes de modelado

SysML (*System Modeling Language*) es el lenguaje dominante asociado a MBSE, pero es importante señalar que SysML no es igual a MBSE. De hecho, SysML no es el único lenguaje ni siempre es el lenguaje correcto. SysML fue desarrollado como un perfil del Lenguaje Unificado de Modelado (UML, *Unified Modeling Language*) desarrollado por la comunidad de ingeniería de software para ayudar a cerrar la brecha de comunicación entre los ingenieros de sistemas y de software durante el auge de los sistemas intensivos en software. En este sentido, SysML es un lenguaje gráfico de propósito general destinado a modelar sistemas, no necesariamente a modelar todos los aspectos de la ingeniería de sistemas, a pesar de que la comunidad practicante abusa de él en este sentido. Otros lenguajes de modelado incluyen el lenguaje de modelado del ciclo de vida (LML, *Lifecycle Modeling Language*), la metodología de proceso de objetos (OPM, *Object Process Methodology*) y varios lenguajes vinculados a herramientas específicas como Capella/Arcadia y el lenguaje de definición de sistemas de Vitech Corporation (SDL, *Vitech Corporation's Systems Definition Language*). Algunas representaciones gráficas se muestran como ejemplos en la Figura 2.

Las herramientas de modelado son una clase especial de herramientas que están diseñadas e implementadas para cumplir con las reglas de uno o más lenguajes de modelado y permitir a los usuarios construir modelos bien formados en estos lenguajes. Las buenas herramientas MBSE son mucho más que herramientas que hacen diagramas. Los diagramas no son el modelo en sí; son simplemente vistas del modelo subyacente que contiene un conjunto de elementos y relaciones que se muestran en los diagramas. Así como una herramienta de diseño asistido por computadora (CAD, *Computer-Aided Design*) puede presentar vistas superior, frontal y lateral del modelo geométrico de una pieza, las herramientas MBSE producen una variedad de visualizaciones del modelo del sistema subyacente. Algunos ejemplos de herramientas MBSE incluyen CapellaTM (código abierto Eclipse), Cameo Systems ModelerTM (Dassault Systems), Enterprise ArchitectTM (Sparx Systems), InnoslateTM (SPEC Innovations), RhapsodyTM (IBM) y GENESYSTEM (Vitech Corporation). En la Figura 3 se muestran ejemplos de algunas de sus interfaces de usuario.

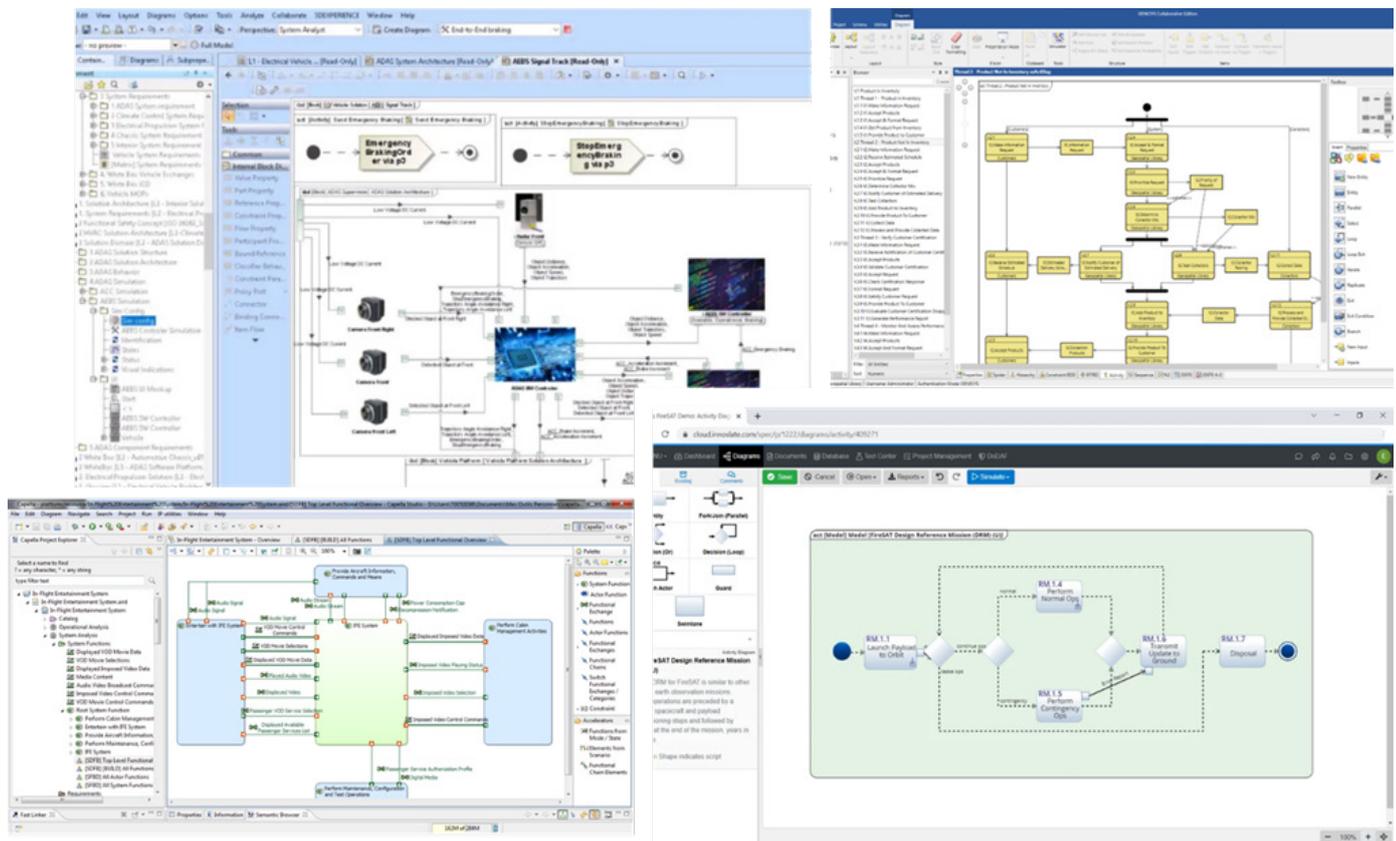


Figura 3. Ejemplos de interfaces de usuario de diferentes herramientas MBSE

Los métodos y/o metodologías de modelado pueden verse como una hoja de ruta con un conjunto de tareas para crear el modelo que garantiza que todo el equipo construya el modelo de manera consistente y trabaje hacia un propósito común. El método ayuda a definir el alcance, la profundidad y la amplitud del modelo (basado también en la opinión de los equipos de desarrollo y el tiempo disponible). Algunas metodologías generales incluyen el método INCOSE de ingeniería de sistemas orientado a objetos (OOSEM, *Object-Oriented Systems Engineering Method*), la caja de herramientas de modelado de sistemas (SYSMOD, *System Modeling Toolbox*), la metodología de proceso de objetos (OPM, *Object-Process Methodology*) y la ingeniería de sistemas integrados y canalizaciones de procesos en arquitecturas orientadas a objetos (ISE&PPOOA, *Integrated Systems Engineering and Pipelines of Processes in Object-Oriented Architectures*). Cabe señalar que estas metodologías, si bien se desarrollaron con cierto nivel de generalidad, pueden no ajustarse a las necesidades de todas las organizaciones. De hecho, la adopción de una metodología MBSE particular requiere adoptar un enfoque específico de la ingeniería de sistemas. Por lo tanto, una organización puede implementar metodologías a medida para alinear mejor la implementación de MBSE con sus procesos de trabajo y la adopción específica de prácticas de ingeniería de sistemas. Además, debido a esta fuerte conexión entre las metodologías MBSE y las prácticas de ingeniería de sistemas, la coherencia con los estándares de ingeniería de sistemas puede limitar la implementación de MBSE o viceversa, la adopción de una metodología MBSE específica puede requerir el cambio o adaptación de los estándares existentes.

Finalmente, cabe resaltar que las personas son fundamentales para implementar MBSE. Al formar un equipo para realizar MBSE, es necesario que el equipo tenga experiencia tanto en ingeniería de sistemas, como en los lenguajes y las herramientas de modelado elegidas, además de cualquier otra experiencia específica requerida para completar el proyecto en cuestión (por ejemplo, diferentes métodos analíticos, dominios de aplicación, etc.). Algunos equipos optan por dividir la experiencia entre diferentes miembros del equipo, mientras que otros optan por garantizar que todos los miembros del equipo tengan experiencia en todas las áreas. Cada enfoque tiene ventajas y desventajas (por ejemplo, dividir la experiencia acelera la curva de aprendizaje, pero es frágil y menos escalable). No se sabe cómo evolucionará la práctica de MBSE en los próximos 15 años, lo único que se sabe es que en estos momentos es muy fluida y volátil. Por tanto, lo único que se puede sugerir es buscar un equipo que tenga las características adecuadas para su propósito dadas sus necesidades comerciales y el contexto de aplicación.

3. MODELOS, MÁS QUE SIMPLES DIBUJOS

Cuando alguien que no está familiarizado con la ingeniería de sistemas basada en modelos, mira una herramienta MBSE, ve una colección de dibujos y se pregunta en qué se diferencia de usar otras herramientas tales como Microsoft PowerPoint o Microsoft Visio. El poder detrás de MBSE proviene de las estructuras de datos subyacentes, la sintaxis, los lenguajes, etc. que conforman el modelo real. Lo que se ve cuando se mira un diagrama se considera una representación o vista del modelo del sistema, como se ha explicado anteriormente. La capacidad de realizar diagramas de MBSE es una gran herramienta para comunicarse con las diferentes partes interesadas, pero lo que realmente lo hace basado en modelos es la conectividad entre los diferentes componentes que están representados sin ambigüedades debido a la sintaxis y semántica definidas del lenguaje de modelado.



Figura 4. Modelo Iceberg de lo que un usuario ve en un diagrama versus la información contenida en el modelo

Un ingeniero de sistemas tradicional suele decir que una pizarra es una de sus mejores herramientas. Puede ir a la pizarra, comenzar un boceto, dibujar burbujas, nubes y flechas, y etiquetar los componentes. Esto es perfecto para trazar el alto nivel del sistema y permite la participación del equipo humano. Sin embargo, a medida que se avanza hacia un diseño más detallado, esa flexibilidad en el lenguaje y la simbología se convierte en un obstáculo porque se necesita una mayor precisión. Es necesario acordar el significado de los símbolos y la terminología para evitar errores de comunicación y desalineaciones. Es necesario que haya rigor en el lenguaje para garantizar una comunicación y un razonamiento efectivos tanto por parte de humanos como de computadoras. La flexibilidad puede ser útil para comprender conceptos en el espacio del problema, pero si se desea reutilizar componentes y, eventualmente, conectar modelos con otras herramientas, es necesario que haya precisión y rigor.

Discutir la precisión y coherencia de las terminologías definidas requiere volver a las ontologías. Establecer una ontología es fundamentalmente decir que se han definido las palabras, las interrelaciones entre las palabras, los conceptos que encarnan y el contexto dentro del cual son válidos. La ontología es la arquitectura subyacente que permite capturar datos individuales sin ambigüedades y reflejar las interrelaciones para representar información y conocimiento. Sin embargo, hay que tener en cuenta que todos los modelos de lenguaje que se basan en ontologías

y/o lenguajes de modelado pueden ocultar el significado de sus construcciones si los miembros del equipo los desconocen. A continuación, se presenta un ejemplo de un estudio de investigación que exploró esta diferenciación entre dibujo y modelo.

Se pidió a varios expertos de SysML que evaluaran el comportamiento de un sistema (específicamente un automóvil) dado un modelo de sistema y una condición inicial. [7]. El modelo de comportamiento se representó como un diagrama de máquina de estados SysML, tal y como se muestra en la Figura 5. Se pidió a los participantes que describieran el comportamiento del automóvil cuando estaba en el estado de "frenado" y se cumplieren simultáneamente las condiciones "liberación de freno=0" y "velocidad=0". Se les ofrecieron múltiples opciones para responder: el automóvil no experimentará esas dos condiciones simultáneamente cuando esté en ese estado, la situación está fuera del alcance del modelo, el sistema automáticamente pasa por defecto a una de las transiciones, o se trata de una transición no nominal que no es captada por el modelo. Las respuestas se distribuyeron uniformemente; es decir, no hubo acuerdo entre los expertos. El estudio demostró que, si bien los expertos podían leer el mismo diagrama, su comprensión de lo que transmitía el modelo era diferente, y esto se debía a una falta de comprensión de la construcción del modelo, algo que va más allá de la representación gráfica de algunos de sus aspectos.

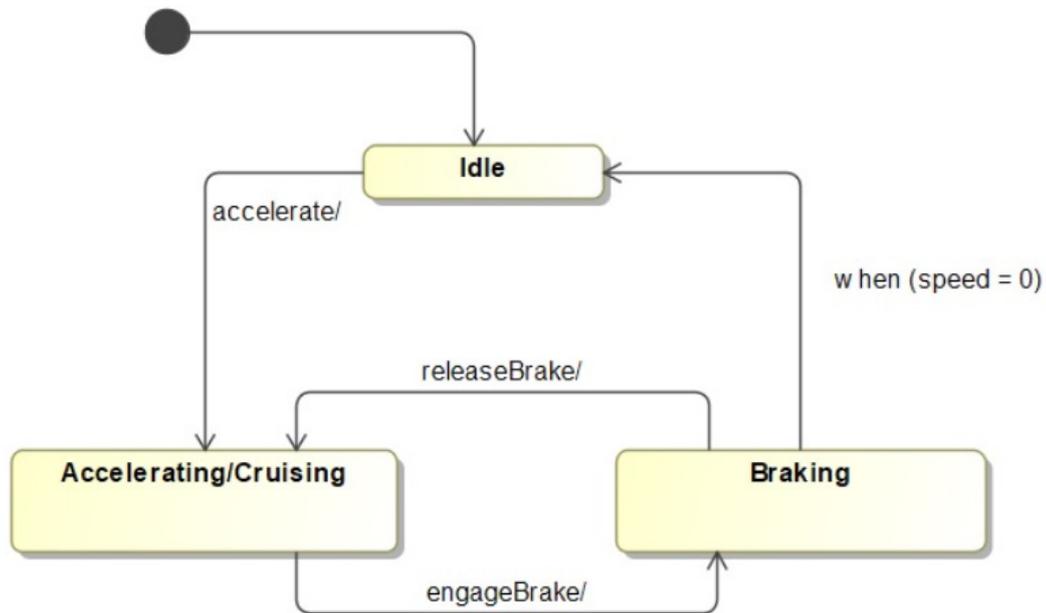


Figura 5. Modelo de comportamiento del sistema del automóvil utilizando un diagrama de máquina de estados SysML [reproducido de [7]]

Uno de los desafíos de las ontologías es que hay múltiples ontologías en juego en el nivel más alto de encapsulación del sistema. Cuando la atención se centra en cómo se realiza la ingeniería de sistemas y se caracteriza el sistema de interés resultante, existe una ontología central bastante pequeña (una ontología general de sistemas) que permite capturar, comunicar y razonar sobre conceptos como requisitos, funciones, intercambios, componentes, interfaces e interrelaciones entre esos conceptos (por ejemplo, satisfacer un requisito con una función o asignar una función a un componente). A medida que se incorpora la analítica y la fenomenología necesarias para desarrollar el sistema de interés, el alcance de la ontología requerida explota. La fenomenología de un avión, de una bomba de insulina o de un sistema informático es muy diferente, por lo tanto, el lenguaje que los rodea (la ontología específica del dominio) es diferente. El camino hacia el éxito en MBSE puede consistir en elegir una ontología de sistemas base que aborde el alcance y el lenguaje del equipo, y luego especializarla para incluir conceptos clave de ingeniería y conceptos que se alineen con los métodos de la organización.

Una comprensión básica de las ontologías justifica mejor por qué los modelos son más que simples dibujos. Volviendo a la analogía del CAD, la ontología subyacente de la geometría son los puntos y los vectores. Si se desplaza la cubierta de un avión hacia atrás 50 centímetros, se cambian los puntos y vectores subyacentes en el modelo de datos. Las vistas superior y frontal presentadas por CAD se ajustarían para mostrar la nueva posición de la marquesina. De manera similar, en MBSE, si se agrega un componente al diagrama de composición de un sistema, esto implica mucho más que agregar un cuadro en un dibujo. Se está especificando que ese sistema tiene un nuevo subcomponente. Cualquier representación de la arquitectura física del sistema reflejaría este nuevo subcomponente.

4. MBSE NO ES UNA SOLUCIÓN MILAGROSA: UNA BUENA INGENIERÍA DE SISTEMAS ES UN REQUISITO PREVIO PARA UN BUEN MODELADO

A pesar de su inclusión en el nombre, MBSE no debería tratarse de modelado. MBSE debería consistir en hacer ingeniería de sistemas aprovechando adecuadamente los modelos y un enfoque basado en modelos. Sin una comprensión adecuada de los principios, procesos y métodos fundamentales de la ingeniería de sistemas, es posible gastar una gran cantidad de tiempo y dinero en el modelado, pero no generará valor agregado. Desafortunadamente, este es un error común que cometen las organizaciones al adoptar MBSE: comprar varias licencias de herramientas y ofrecer a sus ingenieros cursos cortos de capacitación sobre la herramienta y/o el lenguaje de

modelado, sin garantizar una sólida experiencia subyacente en ingeniería de sistemas. Esto se ve exacerbado por un crecimiento de modeladores profesionales (la experiencia en modelado es relativamente fácil de desarrollar) a expensas de malas prácticas de ingeniería de sistemas (adquirir experiencia en ingeniería de sistemas es difícil).

La clave para adoptar MBSE es centrarse en la ingeniería de sistemas y reconocer que “basado en modelos” consiste en gran medida en utilizar técnicas asistidas por computadora para ejecutar mejor lo que se hacía anteriormente en un entorno basado en documentos. MBSE no es una transformación de la ingeniería de sistemas, sino más bien una evolución para seguir el ritmo y aprovechar el panorama tecnológico que cambia rápidamente. De hecho, la tecnología seguirá evolucionando, permitiendo a los ingenieros representar, comunicar y analizar los datos mañana mejor que hoy. Lo que hoy se denomina MBSE es el comienzo de un viaje evolutivo continuo para la ingeniería de sistemas.

También es importante reconocer que la tecnología no siempre es una bendición. Una trampa común de MBSE es que puede incitar a los ingenieros a ser prematuramente precisos, particularmente cuando se depende en gran medida de modeladores profesionales con poca experiencia en ingeniería de sistemas, lo que tiene como coste demasiado esfuerzo y una restricción excesiva del diseño en las primeras etapas del proceso de diseño. Las buenas prácticas de ingeniería de sistemas y el criterio de ingeniería deben tenerse en cuenta al avanzar en el ciclo de vida del sistema, pasando de niveles más altos de abstracción a una comprensión más detallada. El nivel de exactitud que se debe alcanzar debe estar determinado por el propósito del esfuerzo de modelado. Lo mismo se aplica a la precisión. Para saber cómo de exacto y preciso es un modelo, se necesitan unos principios sólidos de ingeniería de sistemas y un conocimiento del propósito detrás del esfuerzo de modelado.

5. NUEVAS CAPACIDADES HABILITADAS POR MBSE

Existen numerosas afirmaciones sobre los beneficios que proporciona MBSE en fuentes bibliográficas y en observaciones realizadas por profesionales. Algunos de estos beneficios incluyen una mejor comunicación, mayor coherencia, reducción de costes, reducción de tiempo, reducción de errores y mejor comprensión del sistema [8]. Una fortaleza clave de MBSE radica en la capacidad de aclarar la comunicación y el entendimiento compartido en todo el equipo. Es la capacidad de capturar mejor información a lo largo del tiempo y liberarla de la llamada deriva. Los recuerdos y la comprensión cambian, por lo que capturar formalmente el sistema proporciona una representación más precisa a lo largo del tiempo. Una representación clarificada resalta dónde se tienen lagunas en

el pensamiento o comprensión inconsistente en todo el equipo. Una representación arquitectónica descriptiva de mayor fidelidad ayuda a alinear al equipo: aclara la comprensión, expone suposiciones y refleja el recorrido de diseño detrás de la solución actual. En algún momento de la implementación habrá dudas sobre una decisión de diseño, una nueva tecnología o un cambio de requisitos, por lo que capturar el recorrido del diseño es tan crítico como el diseño resultante.

Muchos de los beneficios citados anteriormente se derivan del establecimiento de MBSE de una fuente autorizada de la verdad (ASOT, *Authoritative Source of Truth*) para los sistemas [9]. Básicamente, esto es formalizar ese “tejido conectivo” discutido anteriormente. En los inicios de la ingeniería de sistemas basada en modelos, existía la visión de que MBSE se estableciera como la única fuente de verdad; un elemento central que garantizaba que todos los datos utilizados en un proyecto fueran consistentes. En esencia, los datos vivían en un lugar y otros modelos apuntarían a ese lugar. En este sentido, MBSE ayudó a propagar los cambios de datos en todos los modelos que se basaban en dichos datos. Sin embargo, una única fuente de verdad es sólo una solución al problema de la coherencia y la relevancia, y conlleva algunos inconvenientes, especialmente en términos de vulnerabilidad y eficiencia. Últimamente, el concepto se ha transformado en el ASOT más general, donde la clave no está en la singularidad de la ubicación sino en la certificación de los datos que se utilizan y/o se obtienen. MBSE en este sentido permite la identificación y etiquetado de datos utilizados en todo el entorno de modelado.

En esta idea, ASOT es el habilitador de la ingeniería digital, el subproceso digital, el gemelo digital y todos los artefactos de transformación digital relacionados. Si el equipo de ingeniería de sistemas es el tejido conectivo técnico que une al equipo del proyecto, MBSE es el tejido conectivo digital que permite la ingeniería digital, la creación del hilo digital y el desarrollo del gemelo digital. En particular, se podría concebir MBSE como el subconjunto de la ingeniería digital que permite que todos los dominios dispares involucrados en el proceso de ingeniería trabajen colectivamente con esa fuente autorizada de datos [8].

Más allá de estos beneficios, la correcta ejecución de MBSE permite obtener capacidades novedosas que aceleran y avanzan en el ciclo de vida de los sistemas en general. En el centro de cada uno de ellos está la representación de datos y conocimientos en una estructura y manera que sea computable, tanto por humanos como por máquinas. Al liberar el conocimiento de documentos y artefactos, MBSE permite una mejor alineación en toda la empresa, proporcionando los datos correctos en el lugar correcto, en el momento correcto, en el nivel correcto de abstracción en un contexto completo y presentado adecuadamente para que el consumidor los comprenda, analice y decida mejor. Esto contrasta con los métodos tradicionales que representan un conjunto de datos (probablemente tanto información faltante como datos superfluos para la decisión en cuestión) en una sola presentación y sin contexto. Esta consulta dinámica y presentación de información bajo demanda garantiza la coherencia con el modelo subyacente [10-12]. Si se combina adecuadamente con visualizaciones sólidas que incluyen documentos, diagramas, tablas y opciones modernas, como animaciones dinámicas y motores de juegos, MBSE desbloquea el poder de múltiples perspectivas, desde la del ingeniero hasta la del operador y el experto en la materia durante todo el ciclo de vida (ref. Figura 6).

#	△ Name	Text	Verified By
1	1 Power	The system shall operate with a maximum power of 50 kW	Test 2
2	2 Weight	The system shall weigh less than 1800 kg	Test 3
3	3 Latency	The system shall operate with a maximum latency of 200 ms	Test 1
4	4 Function	The system must report errors	

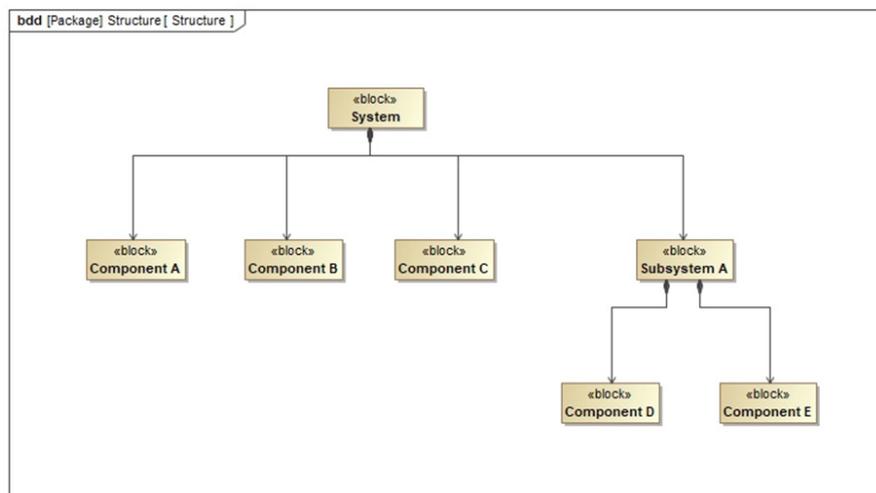


Figura 6. Ejemplo de consulta de un modelo

Bien hecho, MBSE refleja no sólo los bits de datos individuales sino también las relaciones y dependencias entre ellos, lo que ayuda a pasar de los datos a la información y a la representación del conocimiento. Este mapa de conocimiento permite el análisis de trazabilidad clásico, como verificar que todos los requisitos se cumplan en la solución y que todas las funciones estén asignadas a la arquitectura física (ref. Figure 6). Además, facilita la evaluación rápida del cambio,

lo que permite al equipo de ingeniería rastrear el impacto de un cambio propuesto (ya sea un nuevo requisito o un nuevo componente), identificar los aspectos afectados de la arquitectura de la solución y evaluar el cambio en el contexto de decisiones de diseño previas (ref. Figure 8). Si bien los humanos siguen siendo responsables de la ingeniería y el análisis, el alcance y el contexto proporcionados por el mapa de conocimiento mejoran la calidad y aceleran el análisis.



Figura 7. Ejemplos de visualización de análisis de trazabilidad

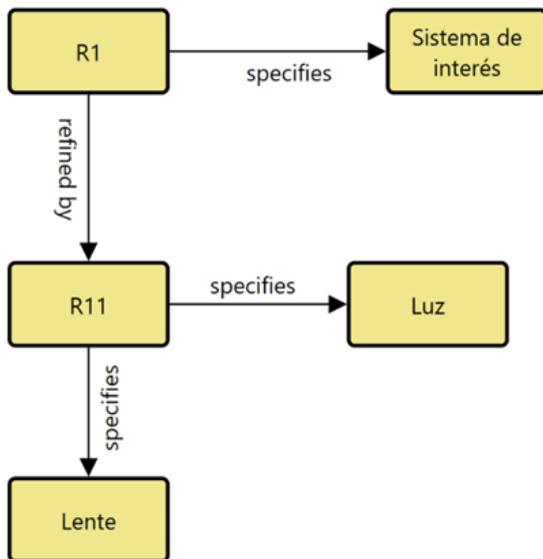


Figura 8. Ejemplo de visualización de análisis de propagación de cambios. (El diagrama se genera automáticamente al marcar que el requisito R1 ha cambiado. Entonces, identifica todos los elementos que están asociados a dicho requisito.)

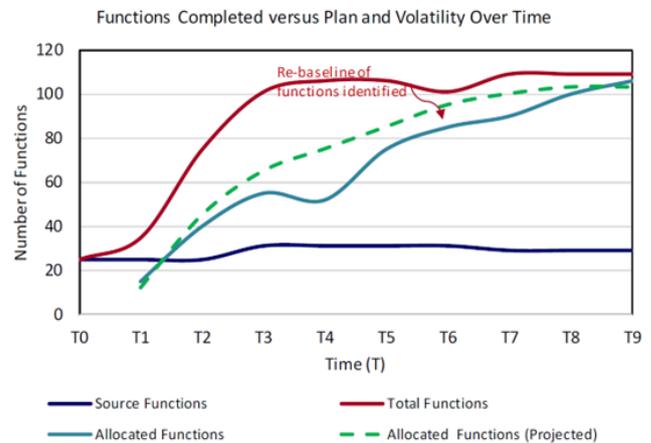


Figura 9. Ejemplo de visualización de métricas de integridad

El mapa de conocimiento generado a través de MBSE se puede analizar mediante técnicas informáticas para identificar problemas de integridad y de consistencia del diseño. Por ejemplo, las comprobaciones de integridad tradicionales, como garantizar que los requisitos se rastrean hasta los elementos de la solución, se verifican los requisitos y se asignan las funciones, se pueden realizar de manera consistente y generar rápidamente métricas para reflejar la madurez del diseño (ref. Figura 9). Verificaciones más sofisticadas pueden identificar problemas de consistencia del diseño, como tener en cuenta todas las entradas, salidas e interfaces durante la descomposición o garantizar que las interfaces sean del tipo correcto en los intercambios de transferencia entre componentes (ref. Figura 10). Más allá de las técnicas informáticas tradicionales, se podrían aplicar algoritmos de aprendizaje automático para identificar patrones y sugerir alternativas de diseño.

Lo mejor de todo es que el modelo del sistema representa un prototipo de sistema virtual desde el primer día, aunque con un alto nivel de abstracción al principio del proyecto. El modelo se puede simular dinámicamente para identificar problemas y confirmar el rendimiento del sistema. A medida que se perfecciona el modelo del sistema y se combinan los modelos analíticos con el modelo arquitectónico descriptivo, aumenta el nivel de detalle y precisión. Esto permite la evaluación y verificación continua del diseño, acelerando la detección de defectos y permitiendo al equipo de ingeniería realizar rápidamente estudios comerciales.

Estos ejemplos no son exhaustivos, pero deberían dar una idea del estado de lo posible cuando la ingeniería de sistemas se traslada a un entorno de modelado.

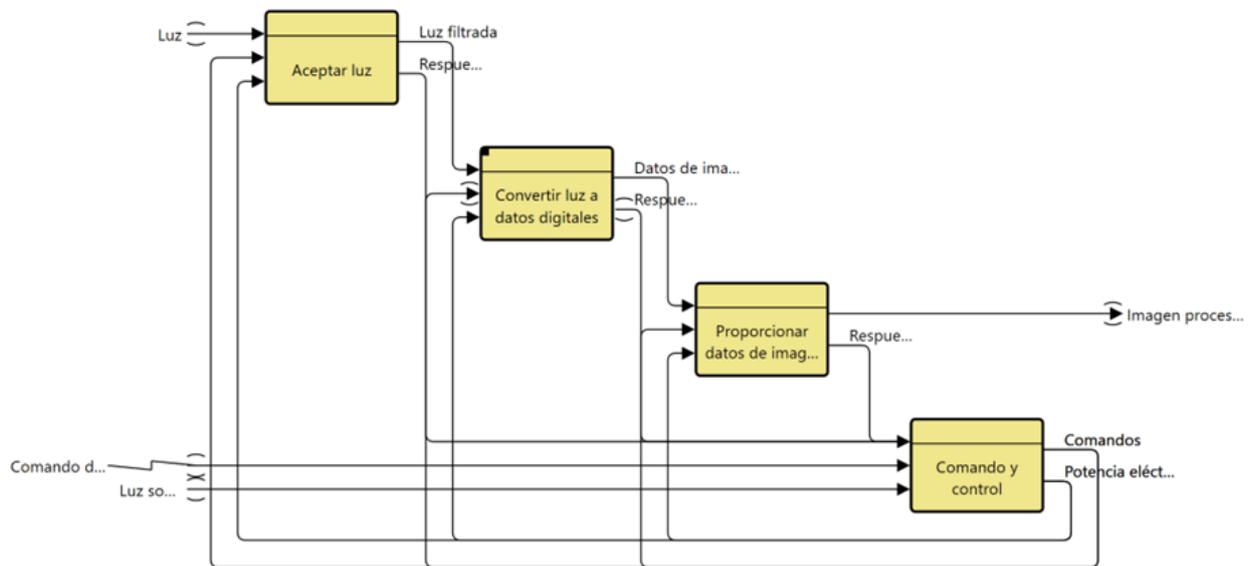


Figura 10. Ejemplo de visualización de la verificación de integridad del diseño automatizado (Nota: Los paréntesis al lado de una señal en los lados del diagrama indican que tal señal falta en el nivel superior de encapsulación; los corchetes en una señal junto a una caja indican que tal señal no está asignada en el nivel inferior de encapsulación.)

Ingeniería digital: un enfoque digital integrado que utiliza fuentes autorizadas de datos y modelos de sistemas como un conjunto de disciplinas que respaldan las actividades del ciclo de vida desde el concepto hasta la eliminación.

Hilo digital: el marco de comunicación que permite el flujo de datos conectados y una visión integrada de los datos de un activo a lo largo de su ciclo de vida a través de una perspectiva funcional tradicionalmente aislada.

Gemelo digital: un modelo computacional de un sistema físico particular con comunicación bidireccional con su contraparte física. Un gemelo digital evoluciona simultáneamente con el sistema físico y refleja el estado, la posición y la historia del sistema.

Diseño organizacional	Habilitadores/ barreras organizacionales	Gestión del cambio organizacional
Conocimientos/ habilidades	Liderazgo/apoyo a la gestión/ compromiso	Aplicación de métodos/procesos MBSE y prácticas de modelado
Integración	Entrenamiento	Estrategia y diseño de adopción/ implementación
Beneficios/ resultados demostrados	Recursos para la implementación	Gestión del cambio cultural
Estructura organizacional	Colección de herramientas	Disposición a utilizar herramientas (aceptación de los empleados y partes interesadas)

Tabla 1. Aspectos necesarios para una implementación y adopción exitosa de MBSE [adaptado de [10]]

Interconexión: las personas dentro de la organización interactúan con frecuencia y están dispuestas a ayudar a otros con problemas.

Estandarización: el uso de herramientas y métodos MBSE está estandarizado en toda la unidad organizacional.

Flexibilidad: las organizaciones pueden adaptarse fácilmente a los cambios, las nuevas tecnologías y los procesos.

Estos factores describen algunas formas en que una organización puede prepararse para el éxito al adoptar MBSE. Es importante tener algún tipo de red establecida para las personas que están aprendiendo MBSE. Esto puede tomar muchas formas: mentores, entrenadores, expertos definidos a los que alguien puede recurrir o una red de compañeros experimentados. También es beneficioso tener estandarizadas las herramientas y métodos que se utilizan en toda la unidad organizativa. Esto puede facilitar que las redes informales/formales de personas ayuden a todos los demás que están aprendiendo. Por ejemplo, si se utiliza la misma herramienta en todo el equipo, es más probable que las personas puedan ayudar si surge un problema relacionado con la herramienta. La conexión con la flexibilidad resalta la importancia de la gestión del cambio. Si bien es más difícil implementar este factor en un período corto, se pueden tomar medidas para garantizar que las personas en la organización estén preparadas para adoptar MBSE. Asegurarse de que el propósito de MBSE sea claro y explicar cómo puede ser beneficioso para el trabajo diario de una persona contribuye en gran medida a tener una plantilla lista para aceptar MBSE.

Si bien este tipo de investigación sobre la adopción de MBSE es útil, es importante saber que no existe un enfoque único para adoptar MBSE con éxito. Como se analizó con los estándares y metodologías, el propósito de una organización detrás del uso de MBSE puede variar ampliamente. Una estrategia de adopción debe estar en línea con ese propósito, junto con varios otros factores organizacionales (por ejemplo, aceptación del liderazgo, apoyo financiero, experiencia de los empleados en la materia, etc.).

Dentro de una organización, hay tres grupos principales a considerar al definir una estrategia de adopción [17]:

- 1. Los iniciadores e impulsores de MBSE.** Se debe contar con cierto nivel de conocimiento y experiencia para impulsar realmente el uso efectivo de MBSE en los proyectos de una organización. Hay muchos errores en el modelado en los que es fácil caer, lo que podría dar lugar a la creación de modelos que en realidad no añaden valor, dificultando aún más la adopción de MBSE. Es importante (especialmente al principio) mostrar los beneficios del modelado, por lo que es clave contar con las personas adecuadas en el equipo de trabajo, de forma que apoyen el esfuerzo a acometer.
- 2. Las unidades organizativas que deben funcionar basadas en modelos.** Construir modelos es perfecto, pero si no se utilizan para nada, ¿qué sentido tiene? Para que MBSE cumpla con el propósito previsto, debe encajar en el proceso de una organización. En algunos casos, esto podría requerir ajustar los flujos de trabajo, las descripciones/ responsabilidades del trabajo o incluso la estructura de la organización. Es más que simplemente traducir enfoques centrados en documentos a un mundo basado en modelos; es la transformación digital. Una unidad organizacional que adopte MBSE debe planificar concienzudamente en qué parte de su proceso tiene sentido utilizar MBSE para crear el mayor valor.

3. Las unidades de organización responsables del tiempo y presupuesto de los proyectos de ingeniería. MBSE representa una importante inversión en dinero y tiempo para una organización. Este hecho debe considerarse y utilizarse para gestionar las expectativas de los líderes y de otros. La unidad organizacional que está adoptando MBSE necesita tener el espacio y los recursos para hacerlo con éxito.

Preparar a los trabajadores, configurar la infraestructura para su establecimiento y definir una estrategia de implantación son factores fundamentales para la adopción de MBSE. Pero una vez que una organización llega a ese punto, debe realizar la adopción real. Un factor clave aquí es la formación. Muchas personas encuentran que aprender MBSE es un desafío. Esto podría deberse a que en realidad están aprendiendo varias cosas a la vez: una herramienta, un lenguaje de modelado, un método e ingeniería de sistemas. Según los profesionales, todas las partes interesadas necesitan algún nivel de formación, pero la cantidad y lo que necesitan aprender varía [13].

La Tabla 2 muestra cuatro categorías de roles y sus necesidades comunes de capacitación en MBSE. Es posible que diferentes roles requieran solo un subconjunto de los componentes de MBSE (es decir, herramienta, lenguaje, método, conceptos de SE) con los que deben estar familiarizados. Por ejemplo, alguien que actúa como “revisor de modelos” probablemente no necesite capacitación relacionada con la herramienta o método específico. Pero sí necesitan poder comprender e interpretar los artefactos MBSE. En este caso, algo de formación en el lenguaje de modelado puede ser todo lo que se necesita. Vale la pena destacar dos aspectos. En primer lugar, los roles no son necesariamente excluyentes entre sí, pero un individuo determinado puede desempeñar más de un rol en diferentes capacidades a la vez. En segundo lugar, mientras que algunos equipos pueden decidir dividir el papel de un modelador y un ingeniero de sistemas, otros equipos pueden asignar a cada ingeniero de sistemas (o ingeniero en general) tareas de modelado sin depender de un equipo dedicado para las tareas de modelado.

Categorías de roles que requieren formación	
Revisores de modelos	Líderes, partes interesadas o clientes que necesitan saber cómo utilizar los modelos para tomar decisiones
Desarrolladores (Modeladores)	Personas que construyen y mantienen los modelos en las herramientas, por lo que necesitarán conocimientos detallados de cómo funciona la herramienta
Otros ingenieros	Personas que trabajarán en el modelo hasta cierto punto. Estas personas suelen ser ingenieros superiores o personas de otras disciplinas que ayudan con el contenido del modelo
Administradores	Personas que trabajan en TIC o software. Administran las cuentas, licencias, herramientas, etc.

Tabla 2. Categorías de roles que requieren formación en MBSE

La integración del trabajo es otro factor crítico para la adopción exitosa de MBSE en toda la empresa [15]. La ingeniería de sistemas en general implica el trabajo conjunto de diferentes equipos/disciplinas funcionales, y este sigue siendo el caso de MBSE. Los miembros del sistema en proceso de modelado están a menudo fuera del equipo de modelado, por lo que la coordinación entre estos dos grupos es clave. Un equipo puede crear un gran modelo, pero si no representa con precisión el sistema, entonces es de poca utilidad. Además, los modelos de sistemas se dividen frecuentemente en una colección de modelos de sistemas más pequeños. En otras palabras, varios equipos son responsables de modelar un componente del sistema que finalmente se integra. En estos casos, es fundamental tener cierto nivel de estandarización/consistencia entre esos grupos, ya que esos modelos dispares tendrán que integrarse en última instancia. Hay muchas maneras diferentes de modelar correctamente la misma cosa, por lo que tener algunas pautas que los grupos deben seguir en relación con algunas de esas decisiones ayudará a que el producto final sea más cohesivo y reduzca el esfuerzo de integración.

La adopción de MBSE es difícil porque hay muchas partes implicadas. Pero, en última instancia, hay diversas maneras en que MBSE puede aportar valor al equipo. Una organización sólo necesita encontrar los beneficios de utilizar ingeniería de sistemas basada en modelos y poder así tomar las diferentes decisiones.

7. CONCLUSIONES.

Dada la velocidad a la que el mundo sigue acelerándose, la transición a enfoques de ingeniería de sistemas basados en modelos digitales es inevitable. Las prácticas estándar simplemente no pueden satisfacer la demanda de un rápido desarrollo y producción de ideas para convertirlas en realidad. La ingeniería de sistemas en sí no está cambiando, sólo la forma en que se realiza. La clave para adoptar MBSE es centrarse en una buena ingeniería de sistemas y reconocer que la tecnología basada en modelos consiste en gran medida en utilizar técnicas asistidas por computadora para ejecutar mejor lo que siempre se ha hecho.

MBSE está ganando terreno en la industria y la administración y poco a poco un número mayor de personas comienzan a ver los beneficios de usarlo. En esencia, MBSE trata de capturar datos de una mejor manera utilizando técnicas modernas que permitan a los equipos comprender, comunicar, razonar y retener mejor el conocimiento. El coste de la adopción en términos de tiempo y dinero es alto, pero el precio de no adoptarlo será aún mayor a largo plazo.

REFERENCIAS

1. Blanchard, B.S. and J.E. Blyler, Systems Engineering Management. 5th ed. 2016, Hoboken, NJ, USA: John Wiley & Sons, Inc.
2. INCOSE, Systems Engineering Vision 2025. 2014.
3. INCOSE, Systems Engineering Vision 2035. 2023.
4. Defense, D.o., DoD Modeling and Simulation (M&S) Glossary. 1998.
5. Henderson, K. and A. Salado, Is CAD A Good Paradigm for MBSE? INCOSE International Symposium, 2021. 31(1): p. 144-157.
6. Long, D. and Z. Scott, A Primer For Model-Based Systems Engineering. 2nd ed. 2011, USA: Vitech Corporation.
7. Cratsley, B., et al., Interpretation Dscrepancies of SysML State Machine: An Initial Investigation, in Conference on Systems Engineering Research. 2020: Virtual.
8. Henderson, K. and A. Salado, Value and benefits of model-based systems engineering (MBSE): Evidence from the literature. Systems Engineering, 2021. 24(1): p. 51-66.
9. Blackburn, M. and T. West, Fundamentals of Digital Engineering, in Systems Engineering for the Digital Age. Practitioner Perspectives, D. Verma, Editor. 2023, John Wiley and Sons, Inc.: Hoboken, NJ, USA. p. 3-24.
10. Salado, A., 5.5.2 Efficient and Effective Systems Integration and Verification Planning Using a Model-Centric Environment. INCOSE International Symposium, 2013. 23(1): p. 1159-1173.
11. Dunbar, D., et al., Transforming Systems Engineering Through Integrating Modeling and Simulation and the Digital Thread, in Systems Engineering for the Digital Age. Practitioner Perspectives, D. Verma, Editor. 2023, John Wiley and Sons, Inc.: Hoboken, NJ, USA. p. 47-68.
12. Dunbar, D., et al., Driving digital engineering integration and interoperability through semantic integration of models with ontologies. Systems Engineering, 2023. 26(4): p. 365-378.
13. Henderson, K., T. McDermott, and A. Salado, MBSE adoption experiences in organizations: Lessons learned. Systems Engineering, 2024. 27(1): p. 214-239.
14. Henderson, K., et al., Towards Developing Metrics to Evaluate Digital Engineering. Systems Engineering, 2023. 26: p. 3-31.
15. Henderson, K., Exploring the Adoption Process of MBSE: A Closer Look at Contributing Organizational Structure Factors, in Grado Department of Industrial and Systems Engineering. 2022, Virginia Tech: Blacksburg, VA, USA.
16. Henderson, K. and A. Salado, The Effects of Organizational Structure on MBSE Adoption in Industry: Insights from Practitioners. Engineering Management Journal, 2024. 36(1): p. 117-143.
17. Weilkens, T., Adoption of MBSE in an Organization, in Handbook of Model-Based Systems Engineering, A.M. Madni, N. Augustine, and M. Sievers, Editors. 2020, Springer International Publishing: Cham. p. 1-19.

DAVID LONG

David Long es el presidente de Blue Holon y científico investigador del Centro de Investigación de Ingeniería de Sistemas (SERC, *Systems Engineering Research Center*). Durante más de 30 años, ha ayudado a organizaciones de todo el mundo a aumentar su competencia en ingeniería de sistemas y, al mismo tiempo, ha trabajado para avanzar en los últimos avances de esta disciplina. Trabaja con organizaciones gubernamentales y comerciales mientras evalúan, adoptan e implementan nuevos métodos y herramientas para mejorar su empresa de ingeniería. A lo largo de su carrera, David ha desempeñado un papel técnico y de liderazgo clave en el avance y la expansión de la práctica de la ingeniería de sistemas. David fundó y dirigió Vitech, donde desarrolló métodos y software innovadores y líderes en la industria (CORE™ y GENESYS™) para diseñar sistemas de próxima generación. David participa frecuentemente en eventos de la industria en todo el mundo, ofreciendo conferencias magistrales y talleres que abarcan los fundamentos de la ingeniería de sistemas, la ingeniería de sistemas práctica basada en modelos, la ingeniería digital y el futuro de los sistemas de ingeniería. Sus experiencias y esfuerzos lo llevaron a ser coautor del libro *A Primer for Model-Based Systems Engineering* para difundir los conceptos fundamentales de este enfoque clave para los desafíos modernos. David, miembro de INCOSE y profesional experto en ingeniería de sistemas (ESEP), fue presidente de INCOSE en 2014/2015. Actualmente, David desempeña el rol de Director de Integración Estratégica de INCOSE y profesor del Instituto de Liderazgo Técnico de INCOSE. David tiene una licenciatura en Ciencias de la Ingeniería y Mecánica y una maestría en Ingeniería de Sistemas de Virginia Tech.



DRA. KAITLIN HENDERSON

La Dra. Kaitlin Henderson es arquitecta de sistemas en Radiance Technologies, donde apoya diversos esfuerzos de modelado SysML. Obtuvo un doctorado en Ingeniería Industrial y de Sistemas con especialización en Sistemas de Gestión de Virginia Tech en Blacksburg, Virginia. También obtuvo sus títulos de licenciatura y maestría en Ingeniería Industrial y de Sistemas en la misma universidad. Kaitlin ha contribuido a los campos de la Ingeniería de Sistemas Basada en Modelos (MBSE, por sus siglas en inglés) y la Ingeniería Digital (DE, por sus siglas en inglés) con sus trabajos sobre la adopción de MBSE, la madurez del dominio de MBSE y la medición del desempeño de MBSE/DE. Kaitlin ganó el premio a la Mejor Presentación en el Foro de Estudiantes Doctorales del SERC en 2020 y ha sido reconocida por sus publicaciones de 2021 y 2022 en Ingeniería de Sistemas.



BELINDA MISIEGO TEJEDA

Belinda Misiego Tejeda es la jefa de área de innovación de Isdefe y coordina el Campus de Ingeniería de Sistemas de la compañía. A través de este Campus se realizan cursos de formación interna con personal propio en los distintos ámbitos de esta disciplina: iniciación en ingeniería de sistemas, fundamentos de ingeniería de sistemas y diversos cursos específicos. Cuenta con más de 20 años de experiencia profesional en los sectores de seguridad y defensa, donde ha desarrollado y gestionado múltiples proyectos de ingeniería de sistemas, principalmente en el ámbito de los sensores y la guerra electrónica. Belinda, miembro de INCOSE y profesional en ingeniería de sistemas (CSEP), formó parte de la junta de la Asociación Española en Ingeniería de Sistemas (AEIS), el capítulo español de INCOSE entre 2017 y 2022. Además, desarrolló el rol de Secretaria del Director de EMEA de INCOSE en 2019-2022. Belinda es Ingeniera de Telecomunicaciones y Licenciada en Investigación y Técnicas de Mercado por la Universidad de Valladolid y Máster en Gestión de Contratos y Programas en el Sector Público por la Universidad de Educación a Distancia (UNED).





“¿Está realmente el universo tan poco cohesionado? ¿O esta pequeña dimensionalidad se debe al hecho de que los humanos que desarrollaron las ecuaciones controlaron sus experimentos de acuerdo con sus limitaciones cognitivas?”

G. Friedman

Transformación Digital en el Desarrollo de Sistemas

Christopher L. Delp, *Jet Propulsion Laboratory, California Institute of Technology* (christopher.l.delp@jpl.nasa.gov)

Dr. Joe Gregory, *The University of Arizona* (joegregory@arizona.edu)

Luis Miguel Aparicio Ortega, *ISDEFE* (laparicio@isdefe.es)

Resumen

Este capítulo presenta las nuevas capacidades disponibles para el desarrollo, integración y validación de sistemas como consecuencia del uso de modelos digitales, la alta potencia de cálculo de los ordenadores actuales, así como de la computación en la nube y en malla. Introduce el concepto de la Transformación Digital y explica cómo representa un marco superior a la simple digitalización. También se hace énfasis en las tecnologías clave que facilitan la Transformación Digital, incluyendo los lenguajes formales y las relativas a la Web Semántica. El capítulo también examina algunos ejemplos de cómo la Transformación Digital puede ayudar a mejorar los resultados en los procesos de ingeniería de sistemas. Por ejemplo, permitiendo una mayor trazabilidad, la generación y evaluación automática de arquitecturas, el diseño basado en conjuntos y la integración de modelos basados en la física con los modelos de ingeniería de sistemas.

Palabras clave

Ingeniería Digital, Hilo Digital, Exploración del Espacio de Soluciones, Diseño Basado en Conjuntos, Diseño Óptimo Multidisciplinario, Web Semántica, Ontologías, Modelos de Sistemas



1. INTRODUCCIÓN

El interés por la Transformación Digital ha experimentado un rápido crecimiento en la última década. En este capítulo, presentamos una visión general de esta disciplina en continua y rápida evolución y discutimos algunas de las tecnologías emergentes más relevantes que son empleadas, proporcionando ejemplos de cómo se están aplicando a la Ingeniería de Sistemas (IS). En primer lugar, para comprender a qué nos referimos cuando hablamos de Transformación Digital, es fundamental hacer una distinción crucial entre este término y otro habitualmente empleado en este contexto, la digitalización.

La digitalización se refiere simplemente al cambio que ocurre cuando un objeto analógico, sea cual sea, se transforma en un objeto digital. Por ejemplo, si tomamos un documento físico, como un informe o una fotografía, y lo convertimos en un formato digital mediante escaneo o simplemente escribiendo el texto en un procesador de texto, hemos realizado la digitalización de ese documento. Este tipo de proceso ha estado ocurriendo durante décadas, y probablemente todos comprendemos los beneficios de trabajar con objetos digitales en lugar de documentos en papel. Por ejemplo, la digitalización permitió que grandes cantidades de información se transmitieran en formato digital y sin errores. Este proceso fue posible gracias a los avances en la miniaturización de computadoras y a la infraestructura de redes y revolucionó la forma en la que la información puede ser gestionada y comunicada. Es lo que conocemos como la Era de la Información o la Tercera Revolución Industrial.

La Transformación Digital, sin embargo, va más allá de la sola digitalización. En ella, los ingenieros trabajan con modelos y datos que están vinculados digitalmente. Cualquier consumo de información es simplemente una visualización de los modelos y datos subyacentes. En

otras palabras, la información y su visualización están desacopladas. Debido a esto, la Transformación Digital a menudo implica replantearse los procesos tradicionales con el fin de aprovechar al máximo las oportunidades que aparecen al conectar datos y modelos de una manera legible por una máquina. Esto implica claramente una reflexión profunda sobre cómo las tecnologías modernas pueden ser aprovechadas de manera efectiva, especialmente cuando se aplican en un contexto empresarial. Cuando se aplica dentro de los límites de una organización, la Transformación Digital requiere una cuidadosa consideración de los cambios culturales y organizativos que van a ser necesarios aplicar, la adquisición y desarrollo de talento que sea relevante, la gobernanza de datos y la seguridad dentro y fuera de los límites de la organización, así como otros múltiples aspectos. En resumen, este proceso de digitalización y mejora de los procesos en un contexto empresarial es lo que se conoce como la Transformación Digital [1].

Antes de examinar algunas de las tecnologías que habilitan la Transformación Digital, consideremos cómo debería ser un proceso de IS sometido a dicha transformación. En otras palabras, para entender qué es lo que se requiere para soportar nuestro proyecto de Transformación Digital, debemos tener una visión clara de lo que nuestra empresa, digitalmente transformada, debería ser capaz de hacer. En resumen, la información generada en todas las etapas del ciclo de vida del sistema está en forma digital, está vinculada a través de una arquitectura impulsada por datos de recursos compartidos, alimenta modelos descriptivos y cuantitativos y puede ser utilizada para la toma de decisiones en tiempo real y a largo plazo [2] (ver Figura 1). En términos prácticos, esto significa que los datos pueden intercambiarse sin problemas entre las diferentes herramientas que apoyan el modelado en cada una de las fases del ciclo de vida del sistema. Esto no solo proporciona la trazabilidad necesaria entre decisiones de diseño, pruebas, requisitos, etc.; también permite realizar análisis de soporte a la toma de decisiones que anteriormente eran inviables.

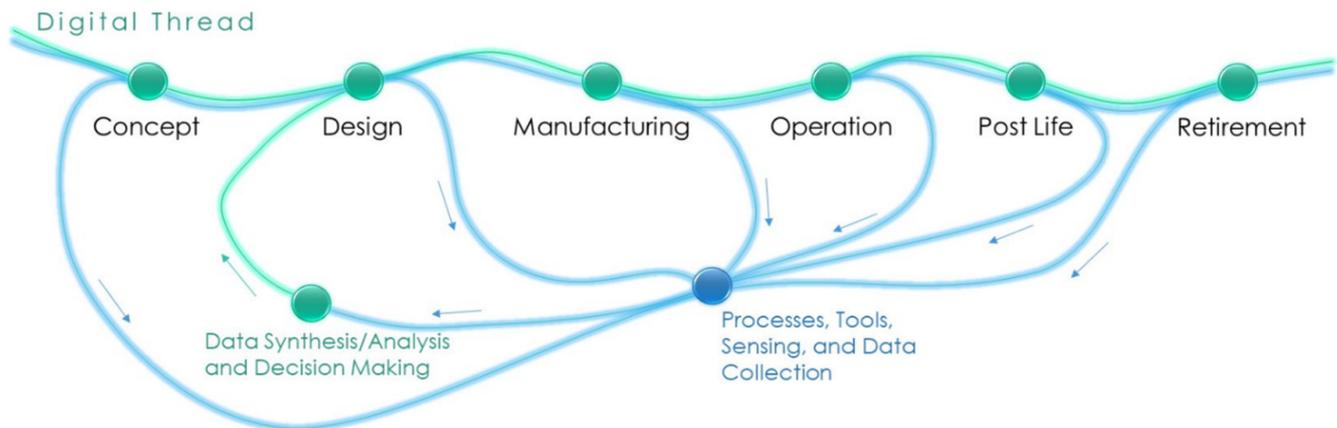


Figura 1. El Hilo Digital relacionado con la Ingeniería de Sistemas, adaptada de Singh y Willcoxy, 2018

Esta capacidad se extiende incluso más allá de los procesos internos de cualquier organización. Un hilo digital ideal permite la colaboración y el intercambio de datos sin interrupciones a través de los límites de las organizaciones para apoyar aspectos críticos tales como la integración de la cadena de suministro. Dentro de los límites de los correspondientes protocolos de seguridad y privacidad, los datos pueden ser rápidamente accedidos a lo largo del hilo digital para apoyar la toma de decisiones a nivel empresarial. Las tuberías adecuadas de propagación de cambios aseguran que los efectos posteriores de un cambio en los datos sean identificados y reconocidos.

2. HABILITADORES DE LA TRANSFORMACIÓN DIGITAL

2.1. Los lenguajes de programación explotan el poder de la matemática aplicada

El desarrollo de sistemas cada vez más sofisticados, junto con la importante criticidad de su calidad y coste, impulsan la necesidad de emplear prácticas de IS que sean cuantificables y analizables. Satisfacer esta necesidad requiere a menudo el desarrollo de una representación ejecutable del sistema. La conjunción entre esta necesidad y la disponibilidad de entornos de computación modernos nos lleva a un resultado bastante poético: ahora es posible utilizar un sistema desarrollado por humanos para entrar en

un ciclo de desarrollo recursivo en el que las mejoras en la disciplina de la IS pueden producir mejoras en la forma en que practicamos la disciplina.

El punto clave de este concepto se basa en la capacidad de modelar matemáticamente el mundo que nos rodea. En la Ciencia de la Computación, la esencia de esta innovación es la combinación de las instrucciones del procesador con lo que posiblemente sea el invento más significativo en la historia de la humanidad: el lenguaje.

Desde sus inicios, la IS se ha centrado en la especificación del comportamiento sistemático de un elemento conocido como "Sistema". Este elemento puede adoptar muchas formas, siempre y cuando el concepto clave que defina su existencia sea su comportamiento sistemático. La IS tradicional se enfocaba en descripciones informales, experimentos mentales humanos y heurísticas para abordar los problemas cada vez más complejos que se esperaba que ella resolviera [3]. Este enfoque a menudo dependía de la realización de pruebas costosas en el mundo real y de la existencia de restricciones rígidas e inflexibles sobre los cambios necesarios para enfrentar desafíos imprevistos.

El desarrollo, implantación y adopción de forma generalizada de lenguajes de modelado declarativo adecuados para apoyar la IS ha sido denominado como 'El Desafío de la Ingeniería de Sistemas' [3]. En las últimas décadas, el aumento de la potencia computacional y el desarrollo de una multitud de lenguajes de modelado han contribuido al avance de la IS. Algunos de estos se muestran en la Figura 2.

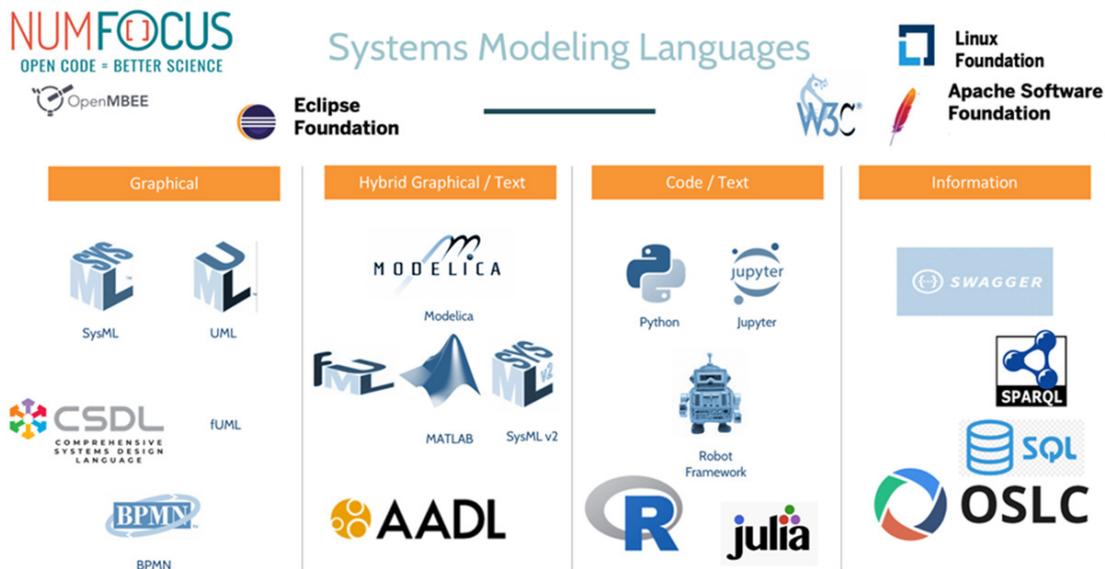


Figura 2. Ejemplos de Lenguajes de Modelado Declarativo para apoyar la ingeniería de sistemas y algunos organismos y fundaciones que soportan los estándares y el software de código abierto en los que se basan

Lenguajes gráficos como SysML versión 1 y CSDL proporcionan la capacidad de modelar visualmente las especificaciones de conceptos y relaciones. Lenguajes basados en texto/código como Julia y Robot amplían el poder a lo largo del ciclo de vida para capturar capacidades detalladas de ejecución, análisis y prueba de sistemas. La nueva generación de lenguajes híbridos como AADL, SysML versión 2 o Modelica apunta a una enorme capacidad para especificar, visualizar, codificar y analizar sistemas a lo largo del ciclo de vida. Los lenguajes de modelado de información como OWL/SparQL y OpenAPI/Swagger también desempeñan un papel crucial en el modelado de sistemas modernos y en la Ingeniería Digital (ID). Todos ellos proporcionan una base para conseguir la interoperabilidad basada en la web requerida para la integración e intercambio de modelos.

Los lenguajes presentados en la Figura 2 pueden ser considerados como ‘Lenguajes Declarativos’. Éstos se centran en semánticas que pueden resolverse de diversas formas, además de la ejecución tradicional. A menudo, pueden ser resueltos utilizando técnicas analíticas en contraposición a las semánticas procedimentales que ejecutan declaraciones en el orden en que se aparecen. Los modelos declarativos tienden a tener fórmulas matemáticas que pueden resolverse de más de una manera. Cabe destacar también que, aunque lenguajes como MATLAB, Python o Julia no son estrictamente declarativos, construyen bibliotecas y capacidades que se usan con frecuencia en ingeniería para realizar análisis sobre modelos declarativos, como solucionadores de álgebra y otros análisis basados en ecuaciones. Ejemplos de esto son la Lógica de Primer Orden (LPO) y la Máquina de Estados Finitos (MEF). La LPO en particular ha recibido una atención significativa, y no solo en ingeniería, debido a su capacidad para capturar conocimiento de manera precisa e inferir nuevo basado en un conjunto de reglas [5]. Las MEFs son un buen ejemplo de este tipo de lenguaje. Declaran estados, transiciones y eventos. Una MEF puede interpretarse de manera diferente mientras se mantiene la formalidad matemática. La manera más directa de resolver una MEF es definir una trayectoria de eventos y probar sobre la máquina para ver el resultado. Un ejemplo menos obvio sería usar un solucionador para analizar toda la MEF y determinar si una secuencia de eventos es válida. Un tercer ejemplo sería analizar la MEF para determinar si existen estados definidos que nunca pueden ser alcanzados.

Aunque los lenguajes declarativos tienden a emplear métodos formales en la ID, sigue existiendo valor en representaciones descriptivas y cualitativas informales, tales como texto narrativo, gráficos, diagramas y pseudocódigo. Éstas siguen siendo unos elementos importantes del ciclo de desarrollo. Cuando los modelos se desarrollan mediante representaciones informales, pasan por un proceso denominado de robustecimiento del modelo. Éste captura

las relaciones entre las representaciones cualitativas y cuantitativas a medida que el diseño madura.

Con los ingredientes adecuados, la IS puede producir representaciones digitales precisas, exactas y completas de los sistemas y los procesos que los llevarán a cabo. Los “ingredientes adecuados” son aquellos que son capaces de aprovechar el poder de la computación y la precisión de los lenguajes formales para gestionar la enorme cantidad de datos asociados con un proyecto de IS, automatizar sus procesos mediante inferencias y razonamientos y preservar la coherencia dentro del ciclo de vida del sistema. Un área que puede servir de inspiración a este respecto es la World Wide Web, que ha tenido un enorme progreso en los últimos 30 años en términos de gestión y recuperación de la información.

2.2. Las tecnologías de la Web Semántica

La Web Semántica es una extensión de la World Wide Web que tiene como objetivo hacer que los datos puedan ser leídos por las máquinas y proporcionar una estructura estándar para la representación y el razonamiento de datos [6]. En el centro de esta evolución está la transición de una web que “consistía principalmente en documentos para que los humanos los leyeran a una que incluye datos e información para que los ordenadores los manipulen” [7]. Un ejemplo típico podría ser la evolución de los motores de búsqueda que, lejos de ser un simple algoritmo de rastreo web basado en palabras clave, explotan esta hiperconectividad entre datos para entender el significado detrás de las consultas de búsqueda, emplean algoritmos complejos basados en aprendizaje automático para entender la intención del usuario y a menudo generan una respuesta personalizada en lenguaje natural.

Las tecnologías asociadas a la Web Semántica son las que proporcionan estas capacidades. Cuentan con formas estandarizadas de representar datos con un formato de tuplas de sujeto-predicado-objeto que permiten describir conocimientos tales como “Curiosity es un Rover de Marte” o “Curiosity ejecuta la función ‘analizar suelo’”. De esta manera, es posible construir redes complejas de datos interconectados que se basan completamente en este patrón de tuplas. Al utilizar una ontología, estas redes pueden ser aprovechadas para verificar la validez de los datos, así como para inferir nueva información que no hemos declarado explícitamente. Por ejemplo, consideremos la ontología en la Figura 3. En ella, se definen tres clases: Sistema, Función y Rover de Marte. También se ha afirmado que el Rover de Marte es una subclase de Sistema. Una propiedad de objeto (una relación entre dos clases) llamada ‘ejecuta’ especifica que el dominio solo puede ser un Sistema, y el rango solo puede ser una Función. Esta ontología se puede utilizar para

validar el conjunto de datos presentado anteriormente (es decir, “Curiosity es un Rover de Marte” y “Curiosity ejecuta la función ‘Analizar Suelo’”) y para inferir nueva información que no se había declarado previamente. Debido a que Curiosity es un Rover de Marte y la ontología específica que el Rover de Marte es una subclase de Sistema, es posible inferir que Curiosity también es un Sistema. Además, debido a que Curiosity (un Sistema) ejecuta ‘Analizar Suelo’, se puede inferir que ‘Analizar Suelo’ es una Función. La validez del conjunto de datos se confirma porque no se han violado reglas en la ontología. Si hubiéramos afirmado que ‘Analizar Suelo’ ejecuta ‘Curiosity’, nuestro conjunto de datos se habría declarado inconsistente, ya que la relación ‘ejecuta’ solo puede ser una Función.

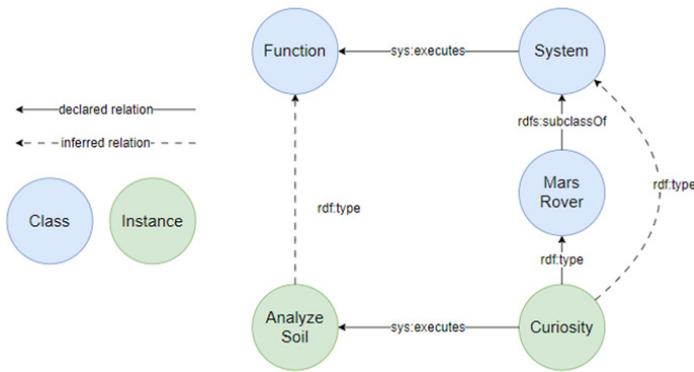


Figura 3. Ejemplo de Grafo de Conocimiento [Legendas: rdf: Resource Description Framework; rdfs: Resource Description Framework Schema; sys: System Ontology]

Como ejemplo, la Figura 4 muestra una representación parcial de la ontología que soporta el modelado del diseño del arnés para una nave espacial.

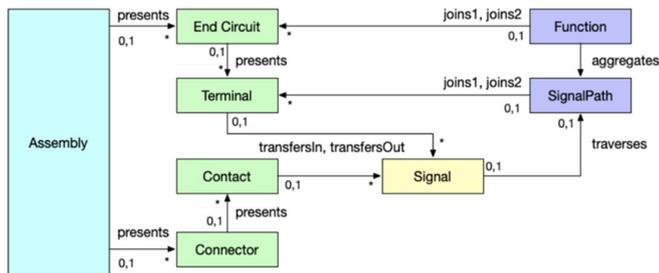


Figura 4. Representación Parcial de una Ontología de Diseño de un Arnés, de Wagner et al. 2020

3. GESTIÓN DEL CICLO DE VIDA BASADO EN MODELOS Y EL ASEGURAMIENTO DE LA CALIDAD

En esta sección, nos enfocamos en cómo los enfoques modernos de ID pueden apoyar el aseguramiento de la calidad en el del ciclo de vida de los sistemas, esto es, la verificación, validación y la certificación. Presentamos ejemplos existentes en la literatura sobre trazabilidad a través de objetos de diseño, verificación y validación (V&V) tempranas y revisiones basadas en modelos.

3.1. Trazabilidad de los objetos de diseño de sistemas

La trazabilidad se refiere a la capacidad de establecer relaciones explícitas entre elementos del diseño del sistema (por ejemplo, un valor de cantidad, una decisión de diseño) a lo largo del ciclo de vida del sistema. En IS, la trazabilidad de requisitos es un típico ejemplo y permite que todos los elementos del sistema, las decisiones de diseño y los casos de prueba puedan rastrearse hasta un requisito particular (o conjunto de requisitos). El propósito de esto es poder garantizar que los resultados del diseño logren ser explicados y verificados para su revisión y auditoría.

La ID, y particularmente el hilo digital, permiten una trazabilidad sin precedentes a lo largo del ciclo de vida del sistema. Uno de los principales beneficios de este enfoque es que los datos pueden ser definidos una vez en una Fuente Única de la Verdad (FUV) y luego distribuidos a lo largo del hilo digital hacia otros modelos y objetos que requieran acceso a los datos. Una FUV proporciona una línea base de los datos y debería ser accesible, por ejemplo, para aplicaciones autorizadas que tengan la intención de utilizar los datos en algún análisis o toma de decisión. De esta manera, los ingenieros pueden estar seguros de que las aplicaciones a lo largo de todo el hilo digital (representando así todo el ciclo de vida del sistema) están utilizando un conjunto válido de datos. El establecimiento de una FUV dentro de un hilo digital es una manera efectiva de evitar inconsistencias.

De forma similar, la trazabilidad dentro de un hilo digital permite la identificación de los caminos de propagación de cambios. Si consideramos un cambio en un requisito del sistema, por ejemplo, la trazabilidad a lo largo del hilo digital permite a los ingenieros identificar los elementos del sistema y las pruebas que se verán afectadas, entre otros. Para lograr este grado de trazabilidad se debe considerar

tanto la interoperabilidad de datos como la técnica [9]. La interoperabilidad de datos asegura que exista una comprensión consistente de la terminología relevante y una estructura estándar a la cual los datos puedan ser mapeados. La interoperabilidad técnica resuelve el problema de obtener datos de múltiples fuentes en la misma base de datos o al menos proporciona conexiones punto a punto entre herramientas relevantes. La interoperabilidad técnica puede mejorarse mediante el uso de protocolos estándares de integración como API REST. El Open Services for Lifecycle Collaboration (OSLC) es una comunidad que desarrolla y publica estándares de código abierto destinados a mejorar la integración utilizando dicho protocolo [10].

La Arquitectura Dragon, en desarrollo por NASA JPL, es un entorno de ID que comprende múltiples sistemas y aplicaciones de ingeniería de software e implementa el hilo digital conectando las herramientas entre sí en una estructura orientada a grafos, enfatizando relaciones explícitas para formar un modelo heterogéneo integrado [11] (ver Figura 5). El uso de FUVs y la interoperabilidad técnica son aspectos claves de la Arquitectura Dragon. Permite al usuario definir relaciones entre herramientas, proporcionando así

conectividad que integra requisitos, arquitectura, diseño detallado de hardware y software y casos de prueba, entre otros. La Arquitectura Dragon logra esto mediante la aplicación de tecnología y estándares de interoperabilidad basados en Internet y la World Wide Web, siendo este uno de sus principios fundamentales.

Aunque no siempre sea necesario, las ontologías pueden ayudar con la interoperabilidad de datos. Sin embargo, la interoperabilidad en Dragon sí que requiere la expresión de semántica para conseguir integración e interoperabilidad, enfocándose especialmente en las relaciones. A menudo, la ontología está asociada con la Web Semántica y el lenguaje de ontología web. Si bien ésta es ciertamente una forma válida de modelar una ontología, no es la única forma en la que se puede especificar. La aplicación de ontologías a la IS tiene como objetivo "permitir un intercambio más automatizado de información entre modelos para garantizar su consistencia, mejorar el rigor del proceso de ingeniería y, en última instancia, reducir el esfuerzo necesario para obtener una respuesta clara a las preguntas que hace la ingeniería" [8]. La Digital Engineering Factory (DEF) es un entorno de ID en desarrollo en la Universidad de Arizona

Dragon Framework

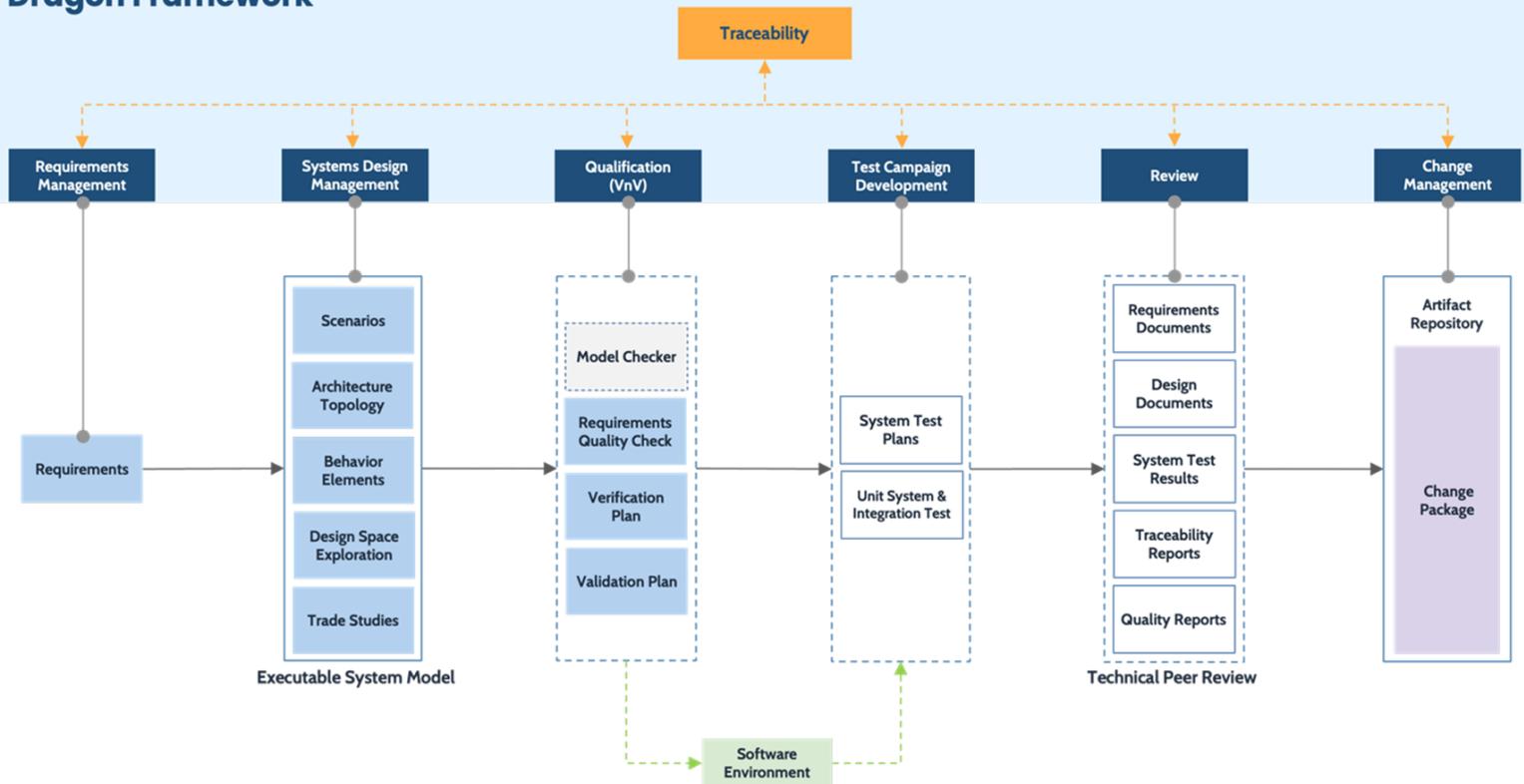


Figura 5. Vista Funcional de la Arquitectura Dragon. Esta es una ilustración destinada a comunicar el flujo general de trabajo y las dependencias conceptuales. No tiene la intención de describir un proceso. Todas las funciones son realizables con tecnología off-the-shelf, excepto la verificación de modelos

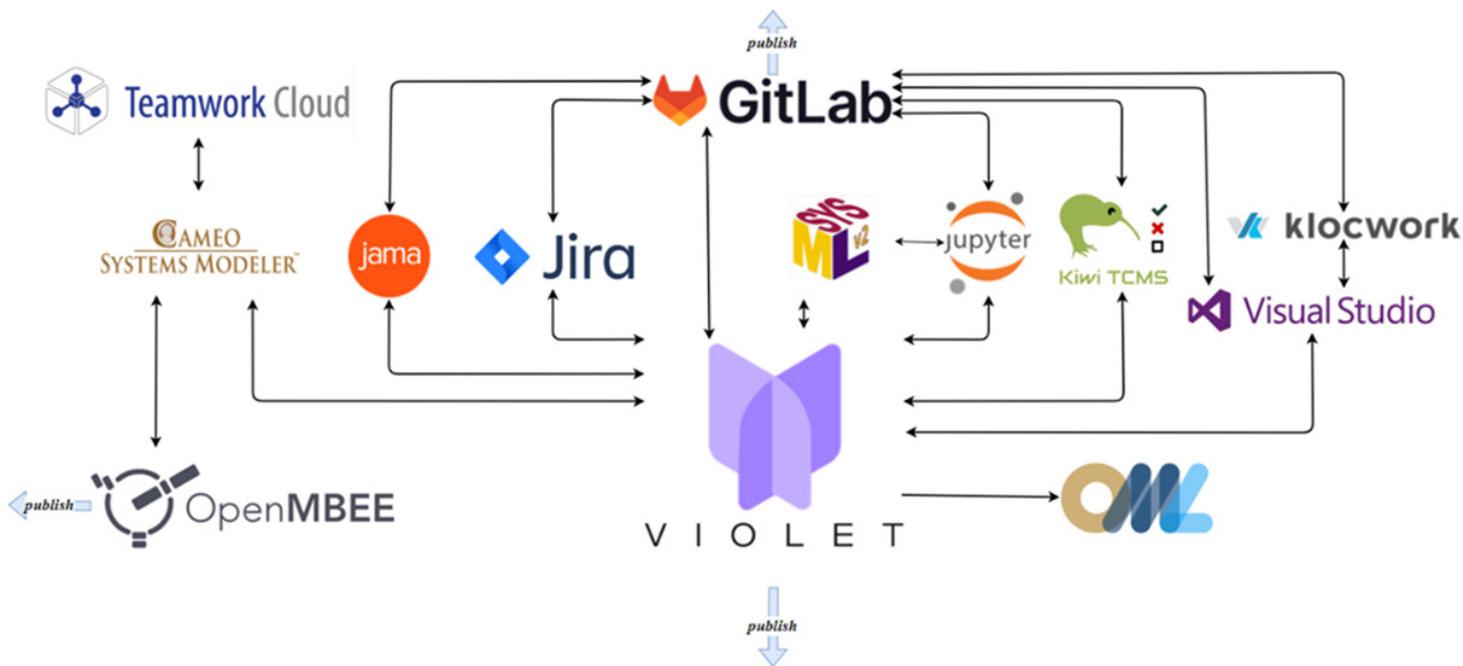


Figura 6. La Digital Engineering Factory (DEF), adaptada de [9]

que permite a los usuarios integrar datos de múltiples herramientas y estructurarlos de acuerdo con la University of Arizona Ontology Stack (UAOS) [9]. Una pila de ontologías es una jerarquía estructurada, basada en la misma ontología de nivel superior, para respaldar la interoperabilidad de datos entre dominios. El DEF utiliza una arquitectura hub-and-spoke para integrar datos de múltiples herramientas en una base de datos central (alojada en la herramienta Violet). Esto permite a los usuarios definir conexiones entre herramientas, apoyando así la trazabilidad entre requisitos, arquitectura del sistema, diseño detallado de hardware y software, actividades de verificación y gestión de proyectos (ver la Figura 6).

3.2. Verificación y validación tempranas

La conectividad potencial ofrecida por el hilo digital y sus tecnologías habilitadoras también hacen posible adelantar la V&V en el ciclo de vida [16]. Por ejemplo, el modelado de la arquitectura del sistema en las primeras etapas de diseño con un lenguaje de modelado ejecutable permite la ejecución de simulaciones y los análisis cuantitativos resultantes sirven para evaluar el cumplimiento de los requisitos [15]. Esto tiene un valor indudable porque “encontrar problemas temprano es un habilitador clave para que la ID proporcione el máximo de su valor potencial” [13]. Varios lenguajes de modelado, pero no todos, proporcionan estas capacidades ejecutables.

Por ejemplo, con SysML, el comportamiento capturado en los diagramas de actividades y de máquinas de estados puede ser simulado.

Además del uso de métodos formales para ejecutar y simular modelos, que implican una representación completa y matemáticamente fundamentada de un sistema, los métodos formales ligeros también pueden contribuir a la V&V tempranas. Éstos implican la representación matemática de una parte de la especificación del sistema. De esta manera, proporcionan algunos de los beneficios de los métodos formales, como la capacidad para detectar errores en las primeras etapas del desarrollo del sistema, pero sin necesidad de redefinir toda la especificación del Sistema [20].

“Una de las fortalezas de utilizar enfoques de ID para el diseño conceptual es que los diversos modelos, herramientas y FUVs están todos conectados a través de un hilo digital. Comparten los mismos datos consistentes y autorizados. Durante el diseño conceptual, una utilidad de este hilo digital es conectar modelos de simulación con modelos de definición de sistemas... y compartir de manera inequívoca requisitos, trazabilidad de requisitos y comportamientos del sistema” [21].

3.3. Revisiones basadas en modelos y aprobaciones digitales

La digitalización de los procesos de IS también se extiende a los principales hitos del ciclo de vida del sistema: las revisiones técnicas. Los procesos tradicionales de revisión basados en documentos han sido utilizados por los ingenieros de sistemas durante muchos años para determinar si los procesos de desarrollo pueden avanzar a las siguientes etapas. Sin embargo, este enfoque tradicional a menudo conduce a “largas evaluaciones de documentos exigidas por contrato “ que “representan visiones puntuales de los sistemas tal y como se ven a través del prisma de los criterios de entrada [a la siguiente fase del ciclo de vida], pero que no representan una visión del sistema en su totalidad” [22]. Este proceso no solo es ineficiente, sino que también a menudo carece de la capacidad de ofrecer la información del sistema que realmente interesa a los stakeholders.

Las revisiones basadas en modelos tienen el potencial de ofrecer mejoras en este sentido (ver [24, 25]). En primer lugar, los revisores pueden aprobar subconjuntos de datos/información en lugar de tener que revisar y aprobar cantidades de información en documentos dispersos y, a menudo, superpuestos. Esto se ha denominado aprobaciones digitales, donde incluso las aprobaciones requeridas pueden asignarse a diferentes fragmentos de datos para que sean versionados, aceptados y/o aprobados [24]. En segundo lugar, los revisores pueden interactuar con la pila de ontologías interoperables para encontrar la información que consideran relevante para la revisión, en lugar de tener que buscar en múltiples documentos. Por ejemplo, un revisor podría consultar el repositorio de ID para identificar todas las pruebas de verificación relacionadas (es decir, trazadas) con un requisito, identificar todas las funciones que participan en una capacidad relacionada con una necesidad específica de un stakeholder, o identificar las actividades pendientes de integración y prueba para un componente específico. En ambos casos, es fácil darse cuenta de cómo el uso de la ID para apoyar las revisiones técnicas puede resultar en un proceso más ágil y fluido, al tiempo que potencialmente puede mejorar su eficiencia y reducir la probabilidad de que queden partes sin revisar.

No obstante, no se debe ocultar que las revisiones basadas en modelos también presentan desafíos. Entre otros, los modelos deben ser integrales, no solo limitados a dibujos tradicionales en 3-D; existe una curva de aprendizaje significativa, ya que las personas involucradas no solo necesitan aprender los nuevos lenguajes y herramientas de modelado, sino también los nuevos procesos que se emplean en una revisión basada en modelos; el éxito de la revisión depende de la capacidad de los revisores para identificar la información necesaria en el repositorio digital [23]. La adopción de la ID puede ayudar a mitigar algunos

de estos desafíos. La interoperabilidad técnica y de datos proporciona a los ingenieros una base de datos común de la que se puede recuperar la información del proyecto. Las tecnologías de la Web Semántica pueden aprovecharse para validar automáticamente este conjunto de datos. Además, las consultas para recuperar y presentar convenientemente la información clave pueden estandarizarse y automatizarse (por ejemplo, en un documento o cuadro de mando generado automáticamente).

4. MÁS ALLÁ DE LA TRAZABILIDAD Y DE LA GESTIÓN DEL CICLO DE VIDA

Hasta ahora, hemos discutido cómo ID puede ser utilizada para mejorar la trazabilidad, permitir la V&V tempranas y apoyar el proceso de revisión basado en modelos. Sin embargo, la ID no es únicamente trazabilidad y gestión. En esta sección, proporcionamos ejemplos de cómo la ID puede soportar un conjunto de análisis sin precedentes, inviables en las prácticas basadas en documentos. Estos incluyen la exploración del espacio de soluciones, el diseño basado en conjuntos, el diseño óptimo multidisciplinar, la integración de modelos físicos dentro de modelos de sistemas y la integración hombre-sistema.

4.1. El paradigma de la Exploración del Espacio de Soluciones

Cuando se desarrolla un nuevo sistema, es crucial centrar los esfuerzos durante las primeras etapas del ciclo de vida en elegir una solución de diseño adecuada. Dado que evaluar todo el conjunto de posibles soluciones requiere una inversión significativa, este análisis generalmente no es exhaustivo debido a la falta de recursos necesarios. Como resultado, se toman decisiones precipitadas y generalmente no justificadas para reducir las opciones de diseño y solo se evalúan unas pocas alternativas. En consecuencia, a menudo se pasan por alto alternativas que podrían ser más valiosas que la elegida.

La Exploración del Espacio de Soluciones es un método que aprovecha los modelos digitales y las altas capacidades de computación actuales para realizar un análisis rápido y exhaustivo del espacio de posibles soluciones [26]. Mediante esta estrategia, se exploran múltiples opciones de diseño como arquitecturas candidatas y se analizan para determinar la más adecuada. En esencia, en lugar de centrarse en encontrar opciones de diseño específicas, el ingeniero define un modelo genérico del sistema que puede ser automáticamente instanciado como la enumeración de

diferentes componentes y valores que pueden tomar sus principales variables. Cada solución resultante se evalúa con respecto a un conjunto de criterios de decisión. Entre ellos, se puede identificar una Frontera de Pareto, formado por aquellas soluciones que dominan el espacio de soluciones (es decir, no hay ninguna otra que funcione mejor en todos los criterios) para elegir aquella deseada dentro del conjunto.

Lo ilustramos con un ejemplo. Consideremos el diseño de un nuevo carro de combate con las siguientes necesidades deseadas: alta capacidad de transporte, alta protección y movilidad aceptable. La arquitectura genérica del carro incluye Armadura, Sistema de Armas y Propulsión como sus principales sistemas, cada uno con diferentes opciones de diseño (Armadura: Acero convencional, Cerámica o Compuesto; Sistema de Armas: Cañón de 120 mm o Lanzamisiles guiados; Propulsión: Motor eléctrico híbrido o Motor de combustión interna). Además, cada instancia puede tener diferentes valores de rendimiento, como tener diferentes masas. Considerando la masa como un sustituto del rendimiento general de las diferentes variantes de las diferentes tecnologías (en este caso, un total de 71 componentes de nivel inferior), es posible enumerar todas las arquitecturas instanciadas posibles (por ejemplo, Acero Convencional - Rendimiento 1/Cañón de 120 mm Rendimiento 1/Motor Eléctrico Híbrido Rendimiento 1, Acero Convencional - Rendimiento 1/Cañón de 120 mm Rendimiento 1/Motor Eléctrico Híbrido Rendimiento 2, Acero Convencional - Rendimiento 1/Cañón de 120 mm Rendimiento 1/Motor Eléctrico Híbrido Rendimiento 3, ... para un total de 3,408 arquitecturas) y representarlas según su coste (cuanto mayor sea el coste, menos interés tendrá la alternativa) y valor proporcionado en operaciones (cuanto mayor sea el valor, más interés tendrá la alternativa; téngase en cuenta que ambos se han calculado utilizando modelos teóricos), identificando luego la Frontera de Pareto (ver la Figura 7, donde cada línea aparente es en realidad el efecto de muchos puntos cercanos entre sí).

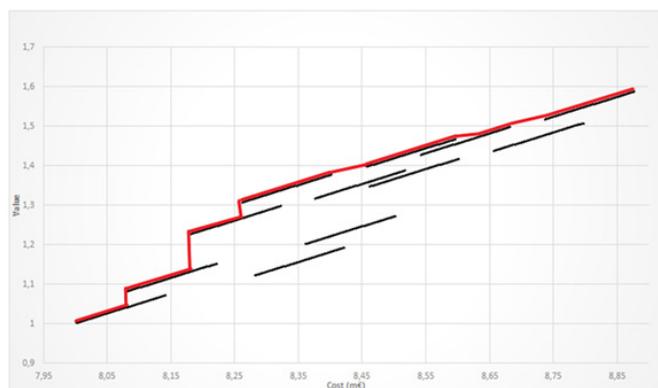


Figura 7. Espacio de Soluciones para el Ejemplo del Carro de Combate relacionando Coste y Valor (Frontera de Pareto resaltado)

4.2. El paradigma del Diseño Basado en Conjuntos

Tradicionalmente, el Diseño Basado en Punto ha sido el utilizado para el desarrollo de nuevos sistemas. Comienza con un análisis de alternativas, teniendo como resultado la selección de un solo concepto, que se va refinando a medida que avanza el desarrollo. Cuando surge un problema, la solución se modifica según sea necesario, lo que generalmente resulta en incrementos de costes y retrasos, mayores cuanto más tarde se realice la modificación.

En contraste con este rígido enfoque, el Diseño Basado en Conjuntos (DBC) fomenta mantener abiertas múltiples opciones y evaluar las ventajas e inconvenientes de cada diseño en lugar de converger rápidamente en un diseño único [30–32]. Esto facilita la exploración exhaustiva del espacio de diseño y promueve la innovación al postergar el compromiso con un solo concepto. El espacio de soluciones se estrecha progresivamente a medida que se adquiere más conocimiento e información sobre el sistema (ver Figura 8). Esto reduce el riesgo de conceptos e hipótesis erróneos asociados a decisiones tempranas en el proceso de diseño. Las decisiones críticas se retrasan hasta que haya suficiente información y conocimiento disponible para ganar flexibilidad. Además, como subproducto de trabajar con conjuntos de soluciones en lugar de con soluciones únicas, el DBC proporciona una mayor flexibilidad para adaptarse a circunstancias imprevistas. El uso de potentes computadoras con alta capacidad de cálculo y el modelado orientado a objetos desempeña un papel significativo en el desarrollo y la evaluación simultánea de varios conceptos de diseño a un coste relativamente bajo.

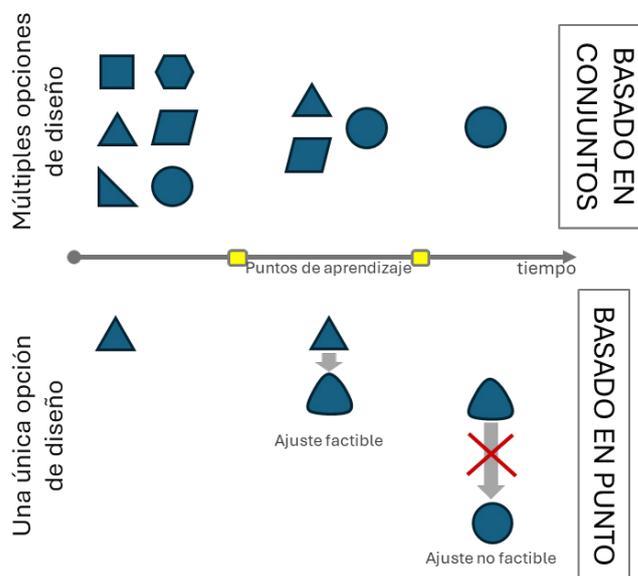


Figura 8. Proceso de Maduración del Diseño Basado en Conjuntos

Un típico proceso de DBC es el siguiente (ver Figura 9):

1. Definición del espacio de diseño, identificando un conjunto de alternativas viables que se desarrollarán progresivamente de manera independiente, incluso por diferentes grupos de ingeniería, generando así conceptos desde diversas perspectivas. Es crucial considerar un gran número de alternativas para su evaluación.
2. Descubrimiento de intersecciones entre diferentes conjuntos de soluciones independientes para definir el núcleo del nuevo diseño progresivamente.
3. Eliminación de forma progresiva de los conceptos que se demuestran incompatibles o que aportan un valor limitado.

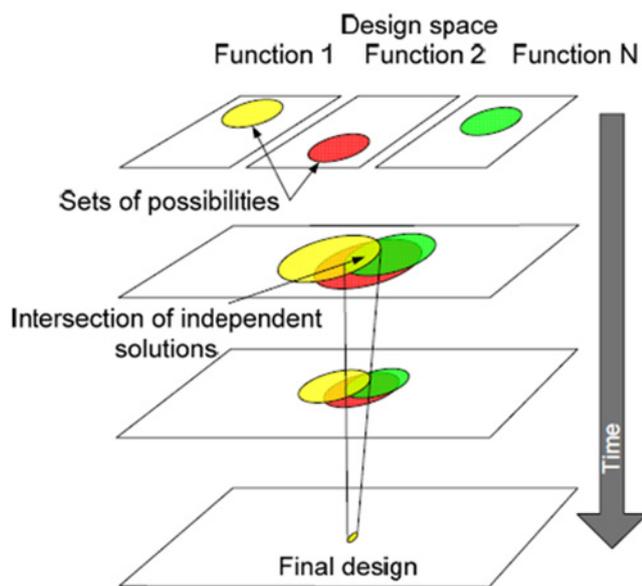


Figura 9. Descubrimiento de intersecciones en el proceso de Diseño Basado en Conjuntos, adaptada de Raudberget, 2010

4.3. El paradigma del Diseño Óptimo Multidisciplinar

Tradicionalmente, la descomposición de sistemas se realiza fijando manualmente las características de los componentes que forman el sistema y repitiendo el proceso entre ellos hasta que su integración lleva a una solución satisfactoria del sistema. Esto se debe al acoplamiento que generalmente existe entre los diferentes componentes de un sistema. Por ejemplo, un equipo que diseña un satélite puede comenzar estimando el consumo de energía eléctrica de cada componente. Luego, agregarían la energía disipada en los diferentes paneles estructurales del satélite y utilizarían esa información para diseñar el sistema térmico (para

la disipación). El dimensionamiento del sistema térmico puede retroalimentar el dimensionamiento de los paneles estructurales, y así sucesivamente. Este proceso iterativo generalmente se realiza manualmente: propagando las diferentes modificaciones hasta que algo parece prometedor para funcionar.

En lugar de eso, en el Diseño Óptimo Multidisciplinar (DOM) el objetivo es encontrar una combinación óptima de componentes gestionando objetivos en conflicto, no solo una solución satisfactoria [34]. El DOM aprovecha los modelos computacionales para ejecutar algoritmos de optimización. En particular, el método consiste en integrar modelos matemáticos de los componentes y ejecutar una función de optimización global a nivel del sistema. De esta manera, el diseño de los componentes no llevaría solo a una solución satisfactoria, sino a un sistema óptimo. El trabajo del equipo de ingeniería pasa de iterar las características acopladas a desarrollar modelos precisos de sus componentes (y su integración), de modo que el ordenador pueda manejar las iteraciones por sí mismo.

El DOM no solo identifica una solución óptima, sino que lo hace de manera más eficiente que la iteración humana y garantiza la alineación entre las decisiones a nivel de componentes y los objetivos del proyecto a nivel de sistema. Sin embargo, es importante tener en cuenta que la robustez de la solución se vuelve crucial, ya que la precisión de los modelos subyacentes para realizar la optimización probablemente sea limitada en la etapa en la que se emplea la DOM.

4.4. Conectando modelos de sistema con modelos basados en la física

La conexión de los modelos de sistemas con modelos basados en la física es una capacidad común aprovechada dentro de un hilo digital. En esencia, los modelos de comportamiento y estructurales de una arquitectura de sistema pueden conectarse a sus instancias de diseño en modelos basados en la física, propagando automáticamente restricciones (en una dirección) y resultados (en la otra dirección) entre los diferentes modelos, garantizando la consistencia y la fiabilidad de los datos [11].

Por ejemplo, la Figura 10 muestra el uso de la Arquitectura Dragon para soportar la arquitectura y el diseño de un Rover. La adopción del uso de lenguajes informáticos permite que descripciones informales y cualitativas se transformen en representaciones ejecutables muy precisas, basadas en las matemáticas y la física. En este caso, la arquitectura del sistema se modela utilizando SysML, que se conecta a su modelo basado en la física, establecido en Modelica. SysML puede representar de manera efectiva el comportamiento

funcional y las interfaces del Rover, mientras que Modelica hace lo mismo para el comportamiento físico y las interfaces de manera ejecutable, lo cual no es posible cuando se utilizan lenguajes naturales, diagramas informales o representaciones gráficas. En conjunto, ambos pueden usarse para representar cómo se pueden transmitir señales

para ejecutar la respuesta física del Rover basada en el comportamiento en el modelo de sistemas. Este ejemplo muestra una forma de realizar la V&V de las Especificaciones de Comportamiento de Potencia y Conducción del Rover contra una representación física del producto.

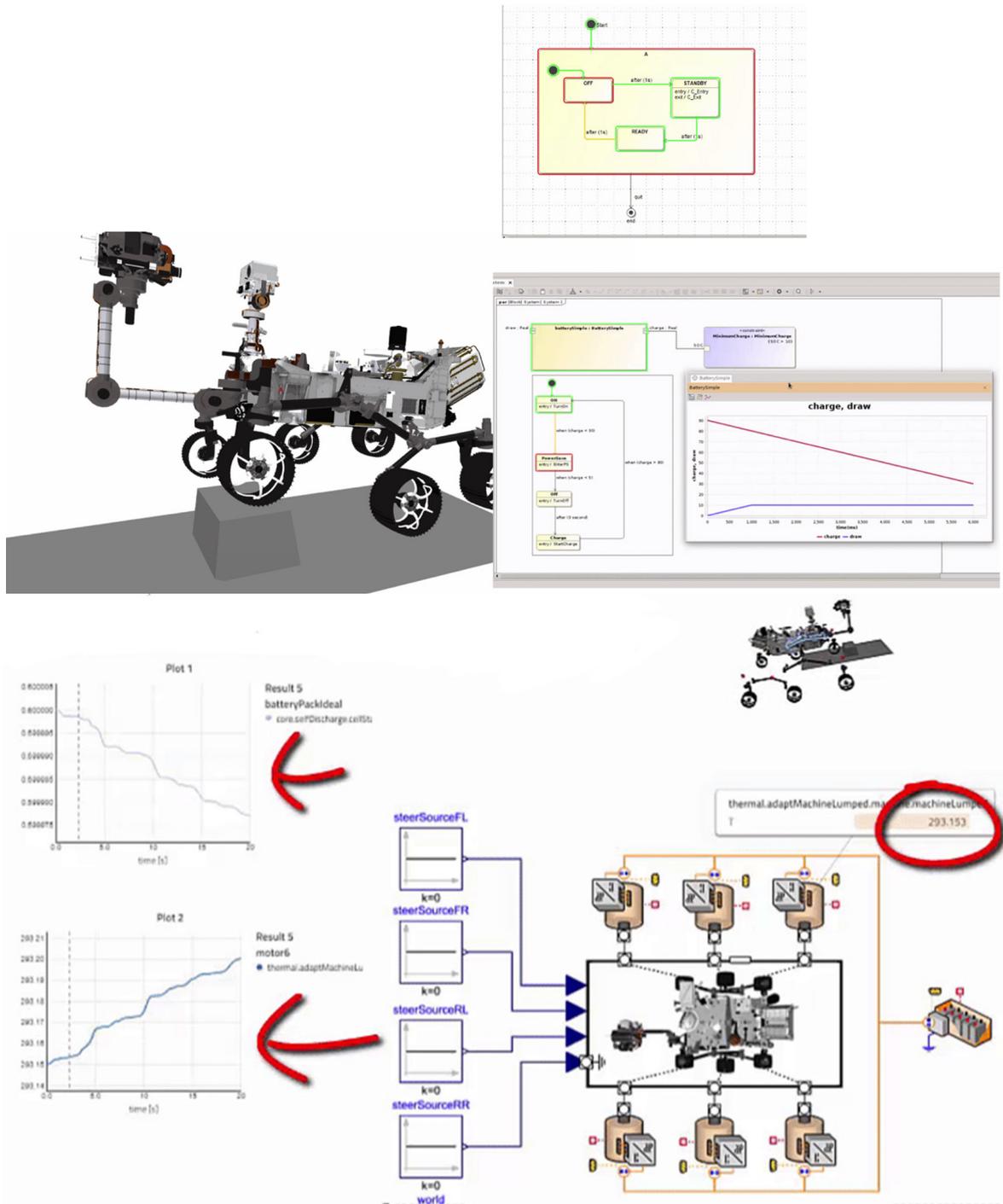


Figura 10. Iteraciones del Proceso de Desarrollo Basado en Modelos, incluyendo la Simulación Multi-Física. Soportado por la Arquitectura Dragon, de Karban et al. 2023

4.5. Asistentes Cognitivos

Los avances en inteligencia artificial y la potencia de cálculo disponible hoy en día están permitiendo el desarrollo de asistentes cognitivos para tareas de IS. Un asistente cognitivo es una máquina que actúa como un ingeniero virtual, un compañero senior con quien el ingeniero humano trabaja y que realiza tareas de ingeniería que los humanos no son capaces de llevar a cabo de manera efectiva o eficiente.

Con un asistente cognitivo actuando como un ingeniero virtual, el ordenador pasa de ser una herramienta en la que introducimos datos, creamos modelos y ejecutamos simulaciones, a ser un compañero de trabajo altamente experimentado con quien diseñamos conjuntamente, a quien asignamos algunas de nuestras tareas, en quien confiamos para comprender mejor qué decisiones de ingeniería debemos tomar y quien asegura que el conocimiento generado en nuestro proyecto se almacene dentro de la organización para que otros colegas y nosotros mismos podamos beneficiarnos de ello en proyectos futuros [28].

En lugar de evaluar opciones de diseño para un sistema, le preguntaremos a nuestro compañero virtual, *¿Qué piensas de este diseño?*, *¿Nos falta algún requisito clave?*, *¿Qué riesgo crees que asumimos si no realizamos esta prueba?*, o *¿Nos costaría mucho reemplazar esta interfaz sólo con un simple comando?*

Los asistentes cognitivos permiten la extracción y análisis de grandes cantidades de datos de diversas fuentes casi instantáneamente y pueden proporcionar al ingeniero humano las respuestas que necesita directamente en lenguaje natural. Los asistentes cognitivos también pueden iniciar conversaciones de manera unilateral cuando creen que pueden ser útiles para el ingeniero humano, como proporcionar sugerencias para mejorar una arquitectura en la que se está trabajando actualmente. Se muestra una interfaz de usuario de ejemplo en la Figura 11. Al interpretar modelos de sistemas y basarse en ontologías, los asistentes cognitivos pueden mejorar sus recomendaciones y rendimiento al ejecutar estas funciones.

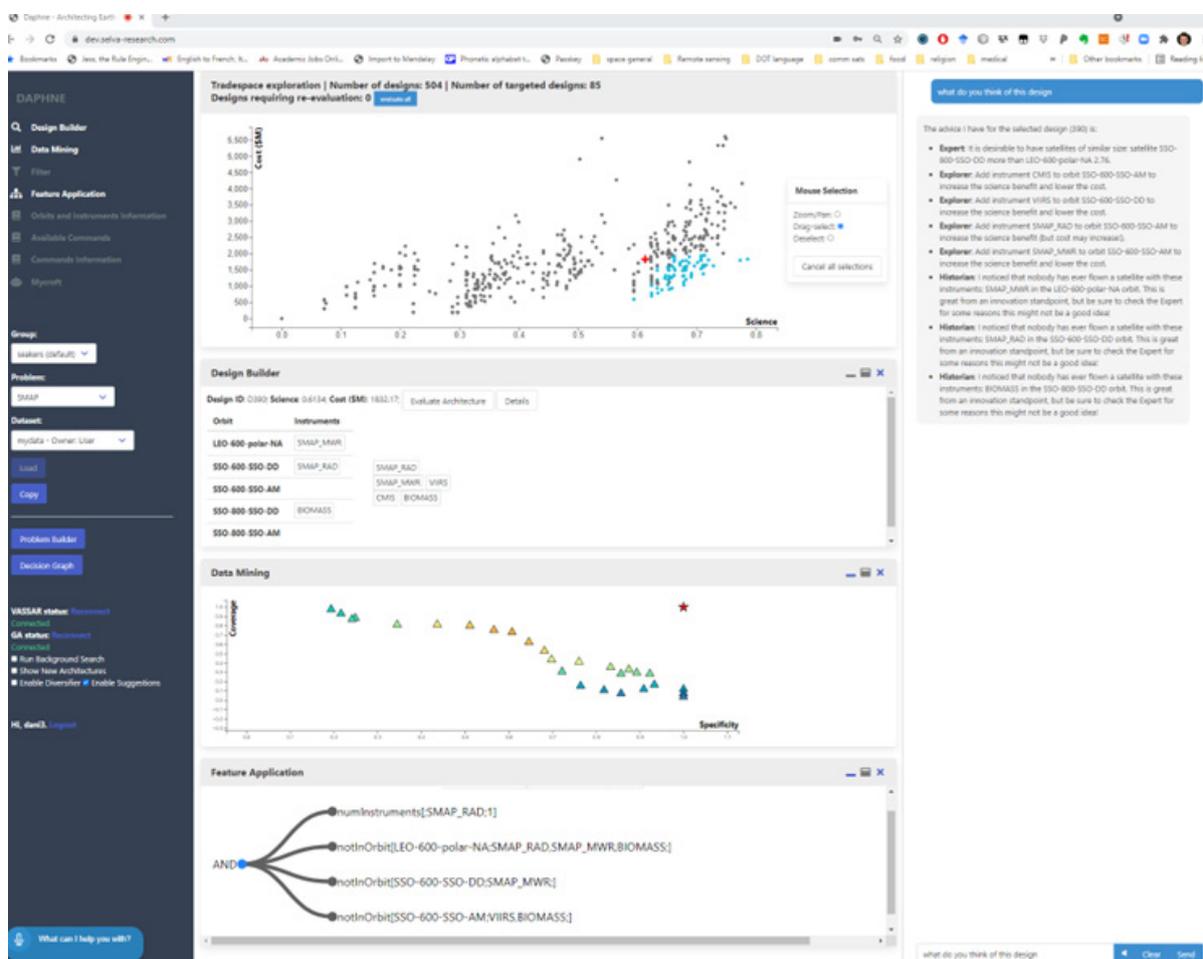


Figura 11. Interfaz de usuario de Daphne, un asistente cognitivo que apoya la arquitectura de sistemas espaciales, de Virós y Selva, 2019

5. CONCLUSIÓN

Este capítulo ha descrito cómo los avances en la computación y las tecnologías digitales brindan a las organizaciones la oportunidad de ir más allá de la simple digitalización, mediante la “Transformación Digital” de sus procesos. La ID, se ocupa de crear conjuntos de datos interoperables, de acceso rápido, que pueden integrarse eficientemente en todo el ciclo de vida de los sistemas en lo que se conoce como el “Hilo Digital”. Los lenguajes declarativos y la interoperabilidad basada en Internet proporcionan las bases para gestionar los aspectos de interoperabilidad de datos y los estándares abiertos respaldan la interoperabilidad técnica. Las tecnologías de la Web Semántica pueden aprovecharse para proporcionar el marco para su implementación.

A lo largo del capítulo, se han presentado múltiples ejemplos. El hilo digital puede proporcionar una trazabilidad sin precedentes en todo el ciclo de vida, es un facilitador de la V&V tempranas y apoya la revisión del diseño basada en modelos. Además, la ID da lugar a capacidades de ingeniería de sistemas sin precedentes, como la exploración del espacio de soluciones, el DBC, la DOM, la integración temprana de modelos basados en física durante la arquitectura del sistema e incluso la adopción de asistentes cognitivos gracias a los avances en inteligencia artificial. Todo esto contribuye a aumentar la efectividad y eficiencia del desarrollo del sistema.

1. "DoD Instruction 5000.89: Test and Evaluation," 2020.
2. V. Singh and K. E. Willcoxy, "Engineering design with digital thread," AIAA/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. 2018, pp. 1–21, 2018.
3. Z. Scott, "Speaking in Tongues: The Systems Engineering Challenge," INCOSE Int. Symp., vol. 29, no. 1, pp. 836–849, 2019.
4. P. De Saqui-Sannes, R. A. Vingerhoeds, C. Garion, and X. Thirioux, "A Taxonomy of MBSE Approaches by Languages, Tools and Methods," IEEE Access, vol. 10, no. October, pp. 120936–120950, 2022.
5. P. Witherell, S. Krishnamurty, I. R. Grosse, and J. C. Wileden, "Improved knowledge management through first-order logic in engineering design ontologies," Artif. Intell. Eng. Des. Anal. Manuf. AIEDAM, vol. 24, no. 2, pp. 245–257, 2010.
6. A. Patel and S. Jain, "Present and future of semantic web technologies: a research statement," Int. J. Comput. Appl., pp. 1–10, 2019.
7. N. Shadbolt, W. Hall, and T. Berners-Lee, "The semantic web revisited," IEEE Intell. Syst., vol. 21, no. 3, pp. 96–101, 2006.
8. D. Wagner, S. Y. Kim-Castet, A. Jimenez, M. Elaasar, N. Rouquette, and S. Jenkins, "CAESAR Model-Based Approach to Harness Design," in 2020 IEEE Aerospace Conference, 2020, pp. 1–13.
9. J. Gregory and A. Salado, "A Digital Engineering Factory for Students," in Conference on Systems Engineering Research (CSER), Tucson, AZ, USA, 2024.
10. OASIS Open Projects, "Open Services for Lifecycle Collaboration." 2023.
11. R. Karban et al., "Towards a Model-Based Product Development Process from Early Concepts to Engineering Implementation," in 2023 IEEE Aerospace Conference, 2023, pp. 1–18.
12. J. Gregory and A. Salado, "The Digital Engineering Factory: Considerations, Current Status, and Lessons Learned," in INCOSE International Symposium, Dublin, Ireland, 2024.
13. A. M. Madni and M. Sievers, "Model-based systems engineering: Motivation, current status, and research opportunities," Syst. Eng., vol. 21, no. 3, pp. 172–190, 2018.
14. R. Karban, F. G. Dekens, S. Herzig, M. Elaasar, and N. Jankevicius, "Creating system engineering products with executable models in a model-based engineering environment," Model. Syst. Eng. Proj. Manag. Astron. VI, vol. 9911, p. 99110B, 2016.

15. M. A. Bone, M. R. Blackburn, D. H. Rhodes, D. N. Cohen, and J. A. Guerrero, "Transforming systems engineering through digital engineering," *J. Def. Model. Simul.*, vol. 16, no. 4, pp. 339–355, 2019.
16. K. Giammarco and K. Giles, "Verification and validation of behavior models using lightweight formal methods," *Discip. Converg. Syst. Eng. Res.*, pp. 431–447, 2017.
17. R. Stevens, "Digital Twin for Spacecraft Concepts," *IEEE Aerosp. Conf. Proc.*, vol. 2023-March, pp. 1–7, 2023.
18. W. K. Vaneman, R. Carlson, and C. Wolfgeher, "Defining a Model-Based Systems Engineering Approach for Milestone Technical Reviews," 2019.
19. M. R. Blackburn and B. Kruse, "Conducting Design Reviews in a Digital Engineering Environment," *Insight*, vol. 25, no. 4, pp. 42–46, 2022.
20. V. Romero, R. Pinquie, and F. Noel, "An Open Benchmark Exercise for Model-Based Design Reviews," in *IFIP International Federation for Information Processing*, 2023, pp. 176–185.
21. R. Pinquié, V. Romero, and F. Noel, "Survey of Model-Based Design Reviews: Practices & Challenges?," *Proc. Des. Soc.*, vol. 2, pp. 1945–1954, 2022.
22. A. M. Ross and D. E. Hastings, "The tradespace exploration paradigm," *15th Annu. Int. Symp. Int. Counc. Syst. Eng. INCOSE 2005*, vol. 2, pp. 1706–1718, 2005.
23. N. Shallcross, G. S. Parnell, E. Pohl, and E. Specking, "Set-based design: The state-of-practice and research opportunities," *Syst. Eng.*, vol. 23, no. 5, pp. 557–578, 2020.
24. D. J. Singer, N. Doerry, and M. E. Buckley, "What is set-based design?," *Nav. Eng. J.*, vol. 121, no. 4, pp. 31–43, 2009.
25. C. Small et al., "A UAV Case Study with Set-based Design," *INCOSE Int. Symp.*, vol. 28, no. 1, pp. 1578–1591, 2018.
26. D. Raudberget, "Practical applications of set-based concurrent engineering in industry," *Stroj. Vestnik/Journal Mech. Eng.*, vol. 56, no. 11, pp. 685–695, 2010.
27. H. Chen, "Multidisciplinary Design Optimization (MDO)," *Enycl. Ocean Eng.*, no. July, 2021.
28. A. Salado and D. Selva, "Asistentes Cognitivos en Ingeniería de Sistemas," *UEM STEAM Essentials*, pp. 1–8, 2021.
29. A. Virós and D. Selva, "From design assistants to design peers: Turning daphne into an ai companion for mission designers," *AIAA Scitech 2019 Forum*, pp. 1–12, 2019.

Agradecimientos y limitación de responsabilidad

Parte de la investigación presentada en este capítulo se llevó a cabo en el Jet Propulsion Laboratory del California Institute of Technology, bajo un contrato con la National Aeronautics and Space Administration (80NM0018D0004).

La mención de cualquier producto comercial específico, proceso o servicio por su nombre comercial, marca registrada, fabricante o de cualquier otra manera, no constituye ni implica su respaldo por parte del Gobierno de los Estados Unidos o del Jet Propulsion Laboratory del California Institute of Technology.

CHRISTOPHER DELP

Chris Delp desempeña el puesto de Supervisor Técnico del Grupo de Soluciones de Sistemas e Ingeniería del Software, un grupo dedicado a implementar metodologías orientadas a procesos para la creación de entornos de software robustos diseñados para ofrecer soluciones avanzadas de ingeniería basadas en modelos digitales.



Además, ocupa el cargo de Gerente del Entorno de Sistemas de Ingeniería Asistida por Computadora en NASA JPL. Debido a su experiencia, está a la vanguardia de la innovación en ingeniería, proporcionando un entorno de última generación para la modelización de sistemas, evaluaciones analíticas y desarrollo de simulaciones basadas en los principios de Ingeniería Abierta. En su rol anterior, el Sr. Delp lideró el Desarrollo del Entorno de Modelado para la misión Europa Clipper, estableciendo un Entorno de Ingeniería Basada en Modelos y fomentando el crecimiento de la comunidad Open MBEE, un colectivo centrado en modelos, software y marcos de trabajo de Ingeniería Basada en Modelos de código abierto. Con una amplia experiencia en Ingeniería de Sistemas e Ingeniería de Software en Proyectos de Vuelo del JPL, su experiencia abarca la arquitectura y diseño de sistemas, el desarrollo y prueba de software crítico para la seguridad, y la aplicación de principios de Ingeniería de Sistemas a lo largo del ciclo de vida del proyecto. El Sr. Delp es considerado una autoridad líder en Ingeniería de Sistemas Basada en Modelos (MBSE) y Entornos de Ingeniería Basada en Modelos. Tiene un Máster en Ingeniería de Sistemas de la Universidad de Arizona.

DR. JOE GREGORY

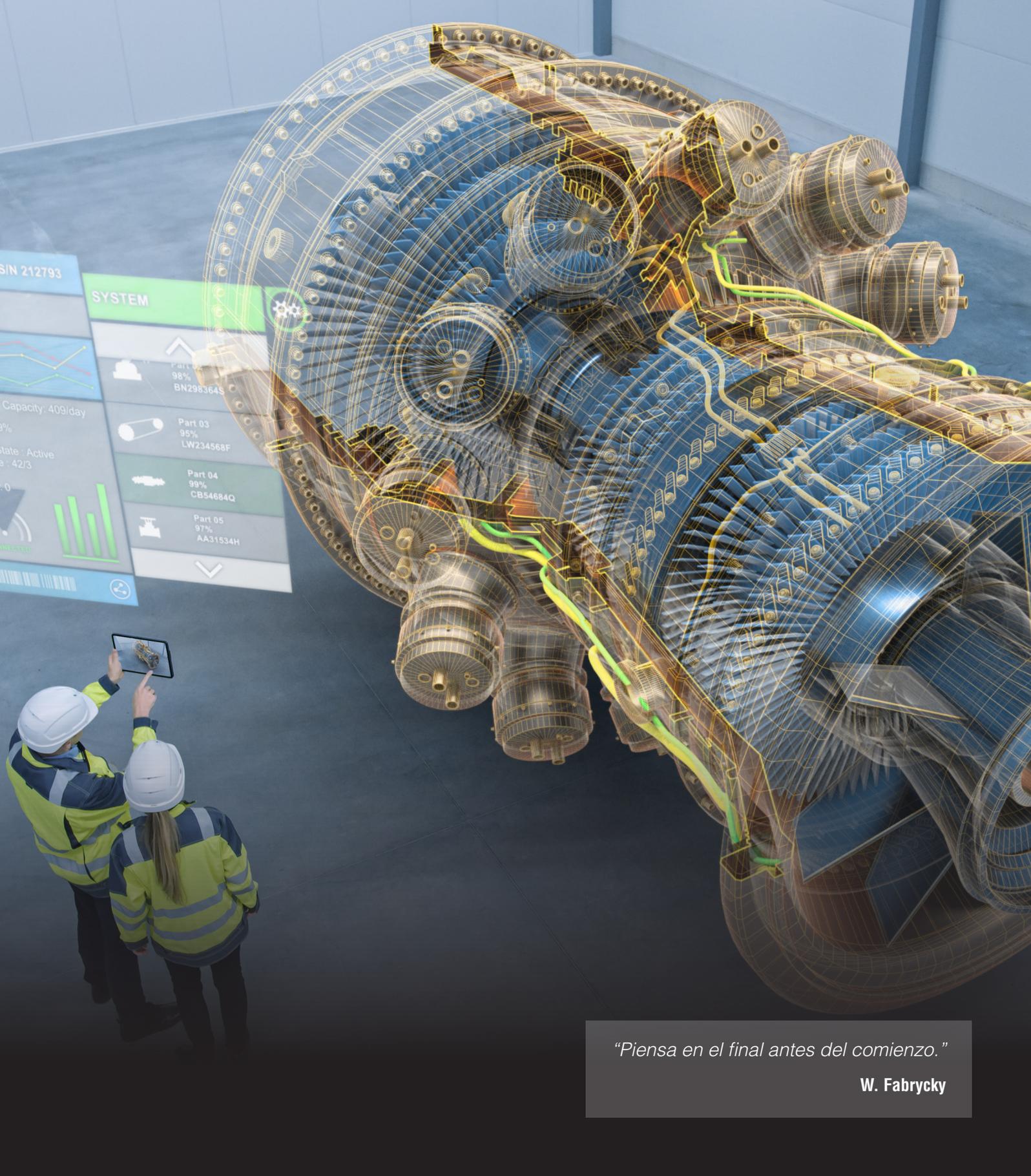
El Dr. Joe Gregory es un investigador de postdoctorado en la Universidad de Arizona. Sus áreas de interés para investigación incluyen la ingeniería digital, la ingeniería de sistemas basada en modelos y la aplicación de tecnologías de la Web Semántica para apoyar la ingeniería. En 2022, obtuvo su doctorado en Ingeniería Aeroespacial de la Universidad de Bristol por su desarrollo del 'Modelo de Análisis Temprano de Naves Espaciales' basado en SysML. Es copresidente del Grupo de Trabajo de Taxonomía del Intercambio de Información de Ingeniería Digital (DEIX).



L. MIGUEL APARICIO

Luis Miguel Aparicio tiene una licenciatura en Ingeniería Informática por la Universidad Politécnica de Madrid y es diplomado en Alta Gestión Logística por el CESEDEN. Posee varias certificaciones en logística, ingeniería de sistemas y gestión de proyectos, incluyendo Lean Six Sigma, ITIL, INCOSE, PMP y PRINCE. Después de adquirir experiencia profesional en Coritel, Telefónica y en el sector público, se unió a Isdefe en 1999, donde sigue trabajando hasta el día de hoy. Ha colaborado como profesor adjunto en la Universidad Rey Juan Carlos y actualmente en la Universidad Europea de Madrid. Su experiencia profesional siempre ha girado en torno a los sistemas de información, con un enfoque particular en logística. Actualmente desempeña el puesto de Jefe del Área de Sistemas Logísticos en Isdefe, liderando proyectos para apoyar la racionalización y modernización de los sistemas logísticos de las Fuerzas Armadas Españolas. También cuenta con una amplia experiencia en proyectos relacionados con el Sistema de Catalogación de la OTAN, brindando servicios de consultoría en este campo y apoyando el desarrollo y modernización de SICAD, el Sistema de Catalogación de Defensa, una herramienta utilizada en las Oficinas Nacionales de Catalogación de Arabia Saudita, Bélgica, Colombia, España, Perú y Polonia. Entre el portafolio de proyectos gestionados dentro de su área, se encuentran varios proyectos de consultoría internacional, por ejemplo, con la Agencia Europea de Defensa, la OTAN, la Corporación de Industrias Militares de Arabia Saudita o la Oficina Nacional de Catalogación de Jordania. También ha colaborado con las Fuerzas Armadas de Perú, Emiratos Árabes Unidos, Chile y Ecuador. Actualmente, destaca el papel activo que desempeña en el apoyo a la introducción de nuevas tecnologías disruptivas en los sistemas de información logística de Defensa, con proyectos relacionados con la transformación digital, inteligencia artificial, big data o blockchain.





SIN 212793

SYSTEM

- Part 02
98%
BN298364S
- Part 03
95%
LW234568F
- Part 04
99%
CBS4684Q
- Part 05
97%
AA31534H

Capacity: 409/day

State: Active

42/3

"Piensa en el final antes del comienzo."

W. Fabrycky

Transformación Digital en la Fase de Servicio de un Sistema

Dr. Kaitlynn Castelle, *University of Maryland Applied Research Lab
for Intelligence & Security* (kcastelle@arlis.umd.edu)

Miguel Ángel Coll Matamalas, *Isdefe* (macoll@isdefe.es)

Resumen

Este capítulo presenta las nuevas capacidades habilitadas por la Ingeniería de Sistemas Basada en Modelos (MBSE¹, por sus siglas en inglés) y la ingeniería digital para apoyar la entrada en servicio, las operaciones, el sostenimiento² y la baja de los sistemas durante la Fase de Servicio³. Los temas a tratar incluyen avances en la aplicación de las tecnologías de gemelos digitales y de hilos digitales⁴ para apoyar el sostenimiento de los sistemas a lo largo del ciclo de vida, así como algunas consideraciones para llevar a cabo la transformación digital y la adopción de tecnologías avanzadas basadas en modelos. El capítulo discute la importancia de planificar el uso de estas tecnologías desde las primeras etapas del ciclo de vida de los sistemas para establecer una base y una estrategia conducente a la aplicación de metodologías basadas en modelos, la automatización, la realidad virtual, la inteligencia artificial y otras tecnologías avanzadas.

Palabras clave

Entrada en Servicio, Operación, Sostenimiento, Baja, Gemelo Digital, Hilo Digital, Automatización, Apoyo Logístico Basado en Modelos (MBPS), Mantenimiento Basado en la Condición (CBM), Mantenimiento Predictivo (PdM), Mantenimiento Prescriptivo (RxM), Tiempo Restante de Vida Útil (RUL), Overhaul, Gestión de la Obsolescencia, Gestión de la Configuración, Transformación Digital.



1. INTRODUCCIÓN

El Capítulo 5 ha tratado el uso de la tecnología de ingeniería digital durante las primeras fases del desarrollo del ciclo de vida de los sistemas. Este capítulo profundizará en cómo el origen y la base digital de los sistemas, y las inversiones realizadas en tecnologías de ingeniería digital que permiten el desarrollo de los gemelos digitales y los hilos digitales, pueden ser utilizados a lo largo de las etapas posteriores del ciclo de vida de los sistemas. El capítulo analiza también algunas consideraciones al tratar de implementar esta tecnología, abordando oportunidades y riesgos a lo largo de la preparación para la entrada en servicio, las operaciones, el sostenimiento, y la baja del sistema, incluyendo su apoyo logístico.

El interés en las tecnologías relacionadas con la ingeniería digital, especialmente aquellas que posibilitan los gemelos digitales y los hilos digitales, ha aumentado en los últimos años en diversas industrias, especialmente en la fabricación de bienes de equipo y en el sector de la Defensa. Tres conceptos principales están en el núcleo de este interés: la ingeniería digital, el gemelo digital y el hilo digital.

La **ingeniería digital** es una disciplina que aprovecha tecnologías avanzadas, como el Diseño Asistido por Computadora (CAD⁵, por sus siglas en inglés), la simulación y el análisis de datos, para apoyar el diseño, desarrollo y gestión de sistemas y productos complejos a lo largo de su ciclo de vida. Implica la creación e integración de modelos digitales, simulaciones y toma de decisiones basadas en datos para optimizar los procesos de diseño, fabricación, entrada en servicio, operación, sostenimiento y baja del sistema, incluyendo su apoyo logístico.

Un **gemelo digital** es, esencialmente, una representación virtual de un objeto, sistema o proceso físico (el gemelo físico) que permite la monitorización, el análisis y la simulación en tiempo real. El concepto de gemelo digital permite a las organizaciones tener un entendimiento más profundo de sus activos, las operaciones y su rendimiento, al conectar los mundos físico y digital a través del hilo digital.

Sin embargo, se ha abusado significativamente del término gemelo digital. Si bien todo gemelo digital es un modelo digital, no todos los modelos digitales necesariamente constituyen un gemelo digital. Idealmente, se podría sustituir un gemelo digital por su gemelo físico y viceversa, sin notar el cambio. Esta dualidad resulta muy valiosa porque (1) se puede experimentar con el gemelo digital, tanto en la fase de desarrollo como en las pruebas operativas, de una manera más económica sin la necesidad de arriesgar al gemelo físico (el sistema real); y (2) un modelo digital permite acelerar el descubrimiento de propiedades del sistema, ya que la simulación de software a menudo sucede más rápido que las pruebas físicas.

La Figura 1 ilustra que existen muchos tipos de gemelos digitales (gemelo virtual del producto, gemelo virtual del proceso, gemelo digital de fabricación, y gemelo digital), dependiendo de la fase del ciclo de vida en la que se encuentre el producto.

El concepto de **hilo digital** hace referencia a los medios utilizados para conectar el flujo de datos e información a lo largo de todo el ciclo de vida de un sistema o producto. Permite una comunicación y una colaboración efectivas entre diversas partes interesadas en todos los aspectos y etapas del ciclo de vida de la ingeniería, para garantizar la coherencia y la trazabilidad en todas las actividades.

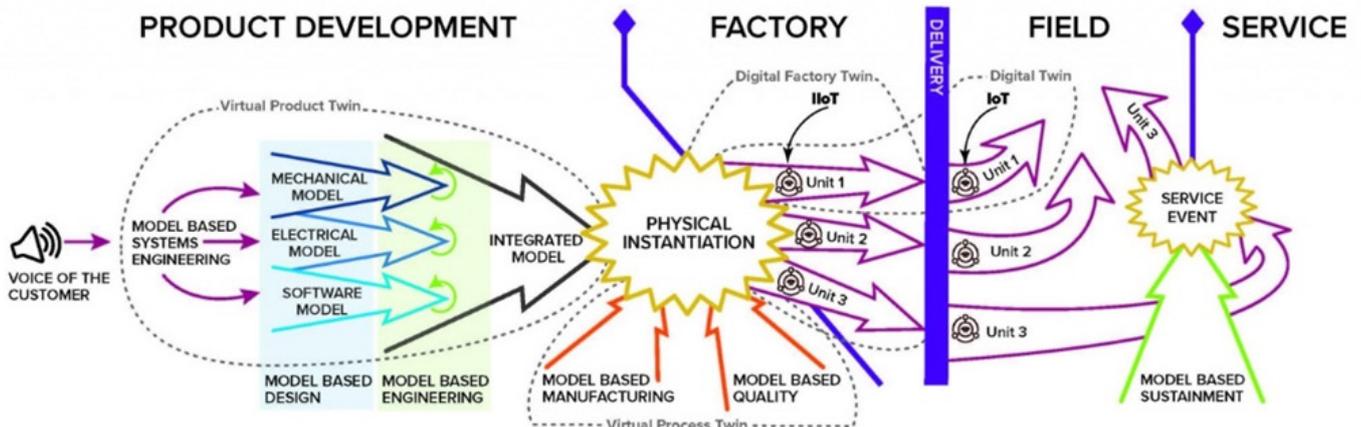


Figura 1. Gemelos digitales a lo largo del ciclo de vida de un sistema

Una definición más formal de **hilo digital**, proporcionada por la Defense Acquisition University (DAU), es la siguiente⁶:

“Un marco analítico extensible y configurable que agiliza de manera fluida la interacción controlada de datos técnicos, software, información y conocimiento en el ecosistema de la ingeniería digital, basado en los requisitos, las arquitecturas, los formatos y las reglas establecidas para construir modelos digitales. Se utiliza para informar a los tomadores de decisiones a lo largo del ciclo de vida de un sistema, proporcionándoles la capacidad de acceder, integrar y transformar datos en información útil para tomar acciones concretas.”

El proceso de transformación digital de una organización en particular es único, y puede hacer posible una mayor innovación, un ahorro de costes y una mejora en la fiabilidad en los productos y sistemas que diseña o produce. Sin embargo, la tecnología y las aplicaciones descritas en este capítulo pueden no ser apropiadas (o necesarias) para todas las organizaciones o todos los Sistemas de Interés (Sol⁷, por sus siglas en inglés). Además, muchos de los resultados esperados pueden no ser alcanzables si la organización no está preparada para aprovechar la tecnología en ciertos contextos o si los modelos de datos no son accesibles por la organización operadora del sistema. Además, cabe señalar que, al igual que ocurre en la aplicación de la ingeniería de sistemas físicos, la participación directa de los usuarios, los operadores, los mantenedores y otras partes interesadas es necesaria para el desarrollo de los gemelos digitales. No obstante, aunque las necesidades del diseño, desarrollo y fabricación del sistema para con el gemelo digital y el hilo digital pueden ser diferentes de las necesidades del operador durante el ciclo de vida, los casos de uso para su uso en etapas posteriores del ciclo de vida deben considerarse tempranamente en el desarrollo del sistema.

2. APLICACIÓN DE TECNOLOGÍAS DE GEMELOS DIGITALES, ENTORNOS VIRTUALES E HILOS DIGITALES EN LA FASE DE SERVICIO

2.1. Transformando digitalmente la Entrada en Servicio

En general, el uso de las tecnologías relacionadas con la ingeniería digital ha agilizado la entrada en servicio de los sistemas, reducido los costes, aumentado la agilidad y mejorado la calidad general de los sistemas puestos en servicio [1]. A continuación, se presenta una descripción y ejemplos de algunas capacidades habilitadas por la ingeniería digital y sus aplicaciones durante la entrada en servicio.

2.1.1. Entradas en servicio simplificadas mediante la automatización, la virtualización y las pruebas en bucle cerrado del sistema (hardware in the loop testing⁸)

El uso de la tecnología de virtualización permite una mayor predictibilidad en la entrada en servicio de los sistemas. La virtualización implica crear versiones virtuales de cualquier hardware, software, almacenamiento o recursos de red, que pueden aprovechar la emulación para imitar su comportamiento. De esta manera los sistemas pueden ser puestos en servicio, primeramente, en un entorno virtual, reduciendo así la necesidad de infraestructura física excesiva. Esto permite una validación y una entrada en servicio más rápidos y eficientes, ya que los sistemas virtuales pueden ser replicados rápidamente y escalados según sea necesario [2].

Posteriormente, cuando es el sistema real el que está preparándose para su entrada en servicio, las tecnologías digitales permiten la automatización de tareas propias de la entrada en servicio, como son la instalación, la configuración y las pruebas del software. La automatización disminuye las posibilidades de error humano y permite entradas en servicio más consistentes y repetibles. Además, al facilitar la adopción de prácticas de DevOps⁹ (que implican una integración y una entrada en servicio continuas), algunos cambios en el sistema pueden ejecutarse, probarse y ponerse en servicio automáticamente en los entornos de producción, asegurando un proceso de entrada en servicio más rápido y fiable.

2.1.2. Control de Calidad

Los gemelos digitales y los entornos virtuales incorporan información geométrica detallada que, junto con los metadatos asociados, ayuda a garantizar el cumplimiento de las especificaciones y estándares de diseño, permitiendo un mejor control de la calidad durante la fabricación y el montaje de los sistemas. La información geométrica detallada proporciona la base para desarrollar las simulaciones, el análisis y las visualizaciones precisas. Permite una comprensión integral de la forma, la estructura y las relaciones espaciales de la entidad física dentro de un entorno virtual, incluyendo información adicional como pueden ser la representación tridimensional, la precisión dimensional, o las propiedades y texturas de los materiales, entre otros. Los metadatos asociados a los gemelos digitales y entornos virtuales pueden aprovecharse para mejorar la comprensión, la gestión y la utilización de éstos, ya que contienen información descriptiva y propiedades que los caracterizan, y proporcionan contexto a la representación digital del objeto físico o sistema, como pueden ser la información relacionada con el ciclo de vida o la identificación precisa de elementos (por ejemplo, para el control de la configuración).



Figura 2. Ejemplo de Gemelo Digital (link: <https://youtu.be/G26mx4TnKyM?si=gR-fqlhmZw0B21-y>)

2.1.3. Adiestramiento y familiarización del sistema

Los entornos digitales generados a través de la Realidad Aumentada (AR¹⁰, por sus siglas en inglés) o la Realidad Virtual (VR¹¹, por sus siglas en inglés) ofrecen experiencias de aprendizaje inmersivas que pueden mejorar considerablemente el adiestramiento y la familiarización con los sistemas, permitiendo a las dotaciones y a los técnicos de mantenimiento comprender y gestionar mejor los sistemas complejos en un entorno realista y con menor dependencia de estar cerca del activo físico (por ejemplo, véase la Figura 2). Estos sistemas virtuales pueden proporcionar a los técnicos asistencia en tiempo real para el mantenimiento, utilizando superposiciones de AR para comunicar instrucciones visuales y manuales de referencia directamente sobre el sistema físico, lo que resulta en una mayor eficiencia y precisión durante las actividades de mantenimiento.

Un ejemplo de este tipo de visualización del modelo del producto es la maqueta digital¹², que resulta valiosa al vincular la información de diseño y de apoyo logístico con la información geométrica incorporada en los gemelos digitales. Esta conexión, establecida mediante la integración de datos técnicos y logísticos, crea un entorno de conciencia espacial que mejora la comprensión de las tareas de mantenimiento [3]. La visualización del modelo del producto es un concepto

que implica crear y presentar representaciones visuales de objetos físicos o sistemas, típicamente en formato digital. El objetivo principal es transmitir información sobre el diseño, la estructura y el comportamiento de un producto a través de medios gráficos e interactivos. En el contexto del adiestramiento y la familiarización de sistemas de defensa, por ejemplo, la visualización del modelo del producto desempeña un papel crucial al proporcionar una experiencia de aprendizaje inmersiva y efectiva en un entorno virtual. Por ejemplo, cuando se combina con el sistema real, un técnico de mantenimiento de aeronaves puede apoyarse en objetos digitales proyectados y superpuestos directamente sobre el entorno físico, en lugar de tener que depender de memorizar diagramas en 2D y navegar por áreas oscuras y confinadas para ubicar partes específicas [4]. Los activos digitales utilizados de esta manera no solo mejoran la efectividad y eficiencia del adiestramiento, sino que también mejoran los resultados de la ejecución real de las tareas.

A pesar de su potencial, existen desafíos para la adopción de estas tecnologías incluyendo consideraciones de tipo ergonómico y de disponibilidad de soluciones de RV/RA asequibles [5]. Como alternativa intermedia, se pueden proporcionar artefactos digitales al operador o mantenedor en forma de Publicaciones Técnicas Electrónicas Interactivas (IETP¹³, por sus siglas en inglés) o Manuales Técnicos Electrónicos Interactivos (IETM¹⁴, por sus siglas en inglés). Ambos son formatos digitales de documentos técnicos

que proporcionan información completa, instrucciones y orientación para la operación, el mantenimiento, reparación y solución de problemas en sistemas o equipos complejos [6]. Dichas publicaciones están diseñadas para reemplazar o complementar manuales tradicionales en papel, mediante versiones electrónicas que ofrecen elementos interactivos y multimedia para mejorar la experiencia del usuario y facilitar el aprendizaje y la comprensión del sistema de forma eficiente. En comparación con los manuales impresos, los IETMs están cada vez más extendidos porque son interactivos y fáciles de usar y actualizar, además de proporcionar información sobre el sistema que de otra manera sería difícil de transmitir.

2.2. Transformando digitalmente la vida operativa

2.2.1. Apoyo logístico basado en modelos y la gestión de la configuración

El Apoyo Logístico Basado en Modelos (MBPS¹⁵, por sus siglas en inglés) consiste en la aplicación de los principios y conceptos de la ingeniería digital al Apoyo Logístico Integrado (ALI). En este sentido, se refiere a la modelización de cualquier aspecto relacionado con el ALI, incluyendo las características de soportabilidad, fiabilidad y seguridad (safety), tanto en términos de diseño como en los procesos del ciclo de vida. El MBPS aprovecha los modelos digitales, las simulaciones, la interoperabilidad de los sistemas de información y los datos, para mejorar la eficiencia y efectividad de los sistemas de apoyo y para mejorar la preparación operativa y optimizar el sostenimiento y la gestión del ciclo de vida durante toda su vida útil. En particular, los modelos generados durante el desarrollo del sistema se utilizan durante la Fase de Servicio para ayudar a comprender la funcionalidad del sistema y sus interfaces externas e internas.

La continuidad de los datos, que es esencial para apoyar una transición coherente entre los modelos generados o utilizados en el desarrollo del sistema y aquellos generados o utilizados para el MBPS, se habilita mediante el hilo digital. Por otro lado, mantener una línea base de Configuración precisa de un activo o sistema es un aspecto crítico para garantizar que pueda recibir un adecuado apoyo logístico. Mediante la aplicación del MBPS, al utilizarse “fuentes autorizadas de verdad”¹⁶ ya no es necesario mantener manualmente dos configuraciones diferentes, una para el diseño y otra para el apoyo logístico, a lo largo del ciclo de vida, al incorporar sobre los propios elementos configurados del diseño toda la información logística relevante.

2.2.2. Estrategias de mantenimiento proactivas habilitadas por los Gemelos Digitales y los Hilos Digitales

Tradicionalmente, tanto el mantenimiento correctivo como el mantenimiento preventivo han sido las estrategias de mantenimiento más ampliamente utilizadas. En el mantenimiento correctivo, las reparaciones se llevan a cabo después de que haya ocurrido la avería. Por el contrario, el mantenimiento preventivo se centra en prevenir posibles fallos antes de que éstos ocurran, mediante la ejecución de acciones de mantenimiento programadas, como inspecciones, mantenimiento rutinario o reemplazos de componentes (normalmente basados en patrones de uso: horas de funcionamiento, ciclos, número de arranques, etc.). Sin embargo, los avances en los sensores y en las tecnologías de monitorización y el desarrollo resultante de los gemelos digitales e hilos digitales permiten un giro hacia enfoques de mantenimiento basados en el uso de los datos. Estos enfoques utilizan algoritmos para hacer predicciones y recomendaciones con el fin de optimizar la fiabilidad del sistema y la eficacia operativa de los activos.

Al pasar de un mantenimiento reactivo a uno basado en el uso de los datos, las organizaciones pueden minimizar el tiempo de inactividad de los activos y reducir los costes de mantenimiento, ya que los posibles modos de fallo pueden detectarse y abordarse antes de que éstos ocurran, previniendo interrupciones en la operación y las reparaciones de emergencia. Se pueden distinguir, principalmente, tres tipos de mantenimiento proactivo:

- **Mantenimiento Basado en la Condición** (CBM¹⁷, por sus siglas en inglés). El CBM se centra en monitorizar la condición en tiempo real del equipo mediante el uso de varios sensores y herramientas de recopilación de datos para evaluar el estado real del equipo. El CBM utiliza umbrales o condiciones preestablecidas para activar alarmas o acciones de mantenimiento cuando se detectan desviaciones de las condiciones de operación normales.

El CBM típicamente se centra en un conjunto específico de parámetros relevantes para el equipo que se está monitorizando y en la vigilancia continua de algunos factores críticos que impactan directamente en el rendimiento del mismo, o en su fiabilidad inmediata. El objetivo principal del CBM es monitorizar la condición real del equipo y tomar acciones oportunas para prevenir fallos o problemas inminentes.

El CBM se utiliza en una amplia variedad de aplicaciones, incluyendo la fabricación de bienes de equipo, la defensa, los sistemas de control industrial o la atención médica para diagnosticar modos de fallo inminentes, reduciendo así los costes de mantenimiento, mejorando la fiabilidad,

la disponibilidad y la seguridad (safety), extendiendo el tiempo entre revisiones (comparado con el mantenimiento preventivo tradicional) y reduciendo el tiempo de inactividad innecesario. Con la capacidad de monitorizar continuamente la salud del activo y las métricas de su rendimiento, se pueden recopilar datos del sistema para desarrollar modelos que permitan la detección proactiva de modos de fallo.

- **Mantenimiento Predictivo (PdM).** El PdM también implica monitorizar la condición del equipo, pero se enfoca principalmente en utilizar análisis de datos en tiempo real y datos históricos para predecir cuándo se necesita realizar una acción de mantenimiento. El PdM analiza patrones y tendencias en los datos para prever posibles fallos o degradaciones en el rendimiento del equipo.

Aunque el CBM y el PdM comparten similitudes, y a menudo utilizan tecnologías similares, hay algunas distinciones entre ellos: el CBM inicia acciones de mantenimiento basadas en la condición en tiempo real del equipo y se centra en un conjunto específico de parámetros, mientras que el PdM tiene como objetivo programar actividades de mantenimiento justo antes de que el equipo probablemente falle, optimizando el momento del mantenimiento para prevenir fallos y minimizar el tiempo de inactividad.

El PdM a menudo implica un análisis de datos más completo que el CBM, utilizando una gama más amplia de datos e indicadores para predecir fallos. Puede incorporar análisis avanzados, algoritmos de inteligencia artificial, como el aprendizaje automático, y otros algoritmos sofisticados para predecir los fallos, mientras que el CBM los utiliza de manera menos intensiva.

- **Mantenimiento Prescriptivo (RxM).** El RxM va más allá del PdM. No solo predice posibles fallos en los equipos, sino que también prescribe acciones específicas para prevenir esos fallos o mitigar su impacto. Este enfoque integra análisis predictivo con sistemas automatizados de toma de decisiones para proporcionar recomendaciones (o prescripciones) precisas para realizar acciones de mantenimiento. Estas recomendaciones pueden incluir instrucciones detalladas sobre tareas de mantenimiento, reparaciones, ajustes o cambios en el régimen operativo del sistema. El RxM utiliza algoritmos sofisticados de inteligencia artificial y técnicas de “Big Data” para analizar extensos conjuntos de datos. Examina datos históricos y en tiempo real para predecir posibles fallos y determinar el mejor curso de acción para evitar o abordar estos problemas. En el contexto del RxM, la prognosis juega un papel significativo en la determinación de las acciones apropiadas que se recomendarán. La prognosis se refiere a la estimación o predicción de condiciones o eventos futuros basados en el análisis de datos actuales e históricos. En el RxM, la prognosis implica la predicción de posibles fallos, la estimación de la vida útil restante (RUL¹⁹, por sus siglas en inglés) y la predicción de la salud y el rendimiento futuros de activos o equipos [7].

Los gemelos digitales tendrán un rol central en los enfoques de mantenimiento proactivos gracias a sus capacidades para habilitar aplicaciones basadas en el uso de los datos: monitorización y análisis en tiempo real, análisis predictivo y prescriptivo, simulación de escenarios, monitorización de la salud y la prognosis y análisis de optimización. En el mantenimiento predictivo, los gemelos digitales recopilan continuamente los datos en tiempo real de los sensores incorporados en los activos físicos y utilizan los datos históricos para analizar tendencias, patrones y anomalías. Al aprovechar esta información, los gemelos digitales facilitan el análisis predictivo de posibles fallos o la degradación del rendimiento. En cuanto al mantenimiento prescriptivo, los gemelos digitales desempeñan un papel crucial, al proporcionar no solo información predictiva, sino también recomendaciones de acciones concretas. Al simular diferentes escenarios y analizar los datos, los gemelos digitales pueden sugerir acciones de mantenimiento específicas o estrategias para prevenir los fallos u optimizar el rendimiento del activo. También ayudan a prescribir las acciones más efectivas que se deben tomar en función del análisis predictivo y las simulaciones.

Finalmente, cabe señalar que los gemelos digitales también pueden ser valiosos para apoyar el mantenimiento preventivo, ya que pueden ayudar a establecer las líneas base del estado de los activos, monitorizar continuamente las condiciones de los mismos e identificar desviaciones de los parámetros operativos normales mediante el seguimiento de la salud del equipo.

2.2.3. Modernizando el Mantenimiento, las Reparaciones y los Overhaul¹⁹ (MRO)

A medida que los activos envejecen puede surgir la necesidad de reemplazar ciertos componentes, implementar rediseños o actualizaciones en el sistema. La información geométrica integrada en los gemelos digitales, junto con los metadatos sobre materiales y las tolerancias de fabricación, ayuda a identificar piezas de repuesto compatibles y a planificar dichas actualizaciones. Esto puede ayudar a anticipar la integración de nuevos componentes en el sistema existente, evitando problemas de compatibilidad y posibles interrupciones.

La integración del CAD permite una colaboración efectiva entre las partes interesadas, incorporando las modificaciones realizadas durante el ciclo de vida de un activo en el modelo CAD para futuras referencias. Los modelos CAD creados durante las fases de diseño se convierten en la base para proporcionar una representación visual del activo físico en el gemelo digital.

Esta representación digital, con los datos geométricos precisos, permite posteriormente a los equipos de mantenimiento visualizar y comprender los componentes y ensamblajes del activo, ayudando en la resolución de problemas y a la planificación de las actividades de mantenimiento, dando soporte además a la comunicación de un posible cambio en la configuración del activo desplegado.

Junto con la representación geométrica, los gemelos digitales incorporan metadatos esenciales, como las propiedades de los materiales, las tolerancias de fabricación y los números de serie de las piezas. Estos metadatos proporcionan una información crucial para las actividades de fabricación, ensamblaje y mantenimiento futuros. En particular, ayudan al personal de mantenimiento a identificar rápidamente los componentes correctos, acceder a la información relevante y a realizar reparaciones de manera más eficiente, reduciendo el tiempo de inactividad del sistema. Por ejemplo, el personal de mantenimiento puede acceder al gemelo virtual para identificar los componentes exactos, comprender sus dimensiones y revisar registros de mantenimiento históricos. Esto permite actividades de mantenimiento más eficientes y enfocadas, minimizando el tiempo de inactividad y extendiendo la vida útil del activo. De manera similar, pueden utilizar el gemelo virtual para comprender el diseño físico del sistema, acceder a metadatos para la identificación de piezas y solucionar problemas sin necesidad de tener acceso directo al activo físico.

2.2.4. Gestión de la obsolescencia

La gestión de la obsolescencia es fundamental para garantizar continuamente la disponibilidad operativa, la soportabilidad y efectividad de ciertos sistemas a lo largo de dilatadas vidas operativas, que a menudo pueden abarcar varias décadas. Una gestión efectiva de la obsolescencia requiere un enfoque proactivo y holístico, integrando diversas estrategias a lo largo del ciclo de vida de los sistemas para asegurar la preparación operativa, reducir riesgos y gestionar los costes asociados al sostenimiento de estos sistemas con el tiempo. El uso de modelos digitales combinados con algunas tecnologías emergentes de fabricación proporciona nuevas herramientas para la gestión de la obsolescencia.

La información geométrica incorporada en los gemelos digitales facilita la ingeniería inversa de determinadas piezas y la gestión de su obsolescencia, ya que son réplicas exactas de sus gemelos físicos, posibilitando su fabricación. La integración de sistemas CAD y CAM²⁰ (Fabricación Asistida por Computadora) agiliza los procesos de diseño y fabricación de los productos, lo cual es especialmente útil para la modernización y el manejo de partes obsoletas. Esta conexión puede ser aún más reforzada por procesos de fabricación novedosos, como la fabricación aditiva (también conocida como impresión 3D), ya que la conexión entre diseño y fabricación puede establecerse completamente como un hilo digital.

2.3. Transformando digitalmente la baja del sistema

En comparación con otras fases del ciclo de vida, la baja de los sistemas es la menos desarrollada en términos de transformación digital. Para algunas situaciones, puede ser útil preservar los datos de uso de los sistemas y componentes que han alcanzado el final de su vida útil, como pueden ser datos de rendimiento, ya que el mismo componente en el sistema dado de baja puede continuar siendo utilizado en otras aplicaciones similares. Por ejemplo, si se retira en un avión un determinado actuador, éste puede tener una vida útil remanente para seguir utilizándose en otros aviones. Los datos de rendimiento también pueden ser utilizados para futuras simulaciones que influyan en futuros rediseños. El hilo digital también puede ayudar a identificar y tratar mejor los componentes con consideraciones especiales para su eliminación, como pueden ser los materiales peligrosos o el hardware y el software con clasificación de seguridad.

Adicionalmente, para una organización determinada la Etapa de Baja de un sistema dado podría ser, simultáneamente, el inicio de un Programa de Adquisición para otra organización (por ejemplo, cuando se adquiere un avión de segunda mano de otra organización). En este caso, aprovechar el gemelo digital y el hilo digital desarrollado inicialmente con el sistema puede ser un desafío para la organización que recibe los activos, dado que su proceso de transformación digital puede no estar suficientemente maduro.

Al aprovechar los gemelos digitales y los hilos digitales en la Etapa de Baja, las organizaciones pueden mejorar la sostenibilidad, el cumplimiento normativo y la eficiencia en el proceso de desmilitarización. Estas tecnologías contribuyen a implementar prácticas responsables y respetuosas con el medio ambiente, al tiempo que proporcionan valiosos conocimientos para la mejora continua en los procesos de desarrollo y gestión del ciclo de vida.

2.4. Ejemplos de nuevas capacidades que permiten transformar digitalmente el Ciclo de Vida de un Sistema

2.4.1. Desarrollo colaborativo de modelos habilitados por los servicios en la nube

Las herramientas de modelado en entornos colaborativos facilitan las relaciones interfuncionales, ya que permiten integrar suites de CAD en el software de Gestión del Ciclo de Vida del Producto (PLM²¹, por sus siglas en inglés) para permitir que ingenieros, diseñadores y otras partes interesadas trabajen conjuntamente de manera fluida. Los cambios realizados en una herramienta pueden comunicarse a las otras, fomentando la comunicación y alineación entre diferentes equipos.

El uso de la tecnología de desarrollo de modelos colaborativos apoya la formalización de la planificación, desarrollo, integración, curación²² y uso de los modelos para las actividades de ingeniería y toma de decisiones a lo largo del ciclo de vida. A diferencia de los enfoques basados en documentos formales, desarrollados bajo marcos de ingeniería de sistemas “clásicos” heredados, las organizaciones centradas en modelos aprovechan los entornos colaborativos, donde los equipos pueden definir y planificar la creación de modelos para apoyar las actividades de ingeniería, garantizando un enfoque estructurado y a la vez auditable. Esta tecnología facilita el desarrollo formal de modelos, al proporcionar herramientas que incorporan diversas técnicas y algoritmos, integran datos de diferentes fuentes y curan modelos refinándolos y optimizándolos con el tiempo. Se pueden lograr ganancias significativas en la productividad poniendo el énfasis en la definición y evolución de los sistemas, basada en modelos, en comparación con los enfoques tradicionales de desarrollo, basados en documentos.

2.4.2. Agregación de datos y sistemas integrados de información

Para la mayoría de las organizaciones, todavía hay una carga de trabajo significativa asociada al uso de procesos manuales y a la existencia de sistemas de información dispares, utilizados para recopilar y utilizar los datos. Un sistema de información moderno, habilitado por los hilos digitales, puede conectar varias disciplinas y etapas del ciclo de vida de un activo. Los modelos virtuales de productos y procesos, representados por sus gemelos digitales e hilos digitales, pueden permitir el análisis, la optimización del rendimiento, la toma de decisiones y las operaciones mediante la integración de los datos e información para los usuarios finales.

La tecnología de gemelos digitales permite representaciones virtuales en tiempo real (o casi en tiempo real) del estado actual de los activos, en las cuales se pueden integrar bases de datos, repositorios de información y modelos de sistemas de múltiples disciplinas. Un hilo digital, con una arquitectura bien definida, puede proporcionar una integración y una accesibilidad sin precedentes a los datos relevantes en todo el ciclo de vida del activo, desde el diseño hasta la operación y el mantenimiento para aumentar las funcionalidades en el gemelo digital.

2.4.3. Sensores e Internet de las Cosas (IOT)

La aparición de nuevas capacidades en la tecnología de sensores y la implementación del Internet de las cosas (IoT²³, por sus siglas en inglés) han revolucionado la forma en que se desarrollan los gemelos digitales, ofreciendo una gran integración de los dominios físico y digital. Paralelamente, avances en microfabricación y nanotecnología han dado lugar a la creación de sensores altamente sensibles, compactos y eficientes energéticamente, capaces de detectar una amplia gama de fenómenos físicos, necesarios para desarrollar gemelos digitales cada vez más realistas.

Al mismo tiempo, las innovaciones en la tecnología IoT, incluyendo alternativas de conectividad mejoradas, un procesamiento de datos más robusto y la computación en la nube, facilitan la transmisión y el análisis fiable de las grandes cantidades de datos recopilados por estos sensores. Esta infraestructura IoT permite la sincronización en tiempo real de los activos físicos con sus contrapartes digitales, creando modelos virtuales dinámicos que reflejan con precisión el mundo físico.

2.4.4. Computación en la nube

La aparición de la computación en la nube ha revolucionado la puesta en servicio²⁴ de los sistemas. Las plataformas en la nube proporcionan acceso bajo demanda a recursos, lo que permite a las organizaciones escalar fácilmente sus sistemas y ponerlos en servicio a nivel global. Los modelos de puesta en servicio basados en la nube, como son las Infraestructuras como Servicio (IaaS²⁵, por sus siglas en inglés) y las Plataformas como Servicio (PaaS²⁶, por sus siglas en inglés), simplifican este proceso para las organizaciones, al abstraer las preocupaciones de la infraestructura subyacente. Específicamente, se reduce la necesidad de invertir en servidores físicos y centros de datos.

La IaaS es una categoría de servicios de computación en la nube que proporciona recursos informáticos virtualizados a través de internet. En un modelo IaaS, los usuarios pueden alquilar o arrendar varios componentes de infraestructura, como máquinas virtuales, almacenamiento y redes, en lugar de invertir y mantener su propio hardware físico.

La PaaS es un modelo de servicio de computación en la nube que proporciona una plataforma que permite a los clientes desarrollar, ejecutar y administrar aplicaciones de software sin la complejidad de construir y mantener la infraestructura subyacente. En un modelo PaaS, el proveedor de la nube ofrece una plataforma completa e integrada que incluye herramientas de desarrollo, entornos de ejecución y otros servicios necesarios para construir, poner en servicio y escalar las aplicaciones.

3. CONSIDERACIONES PARA ALCANZAR LA TRANSFORMACIÓN DIGITAL A LO LARGO DEL CICLO DE VIDA

3.1. Visión general

La transformación digital a lo largo del ciclo de vida exige un enfoque integral en la estrategia de gestión de los datos técnicos. Esta estrategia debe guiar la adquisición, la gestión y el mantenimiento de los datos técnicos y del software necesario para apoyar a un sistema desde su concepción hasta su baja. Ciertas consideraciones, como la protección de la propiedad intelectual o fomentar las competencias digitales de los usuarios de los datos, deben ser cuestiones centrales en dicha estrategia. Al asegurar los datos y los derechos necesarios, las organizaciones pueden mejorar la comprensión del diseño del sistema, optimizar las operaciones en diversos entornos y posibilitar una mayor eficiencia en el coste de adquisición y mantenimiento.

La eficacia de la tecnología de los modelos digitales y de los gemelos digitales está profundamente vinculada a su accesibilidad y usabilidad de éstos por parte de los usuarios operativos y de mantenimiento. Éstos dependen de herramientas intuitivas que permitan la simulación realista de los gemelos digitales, junto con un acceso fluido a los datos logísticos y técnicos completos. Estos datos deben ser abiertos, interoperables, adaptables y accesibles en entornos de computación tanto en la nube como “en el borde”²⁷, lo que requiere un esfuerzo colaborativo que integre las percepciones tanto de los equipos técnicos como de los usuarios operativos desde la fase de diseño.

Idealmente, un único gemelo digital debería ser suficiente para dar servicio a todas las fases del ciclo de vida de un sistema, incorporando toda la información y los modelos de comportamientos necesarios. Sin

embargo, incorporar de manera eficaz en el gemelo digital las capacidades operativas y de mantenimiento requiere prestar atención sobre ellas durante las etapas de concepto, desarrollo y producción del sistema. La transición hacia un gemelo digital unificado exige una integración gradual de los modelos operativos y de sostenimiento. Hasta que se logre una integración completa, las distinciones entre los gemelos digitales de diseño, operativos y de mantenimiento seguirán siendo necesarias. Sin una implementación cuidadosa, existe el riesgo de crear un gemelo digital que funcione bien para el diseño y la producción, pero que se quede corto en operaciones y mantenimiento debido a la incapacidad de replicar con precisión los comportamientos de su gemelo físico. Por lo tanto, un verdadero gemelo digital debe incorporar efectivamente tanto las funcionalidades de diseño como las operativas y de sostenimiento.

El desarrollo de un gemelo digital holístico, equipado con las capacidades, funcionalidades y casos de uso necesarios, puede considerarse como un proyecto de desarrollo de software independiente. Dicho proyecto debería avanzar en paralelo con el desarrollo del Sistema de Interés (Sol), resaltando la necesidad de aplicar un enfoque multidisciplinario. Lograr los objetivos de la transformación digital requiere una visión integral que abarque a las personas, los procesos, la tecnología, los datos y los objetivos estratégicos, todo ello manteniendo el foco en establecer un hilo digital duradero que interconecte todos los elementos durante el ciclo de vida.

En las siguientes secciones, se proporcionan algunas orientaciones para implementar o adoptar la ingeniería digital en la Fase de Servicio del ciclo de vida del sistema.

3.2. Ante la duda, empezar con un proyecto piloto

Se requiere un trabajo inicial significativo en la preparación para el uso posterior de un gemelo digital y el hilo digital, incluso en las fases de concepto y desarrollo del sistema. Puede ser un reto decidir por dónde empezar, qué casos de uso priorizar y qué recursos serán necesarios para madurar dicha capacidad.

Para muchas organizaciones, los esfuerzos de transformación digital comienzan con proyectos piloto, seguido por lo que típicamente es una transición lenta y llena de retos de pasar de los proyectos piloto exitosos a las operaciones a escala [8]. Una transformación digital exitosa a escala requiere una planificación y una coordinación cuidadosas para implementar de manera integral las nuevas tecnologías en el contexto de la organización. Dicha transformación digital debe estar alineada con los objetivos estratégicos y el compromiso de la dirección para evaluar críticamente los procesos, los marcos de trabajo y las arquitecturas heredadas, y realizar inversiones apropiadas para impulsar los objetivos de transformación digital. También implica el compromiso de la dirección para capacitar a los equipos innovadores y fomentar una cultura abierta a la experimentación y capaz de aprender de los errores.

ORGANIZACIÓN DESARROLLADORA DEL ESTÁNDAR	ALCANCE
ISO/TC 184	Establece estándares de datos industriales en diferentes ámbitos, incluyendo la fabricación, la automatización industrial y los sistemas de información para garantizar la compatibilidad e interoperabilidad de los gemelos digitales en el campo de las llamadas “fábricas inteligentes” (Smart Factories)
IEEE P3144 Digital Twin Working Group	El Estándar para el Modelo de Madurez y Metodología de Evaluación de Gemelos Digitales en la Industria (en inglés “Standard for Digital Twin Maturity Model and Assessment Methodology in Industry”) define un modelo de madurez de los gemelos digitales para la industria que incluye dominios de capacidades de gemelos digitales y los subdominios correspondientes. Este estándar también define las metodologías de evaluación, incluyendo su contenido, los procedimientos y los niveles de madurez
The 3rd Generation Partnership Project (3GPP)	Desarrolla estándares para las redes 5G, las cuales ofrecen las capacidades de comunicación de alta velocidad y fiabilidad requeridas para los gemelos digitales
The Open Geospatial Consortium (OGC)	Administra estándares de información geoespacial, los cuales son cruciales para los gemelos digitales en ciudades inteligentes y otros ámbitos
IEC TC65	Se enfoca en la interoperabilidad de los estándares, específicamente en el contexto de las fábricas inteligentes. Sus esfuerzos contribuyen a armonizar la comunicación y el intercambio de datos entre varios componentes de gemelos digitales en el ámbito de la fabricación de bienes de equipo
oneM2M	Una iniciativa global que estandariza plataformas de IoT de capa de servicio, proporcionando funciones de servicio comunes que son esenciales para el funcionamiento efectivo de los gemelos digitales

Tabla 1. Estandarización de la tecnología de Gemelos Digitales

3.3. Ser consciente de los derechos acerca de los datos técnicos

Las organizaciones deben planificar anticipadamente los derechos acerca de los datos técnicos que necesitan adquirirse si el sistema es diseñado por una entidad externa. Si este trabajo no se realiza por adelantado muchas oportunidades de transformación digital pueden no ser alcanzables, debido a la existencia de formatos propietarios o incompatibles, datos inaccesibles o costes de retrabajo excesivos para que el sistema produzca los datos deseados. En todos los proyectos se deben anticipar los datos técnicos necesarios para realizar las actividades durante el ciclo de vida y establecer los derechos u opciones para comprar esos datos en los contratos y acuerdos de nivel de servicio [9], así como consultar anticipadamente con expertos sobre los estándares de datos y la tecnología más apropiados para incorporar en las especificaciones del contrato.

3.4. Aprovechar la estandarización

Es esencial adoptar estándares de datos semánticamente ricos, abiertos y accesibles, que faciliten la interoperabilidad, la vinculación de los datos y la contextualización de los mismos, para que todas las partes interesadas puedan utilizarlos y desarrollar las aplicaciones según sus necesidades. No se puede subestimar la importancia de enfocarse en estándares de datos abiertos. Para lograr una integración de datos completa y ofrecer una vista holística del historial del activo, su estado actual y simular estados futuros, el hilo digital debe admitir la integración de datos de diferentes fuentes, como datos de diseño, datos de fabricación, datos de sensores, registros de mantenimiento y datos operativos.

Varias organizaciones de desarrollo de estándares (SDOs²⁸, por sus siglas en inglés) han desarrollado especificaciones para abordar la interoperabilidad en gemelos digitales [10]. Se proporcionan algunos ejemplos en la Tabla 1. Además, el ámbito de los gemelos digitales ha presenciado una creciente expansión de actividades de código abierto, especialmente en el desarrollo de plataformas de gemelos digitales y en la gestión de los datos. Estas iniciativas de código abierto contribuyen a la colaboración, la innovación y una mayor adopción de gemelos digitales.

3.5. Conectar la trazabilidad y la gestión de la configuración a lo largo del ciclo de vida

Las organizaciones deben planificar la trazabilidad de los datos para garantizar que éstos se gestionen correctamente y sean fáciles de encontrar y que los flujos de datos y las interfaces se mantengan tanto en las operaciones como en el mantenimiento para los sistemas ya en servicio. Esto puede ser difícil de trasladar contractualmente a las especificaciones de contrato, incluso en industrias altamente reguladas, y especialmente en el caso de sistemas complejos diseñados con procesos de desarrollo dispares. Además, la información técnica y logística de un sistema puede residir y mantenerse en sistemas dispares. Vincular las bases de datos y los conjuntos de datos subyacentes del Análisis de Soporte Logístico (LSAR²⁹, por sus siglas en inglés) requiere un plan para establecer y mantener estas conexiones, especialmente en entornos distribuidos y desconectados.

3.6. Habilitar intencionalmente la automatización y el análisis

Una organización que se somete a una transformación digital puede requerir tecnología de sensores y conjuntos de datos de monitorización para generar perfiles operativos o apoyar el desarrollo de modelos de comportamiento. La tecnología de monitorización puede estar presente en el sistema, mostrando la condición del activo o su comportamiento. La modelización y simulación del activo en remoto, o de la flota, se hace posible al combinar datos en tiempo real con información histórica y contextual. En la práctica, esto requiere una captura significativa de datos y su usabilidad, incluida una planificación cuidadosa para mantener las conexiones entre fuentes de datos autorizadas y métodos robustos para mantener enlaces entre datos autorizados y métodos automatizados para ingresar múltiples fuentes y formatos en entornos seguros. Sin embargo, la falta de una base de procesos preexistentes y la necesidad de simplificar o transformar los procesos existentes son barreras significativas para que las organizaciones adopten dicha automatización [11].

3.7. Gestionar activamente el conocimiento para apoyar las operaciones y el sostenimiento

Al introducir nuevas tecnologías en un sistema de producción, los usuarios y mantenedores generalmente reciben la formación necesaria para una implementación fluida del sistema. Esto generalmente requiere unificar las relaciones, los procesos y los datos en las operaciones del sistema de producción, muchos de los cuales atraviesan límites de la propia organización. Los hilos digitales pueden facilitar la transferencia de los conocimientos y las lecciones aprendidas a medida que una organización

integra y realiza transiciones de un activo a otro. Los datos históricos, las acciones de mantenimiento y el análisis de fallos de activos similares pueden compartirse y utilizarse para mejorar las estrategias de mantenimiento predictivo. Además, los hilos digitales también pueden facilitar la colaboración y el intercambio de conocimientos. El personal autorizado, como los operadores, los ingenieros o los expertos en diferentes áreas (logística, "safety", etc.), puede acceder al hilo digital para compartir ideas y conocimientos, intercambiar información, mantener bibliotecas de datos o brindar apoyo al sistema. Este entorno colaborativo mejora las capacidades de resolución de problemas y facilita los procesos de toma de decisiones.

3.8. Impacto de la implementación en el ciclo de vida

El desarrollo de gemelos digitales de sistemas complejos presenta desafíos en múltiples áreas, que requieren un enfoque estratégico desde su inicio. Por lo general, el desarrollo de los gemelos digitales comienza con un enfoque en el diseño y la producción, pero este enfoque debe evolucionar para que el gemelo digital pueda ser utilizado en todo el ciclo de vida del sistema, desde la preparación para la entrada en servicio hasta las operaciones, el sostenimiento y, eventualmente, su baja.

El valor inherente de los gemelos digitales radica en su capacidad para mejorar la disponibilidad, la accesibilidad y la precisión de la información. Esto es particularmente cierto cuando los datos pueden ser contextualizados en el entorno operativo de los usuarios. Estas capacidades, concebidas en las etapas tempranas del ciclo de vida, son fundamentales para obtener beneficios a largo plazo. Facilitan no solo una integración más fluida de las nuevas tecnologías, sino que también respaldan a la fuerza laboral mientras navega por paradigmas cambiantes. Ajustar las prácticas tradicionales de gestión de programas y proyectos y realizar inversiones iniciales en una infraestructura digital robusta son pasos esenciales en esta dirección.

Una fase crítica en esta progresión implica mejorar el gemelo digital tradicionalmente utilizado para el diseño y la producción con funcionalidades necesarias para el sostenimiento del sistema una vez que éste ha pasado a la vida operativa. En línea con los principios de ingeniería de sistemas, involucrar a los usuarios finales, operadores y mantenedores desde el principio en la redacción de conceptos operativos, la definición de escenarios y el establecimiento de requisitos es clave para el éxito.

4. CONCLUSIONES

La ingeniería digital, apoyada por la Gestión del Ciclo de Vida del Producto (PLM) y la Ingeniería de Sistemas Basada en Modelos (MBSE, por sus siglas en inglés), desempeña un papel crucial en las etapas de preparación para la entrada en servicio, las operaciones, el sostenimiento y la baja del ciclo de vida de los sistemas. Estas metodologías facilitan la creación de gemelos digitales e hilos digitales que sirven como modelos virtuales dinámicos que reflejan activos del mundo real a lo largo de su vida útil de servicio. Tales fundamentos digitales permiten la evolución continua y la mejora de los sistemas, al proporcionar modelos para realizar actividades relacionadas con el sostenimiento, como el mantenimiento. Este enfoque integral garantiza que los sistemas no solo cumplan con los requisitos iniciales, sino que también se adapten a las necesidades futuras, extendiendo así su vida útil y mejorando el rendimiento general.

La implementación de los principios de la ingeniería digital afronta desafíos, incluida la resistencia organizacional al cambio, la existencia de infraestructuras obsoletas y carencia de las habilidades profesionales necesaria. Superar estos obstáculos requiere una estrategia enfocada que destaque los beneficios únicos de la transformación, como la mejora de la eficiencia y la optimización de las operaciones. Además, el rápido avance tecnológico exige una estrategia de transformación digital flexible y colaborativa, que permita a las organizaciones explorar soluciones innovadoras y adaptarse a los nuevos desafíos. El éxito de la transformación digital también implica tratar los datos como un activo estratégico, establecer una sólida gobernanza de datos y garantizar la calidad y seguridad de los mismos. Al priorizar estos elementos, las organizaciones pueden maximizar los beneficios de la ingeniería digital, lo que lleva a sistemas más fiables, rentables y de alto rendimiento a lo largo de su ciclo de vida.

REFERENCIAS

1. Soybel, J. Designing a Make vs. Buy Strategy for Expendable and Attributable Aircraft Engine Development, 2021. Doctoral dissertation, Massachusetts Institute of Technology.
2. Singh, J., & Walia, N. K. A comprehensive review of cloud computing virtual machine consolidation, 2023. IEEE Access.
3. Marino & Castelle, 2021. <https://www.amentum.com/blog/part-1-digital-engineering-supports-asset-availability-and-sustainment/>.
4. Mcneely, J. R. X-Ray Vision: Application of Augmented Reality in Aviation Maintenance to Simplify Tasks Inhibited by Occlusion, 2022. Doctoral dissertation, Monterey, CA; Naval Postgraduate School.
5. Fingas, J. Microsoft's HoloLens headsets are giving US Army testers nausea. Engadget. Published 13 October 2022: <https://www.engadget.com/microsoft-hololens-fails-us-army-tests-135010970.html>.
6. Bolton, M. T. W., Waterworth, S. N., & McClurg, R. J. Enabling, Equipping and Empowering the Support Enterprise through Digital Transformation. (2018, October) In Conference Proceedings of INEC.
7. D. Galar, K. Goebel, P. Sandborn & U. Kumar. Prognostics and Remaining Useful Life (RUL) Estimation, 2022. Taylor Francis, 3: p. 89–134 (DOI: 10.1201/9781003097242).
8. Henderson, K., McDermott, T., Van Aken, E., & Salado, A. Towards Developing Metrics to Evaluate Digital Engineering. Systems Engineering, 2023. 26, p. 3-31. doi:<https://doi.org/10.1002/sys.21640>.
9. Thompson, G. E., & McGrath, M. Technical Data as a Service (TDaaS) and the Valuation of Data Options, 2019. Acquisition Research Program.
10. Song, J., Le Gall, F. Digital Twin Standards, Open Source, and Best Practices, 2023. In: Crespi, N., Drobot, A.T., Minerva, R. (eds) The Digital Twin. Springer, Cham. https://doi.org/10.1007/978-3-031-21343-4_18.
11. Lyjack, 2019. <https://www.linkedin.com/pulse/how-digital-coach-can-help-improve-production-systems-craig-lyjak>.

NOTAS FINALES

1. MBSE (*Model Based Systems Engineering*).
2. En el ámbito de este capítulo, el término “sostenimiento” incluye las actividades de mantenimiento, suministro y gestión de repuestos y materiales, e ingeniería del ciclo de vida.
3. En el ámbito de este capítulo, se entiende que la Fase de Servicio de los Sistemas incluye la Etapa de Preparación para la Entrada en Servicio, la Etapa de Vida Operativa (donde se ejecutan las Operaciones y el Sostenimiento), y la Etapa de Baja.
4. En inglés “*digital thread*”. Este concepto se define en el siguiente punto.
5. CAD (*Computer Aided Design*).
6. Puede accederse a la misma en el siguiente enlace: <https://www.dau.edu/glossary/digital-thread> (consultado el 23/04/2024).
7. Sol (*System of Interest*).
8. El término “*hardware in the loop testing*” se refiere a una técnica de prueba utilizada en el desarrollo y validación de los sistemas, especialmente en la industria de la ingeniería y la fabricación. En este enfoque, se integran componentes físicos reales del sistema (hardware) con un entorno de simulación computarizado (loop). El propósito es evaluar el rendimiento del hardware real en un entorno controlado y reproducible que simula unas condiciones operativas determinadas.
9. DevOps es una metodología de desarrollo de software utilizada como un conjunto de prácticas y herramientas que integra y automatiza el trabajo de desarrollo de software (Dev) y las operaciones en las tecnologías de la información (Ops) como un medio para mejorar y acortar el ciclo de vida del desarrollo de sistemas.
10. AR (*Augmented Reality*). es una tecnología que combina el mundo físico con elementos virtuales generados por computadora, superponiendo información digital (como imágenes, videos, sonido o gráficos 3D) en tiempo real sobre la vista del mundo real, generalmente a través de dispositivos como smartphones, tabletas o gafas especiales.
11. VR (*Virtual Reality*). Es una tecnología que permite a los usuarios sumergirse en un entorno completamente digital generado por computadora que simula la experiencia sensorial del mundo real. Utilizando dispositivos como auriculares especiales, guantes o trajes de cuerpo completo, los usuarios pueden interactuar con este entorno virtual y sentirse completamente inmersos en él.
12. La maqueta digital se refiere a una representación virtual detallada de un objeto físico, sistema o proceso. Esta representación digital es una réplica precisa del objeto real, que incluye información geométrica, estructural y funcional, así como datos sobre su comportamiento y características. La maqueta digital se utiliza como componente central del gemelo digital para simular, analizar y gestionar el objeto físico en un entorno virtual.
13. IETP (*Interactive Electronic Technical Publications*).
14. IETM (*Interactive Electronic Technical Manuals*).
15. MBPS (*Model Based Product Support*).
16. En inglés “*authoritative source of truth*”. En el ámbito de la Ingeniería de Sistemas Basada en Modelos, una “fuente autorizada de verdad” puede ser un modelo digital centralizado y actualizado continuamente que contiene todos los datos relevantes y necesarios para el diseño, desarrollo, operación y mantenimiento de un sistema. Esta fuente de verdad puede ser utilizada por todas las partes interesadas y equipos involucrados en el ciclo de vida del sistema para garantizar la coherencia, consistencia y precisión de la información en todas las etapas del proceso.
17. CBM (*Condition Based Maintenance*).
18. RUL (*Remaining Useful Life*).
19. Overhaul. Se refiere a un proceso completo de revisión, reparación y renovación de un equipo, máquina o sistema. Este proceso se lleva a cabo periódicamente para restaurar el equipo a un estado óptimo de funcionamiento y prolongar su vida útil. Implica desmontar, limpiar, inspeccionar, reparar o reemplazar partes desgastadas, y volver a montar el equipo para garantizar su rendimiento y fiabilidad. Dependiendo del alcance del “overhaul”, y de la calidad del trabajo realizado, el equipo restaurado puede ser prácticamente indistinguible de uno nuevo de fábrica en términos de rendimiento y fiabilidad.
20. CAM (*Computer Aided Manufacturing*).
21. PLM (*Product Life Cycle Management*).
22. La “*curación de modelos*”, en el ámbito de la ingeniería basada en modelos, se refiere al proceso de asegurar la calidad, consistencia y fiabilidad de los modelos utilizados en el desarrollo y análisis de sistemas.
23. IOT (*Internet of Things*). Se refiere a la red de dispositivos físicos que están conectados entre sí y a internet, permitiéndoles recopilar y compartir datos. Estos dispositivos pueden incluir desde dispositivos electrónicos portátiles hasta sensores industriales y otros equipos de monitorización. La IoT permite que estos dispositivos se comuniquen entre sí y con sistemas externos, lo que brinda la posibilidad de recopilar datos en tiempo real, automatizar procesos, mejorar la eficiencia y generar información para la toma de decisiones.
24. El término “*puesta en servicio*” se refiere a las actividades necesarias para poner en funcionamiento el sistema en su entorno operacional, donde estará disponible para su uso por parte de los usuarios finales.
25. IaaS (*Infrastructure as a Service*).
26. PaaS (*Platform as a Service*).
27. En inglés “*edge computing*”. Se refiere a la práctica de procesar, almacenar y analizar los datos cerca de la fuente de generación de los mismos, en lugar de enviarlos a una ubicación centralizada, como un centro de datos en la nube, para su procesamiento y análisis.
28. SDO (*Standards Development Organization*).
29. LSAR (*Logistics Support Analysis Record*). Es una base de datos o conjunto de registros que contiene información detallada sobre los requisitos de apoyo logístico para un sistema. Puede incluir datos sobre piezas de repuesto, herramientas especiales, documentación técnica, mantenimiento y reparación, entre otros aspectos relacionados con el apoyo logístico del sistema.

BIOGRAFÍAS

DRA. KAITLYNN CASTELLE

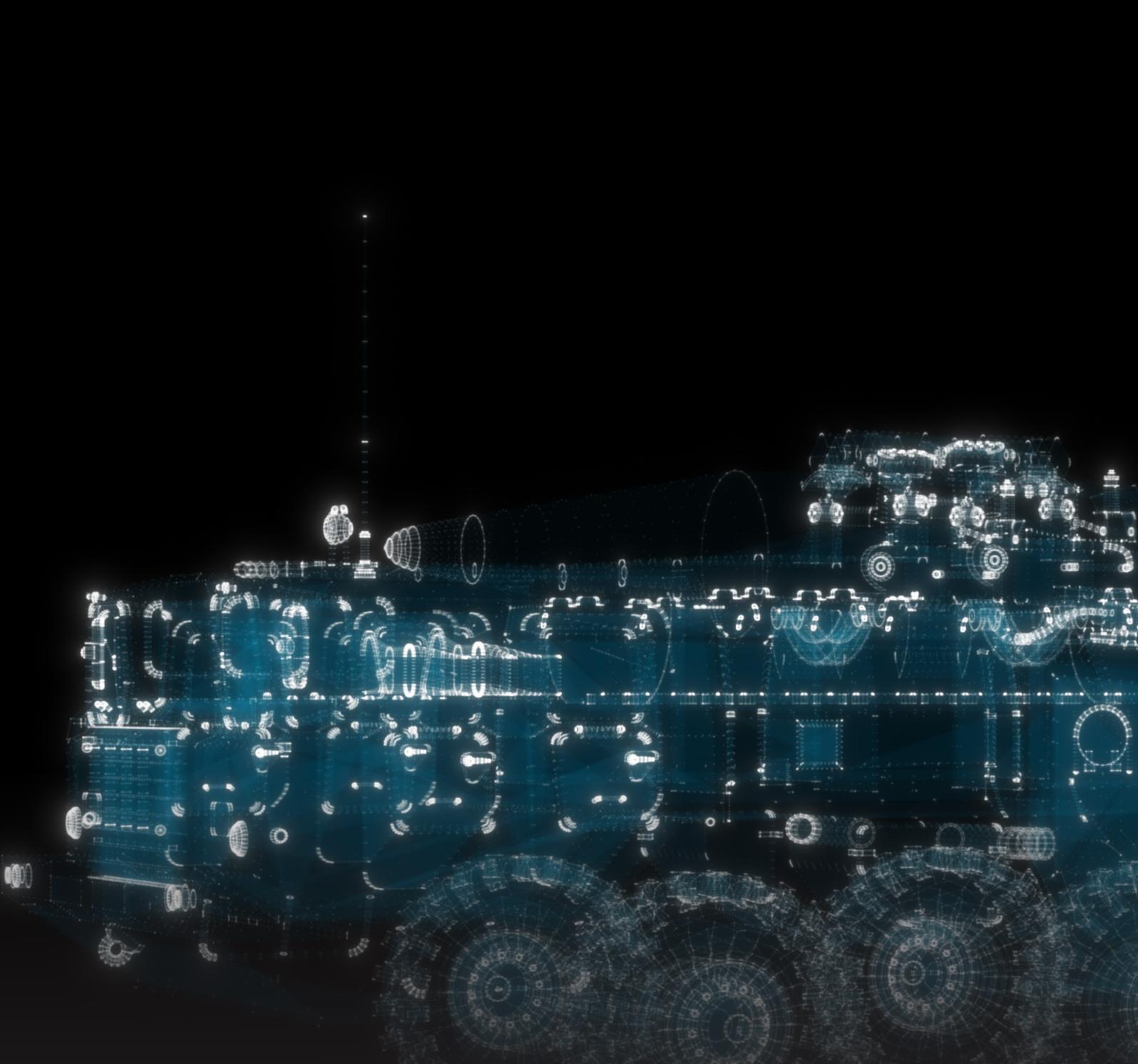
La Dra. Kaitlynn es Ingeniera de Investigación Asociada en el área de misión de Adquisiciones y Seguridad Industrial en el Laboratorio de Investigación Aplicada para Inteligencia y Seguridad de la Universidad de Maryland. Anteriormente, trabajó en defensa como Científica de Datos y gestora *agile* de productos en apoyo a la Oficina del Programa de Submarinos COLUMBIA. Su trabajo de alta visibilidad en el análisis del ciclo de vida, aprovechando la simulación de Monte Carlo, respaldó las inversiones en construcción avanzada en la Base Industrial de Submarinos. Es cofundadora y Gerente de Programa de la célula de innovación en ingeniería digital Proyecto Blue de la Armada estadounidense, centrada en el avance de las capacidades del ciclo de vida en apoyo de la misión de disuasión estratégica basada en el mar en colaboración con otros programas del Departamento de Defensa de los Estados Unidos. Antes de su trabajo en el sector de defensa, fue miembro adjunto del cuerpo docente en la Universidad de Old Dominion. Obtuvo su licenciatura en matemáticas aplicadas y maestría y doctorado en gestión de ingeniería de la Universidad de Old Dominion, en Estados Unidos.



MIGUEL ÁNGEL COLL MATAMALAS

Miguel Ángel Coll Matamalas es Ingeniero de Sistemas en Isdefe. Actualmente realiza labores de consultoría en la Jefatura de Apoyo Logístico de la Armada colaborando en el desarrollo de doctrina logística, y en actividades relacionadas con la obtención de Apoyo Logístico Integrado para los Programas de Obtención en curso. En particular, colabora en el desarrollo del Gemelo Digital del Programa F-110. También participa en el grupo de trabajo OTAN CNAD/LCMG/WG1 responsable de revisar la norma ALP-10 (*NATO Guidance for Integrated Life Cycle Support*). Miguel Ángel es Ingeniero Naval por la UPM (Universidad Politécnica de Madrid) y ha cursado un Executive MBA en el IESE (Universidad de Navarra). En su etapa profesional anterior, ha ocupado diversos cargos relacionados con el mantenimiento y la gestión de activos, tanto en el sector marítimo como en infraestructuras del sector del agua. Con el objetivo de aplicar esta experiencia previa en diferentes áreas del sostenimiento, Miguel Ángel trabaja en la mejora de la soportabilidad de los sistemas mediante la influencia en el diseño, en el desarrollo de gemelos digitales, y en la aplicabilidad del mantenimiento prescriptivo (RxM) y del Apoyo Logístico Basado en Modelos (MBPS).





“La ingeniería de sistemas requiere conocimiento y habilidad. ¡Es imposible adquirir habilidades de ingeniería de sistemas en un curso o taller! [...] Es imposible aprender ingeniería de sistemas con una herramienta de ingeniería de sistemas.”

A.W. Wymore

Los seis capítulos anteriores son evidencia de la profunda transformación a la que está sometida la ingeniería de sistemas. En las últimas tres décadas, hemos sido testigos del gran crecimiento, innovación y maduración en esta rama. Hemos visto expandirse el conocimiento de manera considerable y estamos experimentando la adopción de nuevas metodologías y la ruptura de los límites tradicionales que una vez confinaron nuestro entendimiento y aplicación de los principios de la ingeniería de sistemas. El campo ha cambiado de manera inimaginable con respecto al momento de la publicación de los “libros azules” de Isdefe.

La conclusión de este primer cuaderno sobre la ingeniería de sistemas moderna marca un nuevo capítulo en su difusión en España. Este cuaderno señala el comienzo de la nueva serie de “libros azules”. Lo consideramos prueba del progreso realizado, así como una guía para el futuro. Encapsula algunos temas clave del estado actual de la práctica en esta especialidad, destacando áreas incipientes y prácticas innovadoras que están moldeando el futuro del campo. La diversidad de temas cubiertos, desde la aplicación de la ingeniería de sistemas basada en modelos (MBSE) hasta la integración de inteligencia artificial y gemelos digitales en la gestión del ciclo de vida del sistema, ilustra su naturaleza multifacética y su importancia en un mundo en constante cambio. En el futuro, pretendemos lanzar cuadernos adicionales en la serie para profundizar en cada uno de estos temas, intentando también que los avances sean una oportunidad para que los ingenieros de Isdefe los apliquen a los desafíos que enfrentan, en los programas técnicamente avanzados de nuestra sociedad, particularmente en estos tiempos de inseguridad e incertidumbre.

Lanzamos este cuaderno con orgullo y optimismo. Nuestro objetivo es que sirva como un punto de partida para los profesionales que utilizan esta ingeniería, guiándoles a través de las complejidades de los desafíos de ingeniería modernos e inspirándoles a empujar los límites de lo que es posible en la ingeniería de sistemas. Extendemos nuestra más profunda gratitud a todos los que han contribuido a este trabajo, y a los lectores, que tienen el desafío de adoptar y evolucionar las prácticas de ingeniería de sistemas modernas que hemos presentado aquí.

Que este epílogo no signifique el fin, sino más bien el comienzo de un viaje hacia nuevas fronteras en la ingeniería de sistemas y su más amplia utilización. Que las perspectivas contenidas en estas páginas les inspiren, impulsando el desarrollo de sistemas que no sólo sean adecuados para su propósito, sino también resilientes y sostenibles ante los desafíos del futuro. Juntos, estamos ante el umbral de una nueva era, listos para explorar, innovar y moldear el futuro de la ingeniería de sistemas para ayudar a hacer una sociedad mejor.

Dr. Alejandro Salado
Universidad de Arizona



Isdefe



Isdefe

C/ Beatriz de Bobadilla, 3
28040 Madrid
Tel.: +34 91 411 50 11
Email: general@isdefe.es
www.isdefe.es

