

14

LA CALIDAD DEL SOFTWARE COMO MECANISMO DE ÉXITO EN PROYECTOS MULTIDISCIPLINARES: PROYECTO IMEDEA Y MEET2CARE

*Julián A. García García
Cristina Ramírez de Verger-Ruano
Nicolás Sánchez-Gómez*

Cada vez son más los proyectos en los que colaboran instituciones y personas con diferentes ámbitos de conocimiento y especialidad para desarrollar productos digitales de calidad. Aportar la experiencia que cada miembro tiene en su especialidad, permite alinear expectativas siempre y cuando incluyamos mecanismos de aseguramiento de la calidad en fases tempranas del desarrollo de los proyectos. Las metodologías ágiles son de práctica aplicación habitual y común en este tipo de proyectos multidisciplinarios, en los cuales, suelen identificarse tres grandes fases: Fase de ideación (Inception phase), Fase de planificación (Planning phase) y Fase de Implementación (Implementation phase). Sin embargo, los mecanismos de control de la calidad suelen aplicarse en la fase de implementación, dejando las primeras fases aisladas de este control. En este trabajo se presenta, de una manera práctica ilustrada por dos proyectos reales: iMedea y Meet2Care, cómo la incorporación de mecanismos de aseguramiento de calidad temprano que involucre a los diferentes stakeholders del equipo ya desde la fase de ideación, es un aspecto clave para mejorar el éxito de los proyectos. En los proyectos que se presentan en este capítulo se ofrece una propuesta, que hemos aplicado con éxito en estos proyectos, en los que la aplicación de técnicas de aseguramiento de la calidad se aplica de manera temprana e involucran a todos los diferentes stakeholders. Los proyectos presentados en este capítulo se focalizan en el entorno sanitario y muestran cómo la aplicación temprana de técnicas de aseguramiento de la calidad.

14.1 INTRODUCCIÓN

Las innovaciones tecnológicas surgen constantemente y se vuelven cada vez más importantes en la actual sociedad de la información. De hecho, la aplicación generalizada de Internet y las Tecnologías de la Información y las Comunicaciones (TIC) ha aumentado significativamente nuestra dependencia de los sistemas software en todos los ámbitos de la sociedad y los negocios [13] [17]; desde la salud, la educación, el transporte, o la logística, entre muchos otros, e incluso áreas de entretenimiento como deportes, películas o videojuegos. Esta dependencia genera preocupaciones críticas sobre, entre otros aspectos, la confiabilidad, estabilidad y seguridad del software [7] [11] debido a que cualquier error puede tener consecuencias desastrosas.

Para mitigar estos riesgos surge el concepto de Aseguramiento de la Calidad del Software o Software Quality Assurance (SQA) que aglutina una serie de técnicas, herramientas, y métodos, para ayudar a monitorizar y gestionar los procesos de ingeniería del software utilizados para garantizar la calidad durante el proceso de desarrollo software [9]. El término gestión de la calidad tiene un significado específico en muchos sectores empresariales, pero de manera común no solo se centra en la calidad de los productos, procesos y servicios de una organización, sino también en los medios para lograrla. En el ámbito de la ingeniería del software, cuando hablamos de calidad nos referimos al conjunto de cualidades del software que determinan su utilidad y existencia desde el punto de vista de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad [14]. De esta manera, la obtención de un software con calidad implica la utilización de procesos que permitan controlar el ciclo de desarrollo de software y uniformar la filosofía de trabajo [19], en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el aseguramiento, control y mejora de la calidad del software.

En este contexto, el trabajo del equipo de SQA de un proyecto debe estar encaminado a: (i) participar en la definición de los planes, procesos, procedimientos y estándares para asegurar que se ajustan a las necesidades del mismo y que pueden ser usados para realizar las evaluaciones de QA y cumplir los requisitos del proyecto y también las políticas de la organización; (ii) revisar la documentación técnica elaborada inicialmente por el equipo de desarrollo, y comprobar que está completa y alineada con las necesidades formuladas inicialmente por el área usuaria, y que es correcta desde el punto de vista metodológico.

Para alcanzar estos objetivos y gestionar de manera la calidad de proyectos existen guías de buenas prácticas en dirección de proyectos como, entre otros,

PMBOK (Project Management Body of Knowledge) [4] y PRINCE2 [2], así como estándares como la ISO 9001 [12]. Todos ellos contemplan la gestión de la calidad software como un proceso global que a su vez engloba, principalmente, la gestión de tres componentes o procesos: (i) planificar la gestión de la calidad, en el cual se define el plan de calidad del proyecto donde se establecen, entre otros aspectos, los requisitos y normas de calidad que se debe seguir durante el proyecto software, así como las métricas e indicadores que van a permitir medir y controlar el desempeño de la calidad; (ii) realizar el aseguramiento de la calidad, el cual consiste, a groso modo, en medir los aspectos de la calidad definidos en la etapa anterior; (iii) controlar la calidad, que consiste en realizar un análisis de los datos e indicadores medidos (lo cual puede propiciar actualizaciones del plan de calidad en un proceso iterativo de mejora continua).

En esta línea, la calidad de un producto software se podría medir después de desarrollar dicho producto, pero en el caso de detectar problemas o errores derivados de imperfecciones en los requisitos y/o en el análisis, la eliminación de estos problemas podría acarrear grandes costes, por lo que es imprescindible tener en cuenta, a lo largo de todo el ciclo de vida de desarrollo del software, tanto las especificaciones de calidad software como su control [15], así como tener en cuenta la participación de usuarios finales desde etapas tempranas para identificar incongruencias o errores lo antes posible.

Sin embargo, no es común involucrar al cliente y usuarios finales en el proceso de aseguramiento de la calidad del proyecto software desde el principio del proceso de desarrollo software, a pesar de las ventajas que puede traer hacerlo en términos de calidad y aceptación del producto software final. De hecho, trabajar con equipos multidisciplinares y experimentados aporta grandes ventajas. Por ejemplo, trabajar en equipo con el cliente, y aportando la experiencia que cada miembro tiene en su especialidad hace que el desarrollo del proyecto sea ágil y con un resultado eficiente ya que permite identificar errores o incongruencias desde etapas tempranas (no sólo en el código fuente sino también en el propio análisis y diseño del producto). Todo ello repercute en una reducción de tiempos de desarrollo y una reducción de errores lo que, en última instancia, repercute en una mayor calidad del producto software desarrollado, así como en una mejor aceptación del mismo por parte del cliente y usuarios finales.

La situación previa es sumamente importante en sectores de negocio críticos donde el aseguramiento de la calidad software, la confiabilidad, la estabilidad y la seguridad de los sistemas tienen que ser características clave. Entre estos sectores se podría situar el sector las organizaciones sanitarias, en el cual se manejan datos personales sensibles y en el que intervienen una gran cantidad de interesados o stakeholders: desde personal de administración y servicios hasta el profesional

sociosanitario (en sus diferentes vertientes y especialidades). Tradicionalmente y desde la perspectiva del usuario final, el aseguramiento de la calidad de un producto software se ha realizado en base a iteraciones de ciclos pruebas (en etapas finales del proyecto) sobre la primera versión estable de dicho software. Aunque actualmente existen herramientas que permiten mejorar la productividad de las pruebas y reducir errores humanos a partir de su automatización, el coste de esta etapa del desarrollo software sigue resultando una actividad intensiva en dedicación y recursos. No sólo por parte del equipo de desarrollo, sino también por parte del área funcional y usuaria.

Para mejorar estos aspectos, en este artículo proponemos un marco de trabajo para mejorar el aseguramiento de la calidad desde etapas tempranas del desarrollo software, a través de la involucración del usuario final y el área funcional del cliente desde la etapa de conceptualización y definición del producto software, hasta la etapa de desarrollo, producción y posterior mantenimiento. Con este planteamiento, conseguimos detectar a tiempo los posibles defectos de un producto software incrementando su calidad mediante la detección eficaz de errores en fases tempranas del ciclo de vida de desarrollo, disminuyendo a su vez el impacto y coste que tendría sobre el desarrollo si estos errores se identificaran con posterioridad.

En este artículo, presentamos, además, dos proyectos de innovación tecnológica donde nuestra propuesta ha sido aplicada y validada: el proyecto iMEDEA y el proyecto Meet2Care¹². Ambos proyectos han sido realizados por la empresa tecnológica G7innovation¹³ en colaboración con el grupo de investigación Ingeniería Web y Testing Temprano (IWT2) de la Universidad de Sevilla. Cabe mencionar que los resultados obtenidos han propiciado que G7innovation asuma la propuesta aquí presentada (para el aseguramiento de la calidad software desde etapas tempranas) como método de trabajo en sus proyectos de innovación tecnológica.

Finalmente, después de esta sección introductoria, el resto del artículo ha sido estructurado de la siguiente manera. La Sección 14.2 describe el marco de trabajo que hemos propuesto para mejorar el proceso de aseguramiento de la calidad software desde etapas tempranas del ciclo de desarrollo software, involucrando a usuarios finales desde etapas iniciales. La Sección 14.3 describe los proyectos empresariales iMEDEA y Meet2Care de la empresa G7innovation y cómo nuestra propuesta ha sido instanciada en dichos proyectos. La Sección 14.4 recopila una serie de lecciones aprendidas y, finalmente, la Sección 14.5 presenta conclusiones y trabajos futuros de nuestra propuesta.

12 Meet2Care. Sitio web: <https://meet2care.com/>

13 G7innovation. Sitio web: <https://g7innovation.com/>

14.2 MARCO DE TRABAJO PARA EL ASEGURAMIENTO DE LA CALIDAD SOFTWARE DESDE ETAPAS TEMPRANAS

El marco de trabajo que se propone para el aseguramiento de la calidad software está basado en un proceso completo que se inicia desde etapas tempranas del ciclo de vida del desarrollo software. Uno de los pilares de este marco de trabajo es fomentar la puesta en marcha de herramientas de soporte al desarrollo y pruebas software, la utilización de un entorno de integración y mejora continua, a través de la automatización de tareas y la optimización de pruebas software. Todo ello, con el propósito de garantizar la calidad durante todo el ciclo de desarrollo del software.

Las actividades de producción de software que se propone están agrupadas en tres grandes bloques o etapas: (i) descubrimiento del producto software; (ii) desarrollo del producto software; y (iii) operación del producto software. Para cada una de estas fases, se establecen una serie de técnicas y/o métodos para llevarla a cabo, así como los artefactos resultantes y qué roles participan en cada una de estas etapas.

La Figura 14.1 muestra el mapa del proceso propuesto para el aseguramiento de la calidad software desde etapas tempranas.



Figura 14.1. Mapa del proceso para el aseguramiento de la calidad software desde etapas tempranas

A continuación, se describen cada una de las etapas y actividades que se muestran en la Figura 14.1.

Por una parte, proponemos en primer lugar una *etapa de descubrimiento* (*etapa DISCOVERY; c.f. Figura 14.1*). En esta etapa proponemos aplicar metodologías como Design Sprint [1] y Design Thinking [16] de manera conjunta entre el equipo técnico y el área usuaria y funcional futura del producto software. Con este planteamiento se consigue resolver problemas complejos a través de la creación de innovaciones y el desarrollo de nuevas ideas, todo ello gracias a la validación de propuestas con los usuarios finales. El objetivo de esta etapa es: (1) comprender y conocer el contexto de la funcionalidad demandada por los usuarios, así como saber cuáles son las necesidades y motivaciones; (2) plantear ideas de solución sin importar la calidad (teniendo en cuenta la divergencia), siendo multidisciplinar con el objetivo

de tener diferentes enfoque; (3) decidir y tomar decisiones para consensuar ideas en base a la información recabada anteriormente; (4) prototipar la solución; y (5) validar el prototipo con el área usuaria.

En segundo lugar, planteamos una *etapa completa de desarrollo de productos software (etapa DEVELOPMENT; c.f. Figura 14.1)*, la cual se basa a su vez en tres actividades básicas:

1. **Diseñar (DESIGN; c.f. Figura 14.1)**. Esta actividad consiste en el diseño técnico de la solución, para lo cual proponemos el uso del marco metodológico NDT (Navigational Development Techniques) [6].

Esta metodología propone un marco de trabajo para definir y modelar los diferentes tipos de requisitos de la solución (requisitos de almacenamiento, requisitos funcionales, actores, requisitos no funcionales, etc.), diferentes diagramas para establecer el comportamiento dinámico de la solución software, y diferentes modelos para decidir casos de prueba y la arquitectura tecnológica. Todo ello está basado en estándares como UML (Unified Modeling Language) [8] e IFML (Interaction Flow Modeling Language) [3]; este último permite modelar los flujos de interacción y establecer los requisitos de navegación que será necesario implementar.

2. **Construir (CONSTRUCTION; c.f. Figura 14.1)**. Esta actividad consiste en la construcción o implementación propiamente dicho de la solución software siguiendo los requisitos aprobados y teniendo en cuenta las directrices y buenas prácticas (uso de patrones de diseño, modularidad, etc.).

En este sentido, se plantea la verificación de calidad de código mediante la ejecución de pruebas estáticas, como las pruebas unitarias que permitan verificar el correcto funcionamiento de cada componente software (clase, método, interfaz, etc.). Todo ello con el objetivo de detectar problemas de diseño, duplicidad de código, detección de vulnerabilidades, seguimiento de estándares de codificación y monitorización de cobertura de las pruebas (para verificar que la cobertura de estos tests es aceptable).

3. **Verificar y validar (VER&VAR; c.f. Figura 14.1)**. Por último, en la etapa se establece la actividad de verificación y validación teniendo en cuenta la política de testing de cada organización con el propósito de comprobar si el software desplegado cumple con las especificaciones y expectativas del usuario.

Al objeto de garantizar la calidad software, nuestro marco de trabajo propone, al menos, abordar los siguientes tipos de pruebas dinámicas: (1) pruebas de integración, con las que verificar que los componentes de

la aplicación funcionan correctamente actuando en conjunto; (2) pruebas funcionales (éstas son manuales, al principio, y automatizadas, cuando el producto software esté en fase estable), con las que validar que el software funciona correctamente y, sobre todo, que el producto hace lo que se ha especificado; (3) pruebas de rendimiento, con las cuales se estresa al producto software para determinar su capacidad de respuesta y de rendimiento, la confiabilidad y/o la escalabilidad del sistema, bajo una carga de trabajo determinada y, pruebas de estrés sobre el sistema, sometiéndolo a condiciones extremas, tanto a nivel de datos máximos o una gran cantidad de usuarios concurrentes; (4) pruebas de seguridad, al objeto de identificar posibles fallos y debilidades; y (5) otras pruebas según necesidad (accesibilidad, usabilidad, etc.).

Por último, nuestro marco de trabajo plantea una *etapa de operación (etapa OPERATION; c.f. Figura 14.1)*, la cual se decide bajo el marco de referencia ITIL [18]. Resulta importante remarcar que, en el entorno de producción, es fundamental realizar un proceso de seguimiento continuo (monitorización) del producto desde diferentes perspectivas: rendimiento (tiempos de respuesta del producto); infraestructura (consumo de memoria, espacio en disco,); y gestión de incidencias del área usuaria o del área técnica, adelantándose de forma proactiva a posibles incidencias.

14.3 CASOS REALES DE APLICACIÓN Y VALIDACIÓN

Esta sección tiene como objetivo mostrar cómo ha sido aplicado el marco de trabajo presentado en la sección anterior dentro de los proyectos reales: iMedea y Meet2Care.

14.3.1 Proyecto iMedea

La empresa G7innovation llevó a cabo el proyecto iMEDEA¹⁴ (innovative MEDical Engineering Assistance) desde 2017 hasta 2020 con la colaboración del Grupo IWT2. Este proyecto nace por la necesidad que tenía la clínica de reproducción asistida Inebir¹⁵ (principal cliente de G7innovation) para disponer de un sistema de información robusto y transversal que permitiese gestionar y explotar

14 iMEDEA: Website: <https://g7innovation.com/producto/imedea/>

15 Inebir. Website: <https://inebir.com/>

toda la información asistencial relacionada con tratamientos de fecundación asistida (denominados frecuentemente procesos ART (Assisted Reproduction Treatment) [20], junto con la gestión empresarial de la entidad.

Hasta el momento, la clínica Inebir utilizaba diferentes soluciones tecnológicas para llevar a cabo su actividad de negocio (tanto asistencial como empresarial). Por una parte, toda la actividad asistencial de sus pacientes y soporte a procesos ART estaba basada en la herramienta Sara Plus; actualmente descatalogada y sin soporte desde 2017 que fue desarrollada por la empresa farmacéutica Merck Serono¹⁶. Sara Plus sigue una arquitectura cliente-servidor [10] y presentaba limitaciones técnicas y funcionales tales como falta de control de datos y trazabilidad de éstos, fatal control de accesos según roles o responsabilidades de los usuarios del sistema y falta de gestión eficiente de muestras biológicas, entre otros aspectos. Por otra parte, la actividad empresarial (facturación, recursos humanos, contabilidad, etc.), e incluso la gestión y control de bancos de criopreservación de muestras biológicas, estaba principalmente soportada por herramientas de escritorio ofimáticas y de contabilidad.

Este ecosistema de herramientas propiciaba información redundante, habiendo propensión a errores por transcribir información de manera manual y problemas de trazabilidad de información. Esta situación viene provocada por que todas las herramientas estaban desconectadas unas de otras y eran utilizadas por diferentes perfiles de usuario en plataformas diferentes. La reparación y consolidación de esta información implicaba en la entidad grandes esfuerzos en términos de dedicación de personal.

En consecuencia, la creciente necesidad por gestionar y mantener información asistencial y empresarial de manera sincronizada propició el inicio del proyecto iMEDEA.

En el proyecto iMEDEA se desarrolló una plataforma innovadora para optimizar y facilitar el día a día de los profesionales sanitarios del ámbito de la reproducción asistida. Esta plataforma se plantea como una plataforma web y personalizada, donde la historia clínica electrónica del paciente es el Core de la plataforma y la herramienta fundamental en la actividad clínica diaria, que permite la gestión integral avanzada de las unidades básicas de reproducción humana, junto con la gestión empresarial de la clínica. La plataforma iMEDEA (c.f. Figura 14.2) sigue una arquitectura modular sustentada sobre una solución ERP (Enterprise Resource Planning) open source llamada Odoo¹⁷ [5]. Sustentar la funcionalidad

16 Merck Serono. Website: <https://www.merckgroup.com/es-es>

17 ERP Odoo. Website: <https://www.odoo.com/>

de iMEDEA sobre la arquitectura Odoos ha propiciado que la solución iMEDEA admita configuraciones avanzadas tales como opciones de multiempresa, seguridad, multidioma y multidispositivo sobre una única instancia de iMEDEA. Todo ello, junto con una arquitectura modular, permite implantar instancias de iMEDEA de manera flexible y adaptable a las necesidades de cada entidad sanitaria.

En este contexto, las necesidades del cliente han derivado en un proyecto software con 135 requisitos de almacenamiento de información, 8 requisitos de actores, 12 requisitos no funcionales y 257 requisitos funcionales. Estos requisitos, además, están agrupados en 14 módulos funcionales: (1) módulo de gestión de pacientes, cuyo propósito es gestionar toda la información demográfica de pacientes de la clínica; (2) módulo de gestión de prescripciones y tratamientos médicos, cuyo propósito es gestionar toda la información sobre tratamientos farmacológicos prescritos a los pacientes; (3) módulo de gestión de pruebas diagnósticas, cuyo propósito es la gestión de estudios complementarios y pruebas diagnósticas de laboratorio, de imagen, de genética, de anestesia, etc.; (4) módulo de gestión de episodios clínicos, anamnesis y exploraciones físicas, a través del cual el profesional sanitario indica el motivo de consulta y el juicio clínico después de explorar y estudiar el historial del paciente; (5) módulo de historia clínica de ginecología, andrología y reproducción; (6) módulo de agendas de unidades y de especialistas; (7) módulo de gestión de citas presenciales y por videoconferencia; (8) módulo de gestión y control de tratamientos de fecundación; (9) módulo de estación clínica junto con un repositorio documental de informes clínicos; (10) módulo de gestión de muestras biológicas y bancos de criopreservación; (11) módulo de gestión de donantes propios y externos; (12) módulo de cuadro de mando de indicadores y calidad de muestras; (13) módulo de gestión de recursos humanos e inventario; y (14) módulo financiero y contable.



Figura 14.2. Arquitectura y componentes funcionales de iMEDEA

Desde el punto de vista la ingeniería de software, el proyecto iMEDEA ha sido un proyecto complejo y crítico dado el entorno en el que se contextualiza y considerando el gran número de participantes y profesionales de diferentes áreas (desde profesionales sanitarios hasta profesionales de administración, finanzas y contabilidad, entre otros). Debido a limitaciones de espacio, en esta sección vamos a centrarnos en uno de los principales módulos funcionales de iMEDEA: *el módulo de gestión de reproducción o tratamientos de fecundación*. En este sentido, se describe cómo hemos aplicado y validado nuestro método de trabajo para asegurar la calidad en el diseño y desarrollo de este módulo.

Por su naturaleza, el proyecto iMEDEA (y en particular su módulo de gestión de tratamientos de fecundación) es un claro ejemplo de proyecto multidisciplinar. Durante sus más de 3 años de ejecución del proyecto iMEDEA, el diseño y desarrollo de este módulo se llevó a cabo durante la segunda mitad del proyecto. Durante esta etapa, participaron directamente 2 jefes de proyecto, 1 ingeniero de sistemas, 1 analista funcional y 5 analistas-programadores junto con 9 usuarios finales responsables de diferentes áreas de la clínica: área de atención al paciente, área de enfermería, área de ginecología y andrología, área de reproducción y laboratorio, área de genética, área de contabilidad/finanzas y área de recursos humanos.

14.3.2 Proyecto Meet2Care

La telemedicina no es algo nuevo. La tecnología en el campo de la salud lleva décadas avanzando a un ritmo vertiginoso, y la Telemedicina como una de sus ramas más importantes, no se ha quedado atrás. Pero en esto, como en otros muchos aspectos de la sociedad, ha existido siempre un gran hándicap, la resistencia al cambio. La falta de formación en profesionales, posibles dudas en cuestiones legales o simplemente el miedo a la violabilidad de los datos clínicos han hecho que tanto profesionales como pacientes tuvieran ciertas reticencias a la hora de la implantación y uso de sistemas de ehealth en sus organizaciones. Sin embargo, ha cambiado radicalmente. Desde finales del año 2020 hemos sido testigos de un punto de inflexión como consecuencia de la crisis sanitaria mundial provocada por el **Covid-19**. La Telemedicina ha pasado de ser una opción a ser una necesidad.

Dentro de la TeleMedicina una de las actividades más demandadas durante la crisis del Covid 19 ha sido la **TeleConsulta**. Es decir, la asistencia remota de un profesional sanitario a un paciente mediante sistemas de videoconferencia en tiempo real. En un tiempo récord, especialidades clínicas en las que apenas era utilizada esta modalidad de asistencia, han tenido que adaptarse tanto a nivel tecnológico como a nivel asistencial a las nuevas circunstancias. Esta aceleración también ha supuesto

un aumento en los riesgos y posibilidades de bloqueo en su implantación. Había que hacerlo rápido pero había que hacerlo bien.

En este sentido, dentro del desarrollo de este tipo de aplicaciones y viviendo el contexto actual, es vital la participación, desde un primer momento, de un equipo multidisciplinar formado por agentes de diversas áreas funcionales dentro del sector tecnológico y sanitario.

La aplicación Meet2Care es un claro ejemplo de solución software donde la visión tecnológica y la visión asistencial y clínica han ido de la mano desde su origen poniendo de manifiesto el éxito de estos equipos en el uso de **metodologías ágiles** de desarrollo de software. La empresa G7innovation vió la necesidad de ampliar y complementar su portfolio de soluciones eHealth con una herramienta de videoconsulta, que diera respuesta tanto a profesionales de la salud como a los propios pacientes a las necesidades contextuales, pero donde la experiencia del profesional y la experiencia del paciente se midieran al mismo nivel que la solución tecnológica.

Para ello se conformó un grupo de expertos procedentes de la propia empresa, del Grupo de Investigación (Ingeniería Web y Testing Temprano) de la Universidad de Sevilla (US) y de la Clínica Inebir: especialistas médicos; especialistas de atención al usuario; especialistas en design thinking; analistas funcionales; analistas programadores; ingenieros de sistemas; diseñadores gráficos; investigadores y alumnos de doctorado de la us; y también a los propios pacientes como usuarios finales.

Con la total implicación de este equipo y aplicando metodologías ágiles durante todo el ciclo de desarrollo software, el resultado ha sido una aplicación de Teleconsulta con 4 pilares fundamentales:

- **Interoperabilidad:** Total capacidad de integración segura basada en API Rest segura dentro de entornos informáticos heterogéneos.
- **Accesibilidad:** Funcionamiento multiplataforma compatible con los principales sistemas operativos más populares (Android, iOS, MS Windows y Linux). Accesible desde cualquier dispositivo, PC, móvil o tablet.
- **Seguridad y Confidencialidad:** Se ha vigilado el cumplimiento exhaustivo de los requisitos y normativas mundiales más exigentes de seguridad en el ámbito del cuidado de la salud como por ejemplo la Normativa GDPR (General Data Protection Regulation) de la Comunidad Europea, Normativa HIPAA (Health Insurance Portability and Accountability Act) de Estados Unidos de América y Normativa

PIPEDA (Personal Information Protection and Electronic Documents Act) de Canadá. Todas las comunicaciones entre profesionales y pacientes están encriptadas, guardando los datos de salud de forma segura y Todas las comunicaciones entre profesionales y pacientes están encriptadas, guardando los datos de salud de forma segura

- **Usabilidad:** Desde una misma aplicación el profesional sanitario tiene la posibilidad de configurar su agenda, acceder a las historias clínicas de los pacientes, realizar las video-consultas y tener una comunicación ágil con el paciente a través de chats y mensajería. Por su parte, el personal administrativo, puede gestionar las agendas de los profesionales, las citas y datos administrativos del paciente y controlar toda la gestión administrativa de la clínica. Por último, y no menos importante, el paciente, puede acceder a la aplicación desde cualquier dispositivo, pedir cita online para el especialista médico, gestionar sus citas y datos personales, comunicarse fluidamente tanto con el personal administrativo como con el facultativo y acceder a la video-consulta online de una manera ágil y cómoda.

14.4 LECCIONES APRENDIDAS

En un sector tan crítico como es el ámbito clínico se hace imprescindible la participación de expertos desde las fases iniciales del desarrollo de productos software especializados en Salud.

Para garantizar la calidad del software en este tipo de Proyectos es básico la detección eficaz de errores en fases tempranas de los mismos, y para ello se ha comprobado que es imprescindible la participación del usuario final y experto a lo largo de todo el ciclo de vida del desarrollo de un Proyecto.

Una de las principales consecuencias de la aplicación de estas metodologías, ha sido la participación del Equipo multidisciplinar en la definición de Requisitos, ya que imprecisiones en dicha fase, hubieran supuesto graves problemas posteriormente. También ha sido muy importante la comprobación periódica de que la solución propuesta a través del análisis y el diseño no había obviado ni malinterpretado ninguno de estos requerimientos. Y para todo esto es básica la implicación de expertos en la materia a lo largo de todo el proceso.

El hecho de que expertos en diferentes materias tales como Médicos, Técnicos de laboratorio, o Gerentes sanitarios hayan tomado la iniciativa en estos desarrollos y hagan los Proyectos Software algo suyo, liderando su diseño desde el principio, es una garantía de éxito asegurado y un gran avance en el Desarrollo Software.

14.5 CONCLUSIONES

Las tecnologías de la información y comunicaciones han mejorado significativamente la productividad, seguridad, eficiencia y eficacia de empresas de todos los ámbitos de negocio. En el ámbito de los sistemas de información orientados al ámbito empresarial, los parámetros antes mencionados son más importantes si cabe en sectores críticos como el de la salud, la logística y la energía, entre muchos otros. En estos y otros ámbitos, es esencial llevar a cabo procesos que garanticen y aseguren la calidad del software sobre todo porque son proyectos en los que participan y colaboran grandes equipos de profesionales con diferentes ámbitos de conocimiento y especialidad.

En este capítulo, proponemos un marco de trabajo para el aseguramiento de la calidad desde etapas tempranas del desarrollo de proyectos software en los que participan equipos de *stakeholders* multidisciplinares, quienes aportan experiencia en su especialidad. El marco de trabajo propuesto permite alinear expectativas de cada *stakeholder* incluyendo mecanismos de aseguramiento de la calidad en fases tempranas del desarrollo del proyecto. Para ello se utilizan metodologías ágiles y técnicas corporativas de conceptualización e ideación de la solución software a desarrollar. Una vez presentado este marco de trabajo, el capítulo describe la instanciación de este marco de trabajo en dos proyectos software reales del ámbito sanitario (el proyecto iMedea y el proyecto Meet2Care) llevados a cabo por la empresa tecnológica G7innovation. La aplicación temprana de técnicas de aseguramiento de la calidad en estos proyectos nos ha permitido involucrar a todos los diferentes *stakeholders*, consiguiendo con ello expertos en diferentes materias tales como ingeniero software, médicos, técnicos de laboratorio, o gerentes sanitarios hayan tomado la iniciativa en estos desarrollos y hagan los proyectos software algo suyo, liderando su diseño desde el principio, es una garantía de éxito asegurado y un gran avance en el desarrollo software.

Finalmente, como trabajos futuros planteamos la validación de nuestro marco de trabajo en proyectos software de otros ámbitos de negocio, así como el diseño y desarrollo de herramientas de soporte que permitan autorizar y sistematizar la ejecución del proceso de aseguramiento de la calidad aquí planteado

14.6 AGRADECIMIENTOS

Este artículo de investigación ha sido elaborado dentro del proyecto NICO (PID2019-105455GB-C31), el cual está financiado por el Ministerio de Economía y Competitividad del Gobierno de España.

14.7 REFERENCIAS

- [1] Banfield, R., Lombardo, C. T., & Wax, T. (2015). Design sprint: A practical guidebook for building great digital products. “ O’Reilly Media, Inc.”.
- [2] Bentley, C. (2012). Prince2: a practical handbook. Routledge.
- [3] Brambilla, M., & Fraternali, P. (2015). Interaction Flow Modeling Language. The MK/OMG Press. Morgan Kaufmann.
- [4] Forero, R. S., Cubillos, D. B. P., & Rojas, G. E. C. (2015). Lineamientos de la gestión de la calidad del PMBOK aplicados al teletrabajo. *Tecnología investigación y Academia*, 3(2), 57-62.
- [5] Ganesh, A., Shanil, K. N., Sunitha, C., & Midhundas, A. M. (2016, February). OpenERP/Odoo-an open source concept to ERP Solution. In 2016 IEEE 6th International Conference on Advanced Computing (IACC) (pp. 112-116). IEEE.
- [6] García-García, J. A., Ortega, M. A., García-Borgoñon, L., & Escalona, M. J. (2012, July). NDT-Suite: a model-based suite for the application of NDT. In International Conference on Web Engineering (pp. 469-472). Springer, Berlin, Heidelberg.
- [7] Ghobadi, A., Karimi, R., Heidari, F., & Samadi, M. (2014, February). Cloud computing, reliability and security issue. In 16th International Conference on Advanced Communication Technology (pp. 504-511). IEEE.
- [8] Jacobson, L., & Booch, J. R. G. (2021). The unified modeling language reference manual.
- [9] Juran, J. M., & De Feo, J. A. (2010). Juran’s quality handbook: the complete guide to performance excellence. McGraw-Hill Education.
- [10] Hanson, M. D. (2000). The client/server architecture. *Server Management*, 3.
- [11] Karpinski, M., Ziubina, R., Azatov, A., Shaikhanova, A., Teliushchenko, V., & Falat, P. (2020, September). Evaluation of Information Security Software Using Quality and Reliability Criteria. In 2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS) (pp. 1-5). IEEE.

- [12] Martínez, J. A. G. (2015). Guía para la aplicación de UNE-EN ISO 9001: 2015. AENOR.
- [13] Masuda, Y. (2020). Image of the future information society. In *The information society reader* (pp. 15-20). Routledge.
- [14] Naik, K., & Tripathy, P. (2011). *Software testing and quality assurance: theory and practice*. John Wiley & Sons.
- [15] Oktaba, H. (2002): “Procesos de Desarrollo de Software”, Primer Taller Internacional de Calidad en desarrollo de software, México.
- [16] Serrano, M., & Blázquez, P. (2016). *Design thinking*. Pozuelo de Alarcón: ESIC.
- [17] Simion, E., & Géraud-Stewart, R. (2020). *Innovative Security Solutions for Information Technology and Communications*. Springer International Publishing.
- [18] Van Bon, J., De Jong, A., Kolthof, A., Pieper, M., Tjassing, R., Van der Veen, A., & Verheijen, T. (2008). *Fundamentos de ITIL®* (Vol. 3). Van Haren.
- [19] Wagner, S. (2013). *Software product quality control*. Springer. ISBN 978-3-642-38570-4.
- [20] Zegers-Hochschild, F., Adamson, G. D., de Mouzon, J., Ishihara, O., Mansour, R., Nygren, K., ... & Van der Poel, S. (2009). The international committee for monitoring assisted reproductive technology (ICMART) and the world health organization (WHO) revised glossary on ART terminology, 2009. *Human reproduction*, 24(11), 2683-2687.