3

UN MARCO DE TRABAJO PARA GARANTIZAR LA TRAZABILIDAD EN SISTEMA DE SISTEMAS

Leticia Morales Trujillo, Miguel Ángel Olivero González, Francisco José Domínguez Mayo, Julián Alberto García García, Manuel Mejías Risoto

En los últimos años, la trazabilidad se ha convertido en un procedimiento imprescindible en muchos sectores de la industria ya que es clave a la hora de mantener la seguridad de un proceso. La noción de la trazabilidad no es nueva, pero se siguen realizando muchas investigaciones llevadas a cabo en diferentes dominios. Una de las razones de estas investigaciones es que hoy día, el progreso en el mundo digital implica que los productos y sistemas tecnológicos sean cada vez más interdependientes. Mientras que tradicionalmente los sistemas se dirigían a un solo dominio, los sistemas actuales deben estar compuestos por múltiples sistemas que deben integrarse de manera coherente. Es por ello, por lo que hemos sido testigos del crecimiento de los conocidos como sistemas de sistemas (SoS). Existe una amplia variedad de metodologías y dominios de aplicación en la literatura para formar soluciones enmarcadas en el contexto de SoS, pero no existe un consenso unificado para su uso y menos aun cuando se trata de entornos ágiles de integración y despliegue continuo en los que los requisitos de trazabilidad son críticos. Además, a lo largo de los años se han producido varios errores relacionados con un control deficiente de la trazabilidad. Por tanto, este capítulo presenta un marco ágil que tiene como objetivo garantizar la trazabilidad de un SoS desde las primeras etapas. Este marco unifica el descubrimiento, el desarrollo y las operaciones, brindando una

cobertura total en la conformación de la solución. Finalmente, presentamos un caso de estudio como trabajo de futuro, que se basa en la aplicación de nuestro marco sobre laboratorios inteligentes de reproducción asistida.

3.1 INTRODUCCIÓN

Hoy, el progreso en el mundo digital implica que los productos y sistemas tecnológicos sean cada vez más interdependiente, lo que lleva a una complejidad creciente en el diseño y desafíos para la innovación. Mientras que tradicionalmente los sistemas se dirigían a un solo dominio, los sistemas actuales deben estar compuestos por múltiples sistemas que deben integrarse de manera coherente. Es por ello, por lo que hemos sido testigos del crecimiento de los conocidos como sistemas de sistemas (SoS) [1, 2, 3, 4].

Los SoS son una colección de sistemas dedicados u orientados a tareas que combinan sus recursos y capacidades para crear un sistema nuevo y más complejo que ofrece más funcionalidad y rendimiento que simplemente la suma de los sistemas constituyentes. Se crean a través de un proceso de síntesis de sistemas independientes previamente existentes, pero no relacionados [5, 6, 7, 8, 9]. Tal síntesis crea valor dando lugar a que el nuevo sistema mejore las utilidades o funcionalidades de cada uno de los sistemas anteriores o habilite funciones novedosas del nuevo sistema holístico. Algunos ejemplos de sistemas de sistemas son la gestión del tráfico aéreo, la red ferroviaria europea, transporte terrestre integrado, servicio de emergencia y personal la gestión de la salud, los medios de comunicación, entre muchos otros [10, 11].

Para poder diseñar, analizar, implementar y mantener los llamados sistemas de sistemas (SoS), se requiere un enfoque de Ingeniería de Sistemas de Sistemas (SoSE) [12]. La Ingeniería de Sistemas de Sistemas (SoSE) es el conjunto de procesos, herramientas y métodos de desarrollo para diseñar, rediseñar e implementar soluciones a los desafíos del sistema [13, 14]. Aborda el diseño, desarrollo y operaciones de programas en evolución. Es decir, busca optimizar la red de varios sistemas interactivos heredados y nuevos sistemas reunidos para satisfacer múltiples objetivos de un programa.

Una metodología de desarrollo de software eficaz en SoSE debería preparar a los responsables de la toma de decisiones para diseñar soluciones para problemas del SoS. Debido a la amplia variedad de metodologías y dominios de aplicación presentes en la literatura, no existe un único consenso unificado para los procesos involucrados en la Ingeniería de Sistemas de Sistemas [15].

En muchos sectores, es fundamental contar con un software de trazabilidad que registre y monitorea el rastro de las entidades que interactúan con él. Estas entidades pueden ser objetos, actores o actividades. Si además queremos diseñar una solución de software que garantice el cumplimiento de la trazabilidad en un entorno SoS, la complejidad aumenta aún más. Entendemos la trazabilidad como conjunto de medidas, acciones y procedimientos que permiten registrar e identificar una entidad desde su origen hasta su destino final [16, 17].

A lo largo de los años se ha tenido constancia de varios errores referentes al mal control de la trazabilidad; en el ámbito de la medicina en general [18], en el ámbito de la alimentación [19, 20], en el ámbito de la reproducción asistida [21, 22, 23, 24], entre muchos otros.

Por lo tanto, se pretende desarrollar una solución que garantice el control y seguimiento de las entidades que interactúan con los sistemas, de manera que el riesgo de error se reduzca a niveles mínimos, haciendo uso de las tecnologías de la información como eje central de la solución y aumentando la confianza de los implicados.

Para la consecución de dicho objetivo, este documento presenta una metodología ágil que tiene como objetivo asegurar la trazabilidad de un SoS desde las primeras etapas, aunque puede ser aplicable a sistemas simples. Este marco unifica el descubrimiento, desarrollo y operaciones, lo que implica una cobertura total en la conformación de la solución [25]. Este marco será, en el futuro, aplicado en un laboratorio inteligente de reproducción asistida. Este caso de estudio se presenta en este artículo como trabajo futuro.

El presente trabajo está estructurado de la siguiente manera; En el apartado 3.2 se presentan los antecedentes del trabajo y los principales objetivos que se van a conseguir tanto con este trabajo como en el futuro. En el apartado 3.3 se presenta el marco propuesto para la generación de soluciones software que garanticen la trazabilidad. En el apartado 3.4 se exponen trabajos afines, es decir, publicaciones que consisten en metodologías de descubrimiento, desarrollo y operaciones, ya existentes en la literatura. La sección 3.5 muestra las conclusiones generales del trabajo. Finalmente, en la sección 3.6, se analizan ampliamente las líneas de trabajo abiertas.

3.2 ANTECEDENTES Y OBJETIVOS

A lo largo de los años, se han producido varios errores relacionados con un control deficiente de la trazabilidad; en el ámbito de la medicina en general, como revela el caso de un estudio realizado por el College of American Pathologists en el

que los tejidos recibidos de un paciente en uno o más contenedores, correspondientes al mismo procedimiento, se les asignó un mismo número de identificación de acceso al ingreso en el servicio de patología [18]. En el ámbito alimentario, como en 1999 cuando Coca-Cola reconoció que sus refrescos en Bélgica estaban contaminados a causa de un tratamiento, pero se detectaron casos de intoxicación en Bélgica, Holanda y Luxemburgo [19] o como en el caso de la carne contaminada por listeriosis en España en 2019 [20].

En el ámbito de la reproducción asistida, los laboratorios de reproducción humana son un claro ejemplo en el que se debe conocer la trazabilidad. Es necesario conocer en todo momento la traza seguida de muestras biológicas y de pruebas. Además, los laboratorios de reproducción humana están dentro del contexto de SoS ya que hay una gran cantidad de sistemas interconectados. Como se mencionó anteriormente, los SoS son una colección de sistemas dedicados u orientados a tareas que combinan sus recursos y capacidades para crear un sistema nuevo y más complejo que ofrece más funcionalidad y desempeño que simplemente la suma de los sistemas constituyentes. La gestión de muestras es un aspecto crítico que requiere de todos los mecanismos adecuados para asegurar la trazabilidad de forma continua, evitando errores fatales.

En los Países Bajos, en 1993, una mujer blanca dio a luz a un niño negro y un niño blanco después de recibir "esperma mezclado de una pipeta mal esterilizada" [21]; en los Estados Unidos en 1998, una mujer blanca dio a luz a un bebé negro en lo que se conoció como el caso de los "huevos revueltos". Después de una "amarga batalla por la custodia", la pareja negra cuyo embrión fue implantado por error en la mujer blanca ganó la custodia del bebé negro [22]; en octubre de 2002 hubo otra confusión de la FIV (fertilización in vitro) en Gran Bretaña. En abril, dos mujeres recibieron los "embriones equivocados" en una confusión que involucró a tres mujeres. A una mujer le implantaron sus propios "embriones de peor calidad", mientras que su "pareja de mejor calidad" fue asignada a otra mujer cuyos embriones fueron enviados por error a una tercera mujer. Las mujeres que recibieron los "embriones equivocados" quedaron "devastadas" y "traumatizadas" luego de ser sometidas a un "procedimiento de emergencia para extraer los embriones" [23]; En 2016, el Hospital Universitario de Utrecht hizo pública la posible fecundación de ovocitos de 26 mujeres con espermatozoides ajenos a su pareja, es decir, una veintena de mujeres o parejas han engendrado hijos con el esperma del hombre que no era el indicado. Finalmente, el centro detectó que la pipeta que se había utilizado en algunos procedimientos de fertilización de ovocitos estaba contaminada con el esperma de otro paciente [24].

La trazabilidad de un sistema se traduce en una serie de requisitos que se deben cumplir. Los requisitos de un sistema surgen de las necesidades del cliente,

de las limitaciones del entorno donde se va a implementar o de la gestión de la información que debe realizar el sistema. Normalmente representarán valores que deben cumplir al menos o como máximo cada uno de los aspectos desarrollados. Sirven para limitar la funcionalidad o construcción del sistema, asumiendo límites para el diseño y enumerando todas las funcionalidades que debe cubrir.

Hay varios tipos de requisitos englobados en dos grandes grupos: (i) requisitos funcionales que afectan directamente la funcionalidad principal del sistema y (ii) requisitos no funcionales, que no representan la funcionalidad principal del sistema, pero imponen el diseño o restricciones de implementación. Son propiedades o cualidades que debe tener el sistema.

Los requisitos de trazabilidad se ubican dentro del grupo de requisitos no funcionales e incluyen todas las acciones del sistema, tales como: acciones de las cuales se deben almacenar las trazas (track), el formato y el medio de almacenamiento de cada tipo de traza, la gestión de la caducidad de las trazas y su historial, control y medios de acceso a las trazas del sistema, etc. La trazabilidad podría definirse como las acciones que permiten el seguimiento de diferentes tipos de entidades, mediante la monitorización continua de los parámetros físicos y gracias al uso de nuevas tecnologías [26]. La trazabilidad no solo cubre los requisitos básicos de que los productos se pueden rastrear a lo largo de la cadena de valor, sino también la posibilidad de especificar de qué están hechos y cómo se han procesado.

Para registrar los acuerdos realizados con un cliente durante el desarrollo del sistema, se utilizan los contratos. Un contrato no es más que un acuerdo entre dos o más partes, un entorno que define qué se puede hacer, cómo se puede hacer, qué pasa si no se hace algo, etc. Es decir, unas reglas que permiten a las partes que lo acepten entiendan en qué consistirá la interacción que realizarán. [27, 28, 29].

Hasta ahora, los contratos han sido documentos verbales o documentos costosos, sujetos a leyes y jurisdicciones que a veces requieren notarios. Es decir, más costos, tiempo y terceros involucrados en el proceso. Debido a esto, no son accesibles para todo el mundo. En cambio, un contrato inteligente es un programa informático que facilita, asegura, hace cumplir y ejecuta acuerdos registrados entre dos o más partes.

Los contratos inteligentes tienen como objetivo brindar una seguridad superior al derecho contractual tradicional y reducir los costos de transacción asociados con la contratación, ya que pueden sujetarse y hacerse cumplir por sí mismos, de forma autónoma y automática, sin intermediarios o mediadores.

Por todo lo anterior, se pretende desarrollar una solución que integre un contrato inteligente donde se recojan todos y cada uno de los requisitos de trazabilidad

identificados en un entorno de SoS, para que las entidades puedan registrarse y monitorearse continuamente en la cadena de valor.

En este contexto, el objetivo de este trabajo es proponer un marco ágil, es decir, promover la iteración continua del ciclo de vida del desarrollo de software, enmarcado dentro del contexto de la Ingeniería de Sistemas de Sistemas, para lograr la solución.

3.3 MARCO PROPUESTO

Como se mencionó anteriormente, nuestro marco propuesto está enfocado en garantizar la trazabilidad de una solución de software enmarcada en el contexto de SoS desde las primeras etapas. Es decir, asegura el registro y control continuo de la interacción de diferentes tipos de entidades con un sistema, lo que permite el seguimiento. Para asegurar la trazabilidad desde las primeras etapas es necesario: validar, verificar y monitorear el registro y control de la solución que se creará (Figura 3.1).

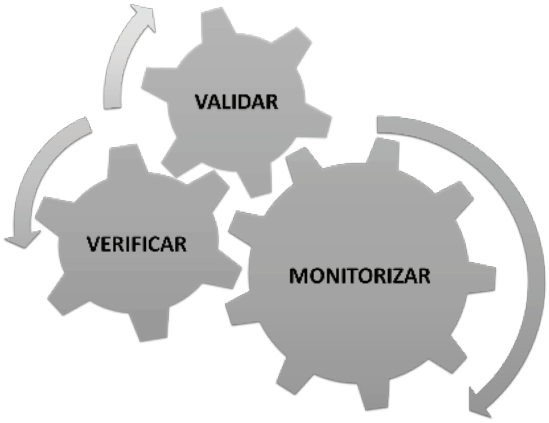

Figura 3.1. Garantizar la trazabilidad.

- Validar. Confirma que lo que se está haciendo se está haciendo bien y que se logra el objetivo previsto. Nos ayuda a conocer, a futuro, que se hace bien el registro y control de las entidades y se registra y monitorea lo necesario.
- Verificar. Comprueba que lo que se dice esté realmente hecho. Nos ayuda a verificar, un pasado, lo que se supone que debe hacerse, es y funciona como se esperaba.

- Monitorizar. Controlar el desarrollo de una acción o un evento a través de uno o más monitores. Nos ayuda a conocer, en la actualidad, el registro y control de todas las entidades.

Estas acciones (validar, verificar y monitorear) se llevan a cabo en todo el marco propuesto de manera preventiva, es decir, tomando precauciones o medidas con anticipación para evitar riesgos, y de manera correctiva, realizando las modificaciones necesarias para eliminar errores.

Esta propuesta metodológica se basa en metodologías empíricas y consta de tres etapas principales: descubrimiento (DIS), desarrollo (DEV) y operaciones (OPS). Por un lado, en la etapa de descubrimiento (apartado 3.3.1) se lleva a cabo una serie de subetapas propuestas por el proceso Design Thinking [30] y es donde se realiza la validación del registro y el control de la solución a realizar. Por otro lado, tanto la etapa de desarrollo (sección 3.3.2) como la etapa de operaciones (sección 3.3.3) aquí propuestas siguen las prácticas establecidas en DevOps [31, 32, 33] y el flujo de trabajo de GitFlow [34, 35, 36], y es donde se realiza la verificación y monitorización del registro y control de la solución a crear.

Como se mencionó anteriormente, el marco propuesto está dentro del contexto de la metodología de desarrollo ágil. Las metodologías de desarrollo ágiles son aquellas que promueven la iteración continua del ciclo de vida del desarrollo y que siguen una serie de principios y valores ágiles.

La Figura 3.2 muestra la iteración continua que tiene lugar entre el proceso general y entre las diferentes etapas. Continuamente se proponen nuevos enfoques en la etapa de descubrimiento, a su vez se están desarrollando otros enfoques en la etapa de desarrollo y ofreciendo un servicio en la etapa de operaciones.

Además, se pueden visualizar flechas de retroalimentación, que ayudan a confirmar que todo se hace según lo planeado y que se logran los objetivos marcados (DEV → DIS), a verificar que lo planeado realmente se hace (OPS → DEV) y a controlar qué se proporciona como un servicio necesario (OPS → DIS).

En este caso, para asegurar la trazabilidad, es muy importante validar, verificar y monitorear continuamente el registro y control de las entidades que interactúan con el sistema. Estas acciones son propias de la figura del tester, por lo que los principios y valores que se siguen en este framework son los propuestos en Agile Testing [37]:

- Construyendo calidad en: los equipos se enfocan en prevenir malentendidos sobre el comportamiento de las características, así como en prevenir defectos en el código.

- Guiar el desarrollo con ejemplos concretos: usar prácticas como el desarrollo impulsado por pruebas de aceptación (ATDD), el desarrollo impulsado por el comportamiento (BDD) o la especificación por ejemplo (SBE).
- Incluir actividades de prueba como tener conversaciones para construir un entendimiento compartido; hacer preguntas para probar ideas y suposiciones; automatizar pruebas; realizar pruebas exploratorias; prueba de atributos de calidad como rendimiento, confiabilidad y seguridad; y aprender del uso de la producción.
- Usar retrospectivas de todo el equipo y pequeños experimentos para mejorar continuamente las pruebas y la calidad y encontrar lo que funciona en su contexto.

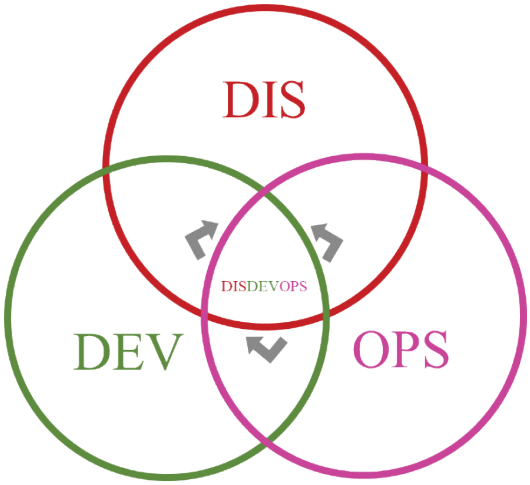

Figura 3.2. Iteración en la propuesta metodológica.

Como se indica a lo largo del trabajo, el marco propuesto consta de tres etapas principales (descubrimiento, desarrollo y operación), que a su vez están formadas por una serie de subetapas. En la figura 3.3 se muestra la estructura del marco en general, es decir, se muestran las etapas y subetapas que lo componen. Independientemente de las iteraciones entre las diferentes etapas y subetapas (Figura 3.2). También se indica mediante puntos, las subetapas donde se llevan a cabo las acciones de validación, verificación y monitorización, tanto preventivas como correctivas.

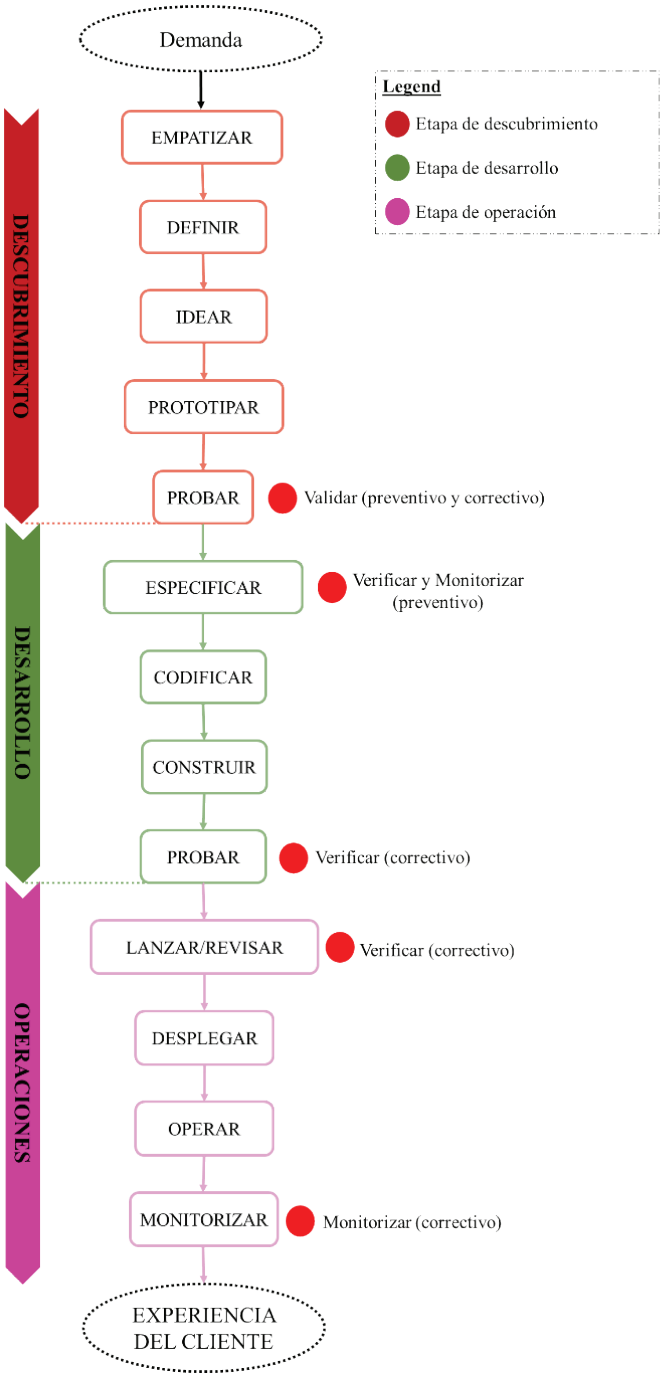

Figura 3.3. Marco propuesto.

3.3.1 Descubrimiento

Su objetivo es lograr el conocimiento y comprensión de un problema en un dominio determinado, proponer una solución y obtener como resultado una propuesta de solución validada por el usuario final. Para lograr este objetivo, se llevan a cabo una serie de etapas propuestas por el proceso Design Thinking [30]:

- Empatizar. Esta etapa tiene como objetivo conocer al público al que se dirigirán los esfuerzos.
- Definir. Toda la información recopilada en la etapa anterior permite especificar una o varias oportunidades de mejora. Será decisión del equipo priorizar cuál será atacará primero y cuál en acciones posteriores.
- Idear. Una vez analizada la información y definidos los problemas según los usuarios, se generarán las ideas que luego se filtrarán para priorizar las más plausibles.
- Prototipar. El propósito de esta etapa es crear versiones realistas y económicas del producto o servicio, donde se apliquen las ideas de la etapa anterior.
- Probar. Los prototipos se utilizan para probar con los usuarios. Las conclusiones obtenidas de ellos permiten iterar, es decir: empatizar aún más, perfeccionar ideas, volver a crear prototipos y volver a intentar obtener soluciones que en realidad respondan correctamente a los problemas de los usuarios. La validación de dicho prototipo se realiza con simulaciones [38], de forma que el usuario final pueda validar que todos los requisitos de trazabilidad necesarios quedan registrados mediante un seguimiento continuo (validación preventiva y correctiva).

En esta etapa se obtendría el prototipo de solución validada, con todos los requisitos de trazabilidad que se deben cumplir en el entorno SoS.

3.3.2 Desarrollo

El objetivo de esta etapa es obtener resultados tangibles y validados de la solución propuesta. Para lograr este objetivo se llevan a cabo una serie de etapas:

- Especificar. Es la tarea de describir el software en detalle. La especificación se realiza con suficiente detalle para que los desarrolladores calificados puedan desarrollar la solución con un esfuerzo adicional mínimo. En

esta tarea se definen en detalle todos los requisitos identificados en la etapa anterior mediante diagramas de clases, diagramas de negocio, pseudocódigo, etc. definidos por estándares ampliamente utilizados, como los propuestos por el OMG (Object Management Group). Entre otras funcionalidades, se especifica cómo se realizará el seguimiento, es decir, la visualización y control de las entidades registradas (seguimiento preventivo). Además, las pruebas se definen siguiendo buenas prácticas definidas por pruebas ágiles (pruebas de cara a la tecnología que guían el desarrollo, prueba de cara a la empresa que guía el desarrollo, prueba de cara a la empresa que critica el producto y prueba de cara a la tecnología que critica el producto) [37] que se llevará a cabo después de que se desarrolle la solución para verificar que todo está registrado y controlado según lo planeado (verificación preventiva).

- Codificar. Es la tarea de traducir la especificación en código y realizar las pruebas relevantes en el módulo desarrollado.
- Construir. Acción para integrar la implementación del código generado en el entorno de desarrollo.
- Probar. Acción para verificar que el software responde/realiza correctamente las tareas indicadas en la especificación, es decir, verificar que se registra lo especificado y controlar el cumplimiento de las disposiciones (verificación correctiva). Con base en las conclusiones obtenidas de las pruebas definidas, se realizan iteraciones para obtener soluciones que realmente respondan a lo especificado.

En esta etapa es donde se obtiene, valida y verifica la solución de software tangible en base a lo acordado con el usuario final y se integra en el contexto de SoS, pero aún no está disponible para uso del usuario final porque aún se encuentra en el entorno de desarrollo.

3.3.3 Operación

Esta etapa tiene como objetivo asegurar el correcto funcionamiento de la solución en entornos de producción una vez implementada. Esta etapa cubre todo lo relacionado con las funciones de las TI (Tecnologías de la Información) que no están relacionadas con el desarrollo y gestión de la aplicación. Para lograr este objetivo se llevan a cabo una serie de tareas:

- Lanzar/Revisar. Despliegue en el entorno de preproducción y realización de las pruebas pertinentes para verificar que se registra lo especificado y monitorear el cumplimiento de las disposiciones (verificación correctiva). Según los resultados obtenidos en las pruebas, existen dos opciones: resultado exitoso, implementación en el entorno de producción, resultado fallido, retorno al entorno de desarrollo y corrección de errores, como se propone en el flujo de trabajo de Gitflow [34].
- Desplegar. Despliegue de tareas en el entorno de producción.
- Operar. El producto está operativo y el usuario final ya lo está utilizando.
- Monitorizar. Tarea de seguimiento continuo de aspectos de la solución software (seguimiento correctivo). La información que se debe monitorear y cómo debe comportarse se define previamente en las etapas de descubrimiento y desarrollo.

En esta etapa es donde se obtiene, valida, verifica y monitorea la solución de software tangible en base a lo acordado con el usuario final, integrado en el contexto de SoS, disponible y utilizado por el usuario final.

3.4 TRABAJOS RELACIONADOS

A lo largo de este trabajo se ha descrito una metodología que garantiza la trazabilidad de los datos en un contexto de SoS desde etapas tempranas como combinación de otras metodologías. Existen diferentes metodologías y buenas prácticas para ayudar en el proceso de capturar los requisitos del software, su desarrollo, su prueba y su monitoreo. En particular, este enfoque se inspira en (1) las metodologías de diseño centrado en el usuario para capturar los requisitos y (2) las metodologías de desarrollo ágiles.

Las metodologías de diseño centradas en el usuario son un conjunto de métodos y técnicas con el propósito de conocer y comprender las necesidades, limitaciones, comportamiento y naturaleza de los usuarios [39]. Al aplicar estas metodologías, se involucran usuarios finales potenciales o incluso reales. La interacción con estos potenciales usuarios finales se lleva a cabo en un proceso iterativo-incremental. En cada ronda incremental, las especificaciones del software crecen cada vez más y se ajustan a las expectativas del usuario final. Design Thinking [30] o Design Sprint [40, 41] son dos enfoques bien conocidos entre las metodologías de diseño centradas en el usuario. En particular, las técnicas de Design Sprint se han aplicado con éxito en trabajos anteriores en un contexto industrial real en el contexto de la salud para

definir las especificaciones del software [42]. De acuerdo con esas experiencias previas, el uso de estas metodologías, que consideran involucrar a los usuarios finales en etapas tempranas de definición, producen una especificación de software que es menos susceptible a modificaciones en etapas tardías del ciclo de vida del software.

Las metodologías de desarrollo ágil son un conjunto de pautas operativas bajo las recomendaciones del manifiesto ágil [43]. Las metodologías ágiles son aquellas metodologías que cumplen con ciertos requisitos como se describe en el manifiesto. Dichas metodologías son métodos de Ingeniería de Software basados en un desarrollo incremental e iterativo. El software se desarrolla en equipos autoorganizados y multidisciplinares [44]. DevOps es uno de los conjuntos más comunes de buenas prácticas de desarrollo ágil que incluye la implementación e integración continuas [45]. Se han realizado diferentes estudios sobre la práctica DevOps aplicada en la industria y en la academia [46, 47]. Otras investigaciones, como la realizada por Virmani, acercaron la integración continua y la entrega continua [48].

Hasta donde sabemos, no existen enfoques que consideren una estrategia de diseño centrada en el usuario que involucre explícitamente el manifiesto ágil.

Los requisitos de trazabilidad son requisitos no funcionales que se han estudiado desde hace más de dos décadas. Los estudios relacionados con la trazabilidad tienen como objetivo describir el comportamiento de un sistema a la hora de atender los cambios en los datos y funcionalidades [49]. Se han llevado a cabo diferentes enfoques para mejorar los requisitos de trazabilidad en sistemas ordinarios. Estos requisitos suelen tener en cuenta el modelado de sistemas y las técnicas para una mejor aplicación [50, 51]. No obstante, todavía es un trabajo en progreso en el contexto de SoS.

El SoS es un campo emergente y no se pueden encontrar muchas metodologías en la literatura para estas construcciones de sistemas. Los autores se han centrado más en las metodologías de desarrollo entre las ya existentes aplicables en SoS. DeLaurentis, por ejemplo, propuso un método en el que se abstrae, modela y analiza un SoS en un método para detectar patrones de comportamiento [15]. Mittal y Risco-Martin han propuesto un proceso de desarrollo unificado a través de diferentes sistemas constituyentes [52]. Con respecto a los requisitos no funcionales en el SoS, el interés por la seguridad está creciendo. En los últimos años se está considerando como una característica de primera clase, que promueve nuevas investigaciones emergentes como el enfoque TeSSoS [53].

Este trabajo combina el diseño centrado en el usuario y las perspectivas del manifiesto ágil y aborda los requisitos de trazabilidad en una metodología ágil centrada en el usuario. Para caracterizar la relevancia de la trazabilidad al compartir datos, la metodología propuesta se diseña centrándose en el contexto de SoS.

3.5 CONCLUSIONES

El desarrollo de este trabajo ha implicado la inmersión en la profundidad de los procesos llevados a cabo en las metodologías ágiles de descubrimiento, desarrollo y operaciones en contextos de Sistema de Sistemas. Mediante la búsqueda de una solución para solventar el problema existente, referido al deficiente control de la trazabilidad, se amplía el conocimiento sobre las tecnologías que se están investigando y las que existen actualmente. Este trabajo surge de esta idea fundamental: *¿cómo se puede incrementar el control de la trazabilidad de las entidades dentro de un contexto de Sistema de Sistemas?*

Cada vez son más los sectores en los que es fundamental contar con un software de trazabilidad que registre y monitoree continuamente la traza de las entidades que interactúan con él y debido al avance en el mundo digital, los productos y sistemas son cada vez más interoperables y esto hace que su diseño sea más complejo. Esta interoperabilidad entre productos y sistemas es lo que se conoce como contexto de Sistema de Sistemas.

A todo ello, se suman los errores que hacen referencia al mal control de la trazabilidad que se han ido registrando a lo largo de los años.

Es por ello por lo que en el futuro se pretende desarrollar una solución que integre un contrato inteligente donde se compilen todos y cada uno de los requisitos de trazabilidad identificados en un entorno SoS. De esta forma, las entidades pueden registrarse y controlar a lo largo de la cadena de valor.

Para lograr este objetivo, este documento presenta un marco ágil enmarcado en el contexto de la Ingeniería de Sistemas de Sistemas.

Dentro de este marco metodológico, la etapa de descubrimiento, que tiene como objetivo lograr el conocimiento y comprensión del problema en un dominio determinado, proponer una solución y obtener una propuesta de solución validada por el usuario final; la etapa de desarrollo, cuyo objetivo es obtener resultados tangibles y validados de la solución propuesta; y la etapa de operación, cuyo objetivo es garantizar el correcto funcionamiento de la solución en ambientes de producción una vez implementada, monitoreando continuamente la interacción de las entidades con los sistemas, se unifican.

La unificación de las etapas de descubrimiento, desarrollo y operación implica una cobertura total de la conformación de la solución.

Cada una de las etapas del marco propuesto, a su vez, se componen de una serie de subetapas.

Al ser un marco metodológico ágil, se promueve la iteración continua entre sus etapas y dentro de cada una de las etapas se promueve la iteración continua entre las subetapas y se siguen una serie de principios y valores ágiles.

3.6 TRABAJO FUTURO

Este trabajo trata sobre el inicio de un trabajo de tesis doctoral. El objetivo final de la tesis doctoral es desarrollar una solución que garantice el control y seguimiento de las entidades que interactúan con los sistemas, de manera que el riesgo de error se reduzca a niveles mínimos, haciendo uso de las tecnologías de la información como eje central de la solución y aumentando la confianza de los implicados, todo ello en un entorno de SoS.

Las siguientes líneas de trabajo propuestas para ampliar esta investigación comienzan con mayor detalle en cada una de las etapas y subetapas de las que se compone el marco, validando el marco propuesto en un contexto real. Lo que implica, la realización de todas las tareas relacionadas con el descubrimiento, todas las tareas relacionadas con el desarrollo y todas las tareas relacionadas con la operación. Sin olvidar nunca que estamos en un contexto de Sistema de Sistemas donde se debe registrar y monitorear continuamente la trazabilidad de las entidades que interactúan con esta solución.

El contexto de la aplicación se enmarcará en el dominio comercial de una clínica de reproducción asistida, más concretamente, en un laboratorio de reproducción inteligente.

La reproducción asistida se ha convertido en un servicio clínico al que cada vez acceden más personas. Han proliferado problemas actuales como el retraso en la edad de paternidad, las opciones de padres solteros, etc. y los diferentes tratamientos que se ponen al servicio de la sociedad. Una parte fundamental de estos procesos radica en el trabajo de laboratorio.

Los laboratorios de reproducción humana están expuestos a múltiples incidentes, pero uno de los más graves es el que provoca la identificación incorrecta de muestras biológicas (óvulos, espermatozoides y embriones), siendo la identificación incorrecta un problema común en todos los ámbitos de la salud.

Un error de identificación ocurre cuando un paciente se asocia incorrectamente con una prueba, muestra, tratamiento o procedimiento y generalmente es causado por estrés, sobrecarga de trabajo, múltiples interrupciones, errores materiales, entre muchos factores.

Ciertos procesos, como la mezcla de óvulos y espermatozoides y la transferencia de embriones al útero, se consideran críticos, ya que representan “el

punto de no retorno". Si en los laboratorios de reproducción asistida se produce una identificación errónea, ésta puede pasar prácticamente desapercibida en cada una de las etapas del proceso que involucra a gametos y embriones. El resultado final será catastrófico para el paciente, el profesional y la clínica, con implicaciones legales que pueden derivar en sanciones e incluso, en casos extremos, en el cierre de la clínica.

Por todo ello, es necesario conocer la trazabilidad que siguen las muestras biológicas y las muestras de pacientes en todo momento dentro del laboratorio. En un laboratorio inteligente, la interrelación entre diferentes sistemas es fundamental, por lo que nos encontraremos en un entorno en el contexto de SoS.

A la hora de validar una propuesta, siempre debemos tener en cuenta que podemos encontrarnos con algunas amenazas. En este caso, la mayor amenaza a la que podemos enfrentarnos a la hora de validar el marco propuesto es a la hora de recopilar información para definir los requisitos de trazabilidad.

Como se comenta a lo largo del documento, los requisitos de trazabilidad son los que permitirán registrar y controlar en todo momento las entidades que interactúan con los diferentes sistemas. Actualmente, no es una práctica extendida entre los ingenieros de software recolectar información específica de este tipo y, por lo tanto, los clientes no están acostumbrados a brindar dicha información. No se sabe cuál es la mejor y más productiva forma de recopilar información de este tipo.

Por otro lado, se realizará un estudio exhaustivo para dar continuidad a este trabajo y aportar una clara explicación del estado actual de la trazabilidad en entornos de Sistemas de Sistemas.

Finalmente, una vez que nos aseguremos de que el marco es aplicable, se extenderá a otros contextos de aplicación.

3.7 AGRADECIMIENTOS

Este trabajo ha sido apoyado por el proyecto NICO (PID2019-105455GB-C31) financiado por el Ministerio de Economía y Competitividad de España; y las Ayudas para la formación de doctores en empresas "Doctorados Industriales" financiado por la Agencia Estatal de Investigación del Ministerio de ciencia, innovación y universidades.

3.8 REFERENCIAS

[1] Popper, S. W., Bankes, S. C., Callaway, R., & DeLaurentis, D. (2004). System of systems symposium: Report on a summer conversation. Potomac Institute for Policy Studies, Arlington, VA, 320.

[2] Ameri, F., Summers, J. D., Mocko, G. M., & Porter, M. (2008). Engineering design complexity: An experimental study of methods and measures. Res. Eng. Des, 19(2-3), 161-179.

[3] De Weck, O. L., Roos, D., & Magee, C. L. (2011). Engineering systems: Meeting human needs in a complex technological world. Mit Press.

[4] Luo, J., & Wood, K. L. (2017). The growing complexity in invention process. Research in Engineering Design, 28(4), 421-435.

[5] Boardman, J., & Sauser, B. (2006, April). System of Systems-the meaning of of. In 2006 IEEE/SMC International Conference on System of Systems Engineering (pp. 6-pp). IEEE.

[6] DeLaurentis, D. (2007, July). Role of humans in complexity of a system-of-systems. In International Conference on Digital Human Modeling (pp. 363-371). Springer, Berlin, Heidelberg.

[7] Eisner, H., Marciniak, J., & McMillan, R. (1991, October). Computer-aided system of systems (S2) engineering. In Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics (pp. 531-537). IEEE.

[8] Jamshidi, M. O. (2008). System of systems engineering-New challenges for the 21st century. IEEE Aerospace and Electronic Systems Magazine, 23(5), 4-19.

[9] Keating, C., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Safford, R., ... & Rabadi, G. (2003). System of systems engineering. Engineering Management Journal, 15(3), 36-45.

[10] Feng Tian, "An agri-food supply chain traceability system for China based on RFID & blockchain technology," 2016 13th International Conference on Service Systems and Service Management (ICSSSM), Kunming, 2016, pp. 1-6.

[11] Systems of Systems (SoS). (2019). https://www.sebokwiki.org/wiki/Systems_of_Systems_(SoS) (accedido junio 21, 2021)

[12] J. Dahmann & K. Baldwin, Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering, Montreal, Canada: IEEE Systems Conference, 7-10 April, 2008.

[13] Cureton, K. L., & Settles, F. S. (2005, October). Systems-of-systems architecting: educational findings and implications. In 2005 IEEE International Conference on Systems, Man and Cybernetics (Vol. 3, pp. 2726-2731). IEEE.

[14] Keating, C., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Safford, R., ... & Rabadi, G. (2003). System of systems engineering. Engineering Management Journal, 15(3), 36-45.

[15] DeLaurentis, D., Sindiy, O., & Stein, W. (2006). Developing Sustainable Space Exploration via System-of-Systems Approach. In Space 2006 (p. 7248). (accedido junio 21, 2021)

[16] TRAZABILIDAD. Según el Comité de Seguridad Alimentaria de AECOC. (2016) https://docplayer.es/9988201-Trazabilidad-segun-el-comite-de-seguridad-alimentaria-de-aecoc.html (accedido junio 21, 2021)

[17] Trazabilidad. (2017) https://trazabilidadalimentaria.blogspot.com/ (accedido junio 21, 2021)

[18] Nakhleh, R. E., Idowu, M. O., Souers, R. J., Meier, F. A., & Bekeris, L. G. (2011). Mislabeling of cases, specimens, blocks, and slides: a College of American Pathologists study of 136 institutions. Archives of pathology & laboratory medicine, 135(8), 969-974.

[19] MECALUX ESMENA, El rastro confuso de la trazabilidad. (2005) . https://www.mecalux.es/articulos-de-logistica/rastro-confuso-trazabilidad

[20] VITÓNICA, Trazabilidad y alertas alimentarias: qué ha podido salir mal en el reciente caso de listeriosis. (2019). https://www.vitonica.com/prevencion/trazabilidad-alertas-alimentarias-que-ha-podido-salir-mal-reciente-caso-listeriosis (accedido junio 21, 2021)

[21] Spriggs, M. (2003). IVF mixup: white couple have black babies. Journal of medical ethics, 29(2), 65-65.

[22] Dyer, O. (2002). Black twins are born to white parents after infertility treatment. BMJ, 325(7355), 64.

[23] BBC NEWS, Embryo mix-up at IVF hospital, (n.d.). http://news.bbc.co.uk/2/hi/health/2367705.stm (accedido junio 21, 2021).

[24] EL PAÍS, Holanda investiga la posible fecundación de 26 mujeres con esperma equivocado|Internacional,(n.d.).https://elpais.com/internacional/2016/12/29/actualidad/1483021366_741815.html (accedido junio 21, 2021).

[25] Morales Trujillo, L., Olivero Gonzales, M. A., Domínguez Mayo, F. J., García García, J. A., Mejías Risoto, M. (2020) A testability and observability framework to assure traceability requirements on system of systems. Journal of Web Engineering. 19, (2)

[26] INDRA, Trazabilidad en la cadena de valor de la Industria, 2018. https://www.minsait.com/es/actualidad/insights/trazabilidad-en-la-cadena-de-valor-de-la-industria (accedido junio 21, 2021)

[27] Radziwill, N. (2018). Blockchain revolution: How the technology behind Bitcoin is changing money, business, and the world. The Quality Management Journal, 25(1), 64-65.

[28] Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016, October). Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 254-269). ACM.

[29] Rojas, E. ¿Qué son los contratos inteligentes o "smart contracts"? (2019). Guía completa. https://es.cointelegraph.com/explained/what-is-a-smart-contract (accedido junio 21, 2021)

[30] A. Ramos, R.Wert. (2014). Design thinking en español, http://www.designthinking.es/inicio/ (accedido junio 21, 2021)

[31] O'REILLY RADAR, What is DevOps?. (2012). http://radar.oreilly.com/2012/06/what-is-devops.html (accedido junio 21, 2021)

[32] Dmitriy Samovskiy's Blog, The Rise of DevOps. (2010). http://www.somic.org/2010/03/02/the-rise-of-devops/ (accedido junio 21, 2021)

[33] John Willis, DevOps Culture (Part 1) - IT Revolution. IT Revolution. (2012). https://itrevolution.com/devops-culture-part-1/ (accedido junio 21, 2021).

[34] ATLASSIAN, Gitflow Workflow | Atlassian Git Tutorial. (2017). https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow (accedido junio 21, 2021)

[35] Driessen, V. (2010). A successful Git branching model. URL http://nvie. com/posts/a-successful-git-branching-model. (accedido junio 21, 2021)

[36] Gerber, A., & Craig, C. (2015). Introducing Git. In Learn Android Studio (pp. 145-187). Apress, Berkeley, CA.

[37] Janet Gregory, Lisa Crispin. Agile Testing Condensed: A Brief Introduction, ISBN-10:199922051X

[38] Pandit, P., & Tahiliani, S. (2015). AgileUAT: A framework for user acceptance testing based on user stories and acceptance criteria. International Journal of Computer Applications, 120(10).

[39] Gracia Bandrés, M.A., Gracia Murugarren, J., Romero San Martín, D. (2015) TecsMedia: Metodologías de diseño centradas en usuarios

[40] Knapp, J. (2012). The Design Sprint.

[41] Banfield, R., Lombardo, C. T., & Wax, T. (2015). Design sprint: A practical guidebook for building great digital products. " O'Reilly Media, Inc.".

[42] Olivero, Miguel Angel; Morales-Trujillo, L; Domínguez-Mayo, F. J.; Mejías, M; ,Systematic Development of ERP Modules using a Model-Driven Strategy Focusing on the Users,4th International Special Session on Advances Practices in Model-Driven Web Engineering in the 15th International Conference on Web Information Systems and Technologies, 2019

[43] Agile Manifesto. https://agilemanifesto.org/. (accedido junio 21, 2021)

[44] IMF Business School, Metodologías ágiles de desarrollo. https://blogs.imf-formacion.com/blog/tecnologia/metodologias-agiles-de-desarrollo-201801/ (accedido junio 21, 2021)

[45] Schaefer, A., Reichenbach, M., & Fey, D. (2013). Continuous integration and automation for DevOps. In IAENG Transactions on Engineering Technologies (pp. 345-358). Springer, Dordrecht.

[46] De Bayser, M., Azevedo, L. G., & Cerqueira, R. (2015, May). ResearchOps: The case for DevOps in scientific applications. In 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM) (pp. 1398-1404). IEEE.

[47] Erich, F. M. A., Amrit, C., & Daneva, M. (2017). A qualitative study of DevOps usage in practice. Journal of Software: Evolution and Process, 29(6), e1885.

[48] Virmani, M. (2015, May). Understanding DevOps & bridging the gap from continuous integration to continuous delivery. In Fifth International Conference on the Innovative Computing Technology (INTECH 2015) (pp. 78-82). IEEE.

[49] Gotel, O. C., & Finkelstein, C. W. (1994, April). An analysis of the requirements traceability problem. In Proceedings of IEEE International Conference on Requirements Engineering (pp. 94-101). IEEE.

[50] Ramesh, B., & Jarke, M. (2001). Toward reference models for requirements traceability. IEEE transactions on software engineering, 27(1), 58-93.

[51] Pohl, K. (2010). Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated.

[52] Mittal, S., & Martín, J. L. R. (2018). Netcentric system of systems engineering with DEVS unified process. CRC Press.

[53] Olivero, Miguel Angel, et al. "Security assessment of systems of systems." 2019 IEEE/ACM 7th International Workshop on Software Engineering for Systems-of-Systems (SESoS) and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (WDES). IEEE, 2019