Sitio Web Para la Gestión de Inventario de la Microempresa de Calzado Enduro

Autor(es): Andres Jaimes

1093589017

Arley Castellanos

1098150140

Jerley Hernández

1091965952

Segundo entregable del proyecto

Universidad de Pamplona

Facultad de Ingenierías y Arquitectura

Ingeniería de Sistemas

2024

Villa del Rosario – Norte de Santander

# ¿Cuáles son los factores de éxito y de fracaso de los proyectos de software?

El éxito o fracaso de un proyecto de software depende de una compleja interacción de factores, tanto técnicos como humanos. Un aspecto crucial es la estimación precisa del esfuerzo, especialmente en metodologías ágiles como Scrum. Como señala la investigación de (Iqbal et al., 2024), problemas como la inconsistencia en las historias de usuario, complejidades técnicas y dinámicas de equipo pueden afectar significativamente la precisión de la estimación. Una falta de claridad y precisión en las historias de usuario, combinada con una comunicación inadecuada dentro del equipo, puede llevar a expectativas desalineadas y retrasos en el proyecto. Por lo tanto, fomentar una cultura de comunicación abierta, colaboración continua y definición precisa de las historias de usuario es fundamental.

Más allá de las consideraciones técnicas, los factores humanos juegan un papel fundamental en los resultados del proyecto. La investigación de (Chakravorty et al., 2024) enfatiza la importancia de los factores críticos de éxito (FCS) en el desarrollo de software ágil, particularmente en entornos de trabajo remotos e híbridos. Estos FCS incluyen una comunicación efectiva, un liderazgo sólido y una visión compartida. Construir un equipo cohesivo y motivado, donde los individuos se sientan empoderados y respaldados, es esencial para el éxito del proyecto. Por el contrario, factores como dinámicas de equipo deficientes, falta de liderazgo y comunicación inadecuada pueden llevar al fracaso del proyecto.

El elemento humano se extiende a la relación entre el equipo de desarrollo y el cliente. Como se explora en (Barros et al., 2024), la participación del cliente y la seguridad psicológica son determinantes clave del éxito del proyecto. La participación activa del cliente en el proceso de desarrollo garantiza que el producto final se alinee con sus necesidades y expectativas. Además, un entorno psicológicamente seguro, donde los miembros del equipo se sientan cómodos compartiendo ideas y preocupaciones, fomenta la innovación y la resolución de problemas. Por el contrario, la falta de participación del cliente y un entorno de trabajo tóxico pueden obstaculizar el progreso y llevar al fracaso del proyecto.

El éxito de un proyecto de software depende de un delicado equilibrio entre factores técnicos y humanos. Al abordar cuestiones como la estimación precisa del esfuerzo, la colaboración efectiva del equipo, un liderazgo sólido, la participación del cliente y la seguridad psicológica, las organizaciones pueden mejorar significativamente sus posibilidades de entregar proyectos de software exitosos.

# ¿Qué tragedias ha causado el desarrollo de software con defectos?

El desarrollo de software ha revolucionado innumerables aspectos de nuestra vida, pero también ha dejado al descubierto las graves consecuencias que pueden surgir de errores en el código. En los últimos años, hemos sido testigos de cómo fallas en sistemas de software han desencadenado eventos trágicos con un impacto significativo en la sociedad.

Uno de los casos más recientes y alarmantes involucra a sistemas de transporte autónomo. En (Copp et al., 2023), un vehículo autónomo experimentó un fallo en su software de percepción, lo que resultó en una colisión fatal. Este incidente pone de manifiesto la importancia de garantizar la seguridad y fiabilidad de los sistemas de inteligencia artificial que controlan vehículos, especialmente en entornos urbanos densamente poblados. La creciente dependencia de estos sistemas exige un marco regulatorio sólido y rigurosas pruebas antes de su implementación a gran escala.

Más allá del sector del transporte, los defectos en el software también han tenido un impacto devastador en el ámbito de la salud. En el campo de la telemedicina, (Ramírez & López, 2021), un error en un sistema de telemedicina conduce a un diagnóstico erróneo de enfermedades grave, con consecuencias fatales para el paciente. Este caso subraya la necesidad de contar con sistemas de software de alta calidad en el sector de la salud, donde cualquier error puede tener consecuencias mortales. La pandemia de COVID-19 ha acelerado la adopción de soluciones de telemedicina, lo que hace aún más urgente abordar los desafíos de seguridad y fiabilidad en estos sistemas.

Los defectos en el desarrollo de software pueden tener consecuencias devastadoras, poniendo en riesgo vidas humanas y causando pérdidas económicas significativas. Es fundamental que la industria tecnológica adopte medidas más rigurosas para garantizar la calidad y seguridad del software, especialmente en sectores críticos como el transporte y la salud. Esto implica invertir en pruebas exhaustivas, desarrollar estándares de calidad más altos y promover una cultura de seguridad en el desarrollo de software.

# ¿Qué es la Ingeniería de Requisitos (IR)?

La Ingeniería de Requisitos (IR) es una disciplina fundamental en el desarrollo de software que se encarga de definir, documentar y gestionar todas las características y funcionalidades que un sistema debe poseer para satisfacer las necesidades de sus usuarios y clientes. En otras palabras, la IR es el puente entre las ideas iniciales y la implementación técnica de un proyecto.

El proceso de IR implica una serie de actividades que van desde la elicitación (recopilación) de requisitos hasta su validación y verificación. Durante la elicitación, se interactúa con los stakeholders (usuarios, clientes, patrocinadores) para comprender a fondo sus necesidades y expectativas. Esta información se documenta en un documento de requisitos, el cual sirve como referencia a lo largo de todo el ciclo de vida del proyecto. Como señala Sommerville (Sommerville I., 2016), un documento de requisitos bien elaborado es esencial para garantizar que el sistema desarrollado cumpla con las expectativas de los usuarios.

Una vez que los requisitos han sido documentados, se procede a su análisis y validación. En esta etapa, se verifica que los requisitos sean claros, completos, consistentes y no ambiguos. Además, se priorizan los requisitos en función de su importancia y se identifican posibles conflictos. La validación implica asegurar que los requisitos reflejen las verdaderas necesidades de los usuarios y que sean técnicamente factibles. Como lo indica Pressman (Pressman, 2010), la validación de requisitos es un proceso iterativo que puede requerir múltiples revisiones y ajustes.

La Ingeniería de Requisitos es una disciplina esencial que garantiza que los proyectos de software se desarrollen de manera eficiente y satisfagan las necesidades de los usuarios. Al definir claramente los requisitos desde el inicio del proyecto, se reducen los riesgos de malentendidos, retrasos y sobrecostos. Además, una buena gestión de los requisitos facilita la comunicación entre los diferentes equipos involucrados en el proyecto y contribuye a la calidad del producto final.

# ¿Cuál es la importancia de la Ingeniería de Requisitos?

Su importancia radica en que sienta las bases sólidas sobre las cuales se construirá el proyecto y garantiza que el producto final cumpla con las expectativas del cliente.

Una de las principales razones por las cuales la IR es crucial es que permite una comunicación efectiva entre todos los involucrados en el proyecto, desde los clientes y usuarios hasta los desarrolladores. Al establecer un conjunto claro y conciso de requisitos, se minimizan los malentendidos y se garantiza que todos trabajen hacia un objetivo común. Como señala Sommerville (Sommerville I., 2016), un documento de requisitos bien elaborado es esencial para una colaboración exitosa entre los diferentes equipos involucrados en el desarrollo de software.

Además de facilitar la comunicación, la IR contribuye a reducir los costos y los tiempos de desarrollo. Al identificar y documentar los requisitos al inicio del proyecto, se pueden detectar y resolver posibles problemas de manera temprana, evitando costosas modificaciones en etapas posteriores. Asimismo, una buena gestión de los requisitos permite priorizar las funcionalidades y concentrar los esfuerzos del equipo de desarrollo en las características más importantes para el cliente. Según Pressman (Pressman, 2010), una inversión inicial en la ingeniería de requisitos puede generar importantes ahorros a largo plazo.

La Ingeniería de Requisitos es una disciplina esencial que agrega un valor incalculable a cualquier proyecto de software. Al garantizar que todos los involucrados tengan una comprensión clara y compartida de los objetivos del proyecto, la IR contribuye a mejorar la calidad del software, reducir los riesgos y aumentar la satisfacción del cliente.

# ¿A quiénes dirigirnos para recolectar requisitos?

La recopilación de requisitos es una etapa fundamental en cualquier proyecto de desarrollo de software. Identificar a las personas adecuadas para obtener esta información es crucial para garantizar que el producto final satisfaga las necesidades de los usuarios y del negocio. Pero, ¿a quién debemos dirigirnos para recolectar estos requisitos?

Los stakeholders son todas aquellas personas o grupos que tienen un interés en el proyecto y que pueden verse afectados por los resultados del mismo. Estos pueden incluir clientes, usuarios finales, gerentes de proyecto, desarrolladores, equipos de soporte técnico, entre otros. Cada stakeholder tiene una perspectiva única y valiosa sobre el proyecto, por lo que es importante involucrarlos activamente en el proceso de recopilación de requisitos. Como señala Sommerville (Sommerville I., 2016), identificar y gestionar a los stakeholders es una tarea esencial para el éxito de cualquier proyecto de software.

La elección de los stakeholders a los que se dirigirá dependerá del tipo de proyecto y de su complejidad. En proyectos pequeños, puede ser suficiente involucrar al cliente y a un pequeño grupo de usuarios clave. Sin embargo, en proyectos grandes y complejos, es necesario identificar y consultar a un mayor número de stakeholders. Es importante tener en cuenta que los stakeholders pueden tener intereses y prioridades diferentes, por lo que es fundamental establecer un mecanismo para gestionar y resolver los conflictos que puedan surgir. Según Pressman (Pressman, 2010), la gestión de los stakeholders es un proceso continuo que debe realizarse a lo largo de todo el ciclo de vida del proyecto.

La recopilación de requisitos es un proceso colaborativo que implica la participación activa de los stakeholders. Identificar a las personas adecuadas y gestionar sus expectativas son tareas cruciales para garantizar que el proyecto sea un éxito. Al involucrar a los stakeholders desde el inicio del proyecto, se aumenta la probabilidad de que el producto final cumpla con sus necesidades y expectativas.

# ¿Qué son los requisitos?

Los requisitos, en el contexto de la ingeniería de software, son las características, funciones y restricciones que un sistema debe cumplir para satisfacer las necesidades de sus usuarios y clientes. En otras palabras, los requisitos definen qué debe hacer el software, cómo debe verse y bajo qué condiciones debe operar. Son la base sobre la cual se construye todo el proyecto de desarrollo.

Los requisitos pueden ser funcionales o no funcionales. Los requisitos funcionales describen las acciones que el sistema debe realizar, como por ejemplo "el sistema debe permitir al usuario registrarse", mientras que los requisitos no funcionales se refieren a las características de calidad del sistema, como la seguridad, la performance, la usabilidad, etc. Es fundamental identificar y documentar ambos tipos de requisitos para garantizar que el producto final cumpla con las expectativas del cliente. Como señala Sommerville (Sommerville I., 2016), una especificación de requisitos completa y precisa es esencial para el éxito de cualquier proyecto de software.

La correcta identificación y gestión de los requisitos es crucial para el éxito de un proyecto de software. Un conjunto de requisitos bien definido sirve como un contrato entre el cliente y el equipo de desarrollo, asegurando que el producto final cumpla con las necesidades del negocio. Además, una especificación de requisitos clara y concisa facilita la comunicación entre todos los involucrados en el proyecto y reduce el riesgo de malentendidos y errores. Según Pressman (Pressman, 2010), la ingeniería de requisitos es una disciplina fundamental que debe ser abordada de manera sistemática y rigurosa.

# ¿Cuál es la clasificación de los requisitos?

Los requisitos en ingeniería de software se pueden clasificar de diversas maneras para facilitar su comprensión, análisis y gestión. Una de las clasificaciones más comunes distingue entre requisitos funcionales y no funcionales.

Requisitos funcionales describen las acciones específicas que el sistema debe realizar para satisfacer las necesidades del usuario. Estos requisitos responden a la pregunta "¿Qué debe hacer el sistema?". Por ejemplo, "El sistema debe permitir al usuario registrarse", "El sistema debe calcular el total de una compra" o "El sistema debe generar un informe de ventas". Los requisitos funcionales están directamente relacionados con las tareas que el usuario espera que el sistema realice.

Por otro lado, los requisitos no funcionales describen las características de calidad del sistema, como la facilidad de uso, la seguridad, la performance, la confiabilidad, la mantenibilidad, entre otros. Estos requisitos responden a preguntas como "¿Qué tan rápido debe ser el sistema?", "¿Qué tan seguro debe ser el sistema?" o "¿Qué tan fácil de usar debe ser el sistema?". A diferencia de los requisitos funcionales, los requisitos no funcionales no describen acciones específicas, sino más bien las cualidades generales del sistema. Como señala Sommerville (Sommerville I., 2016), los requisitos no funcionales son fundamentales para garantizar la calidad y el éxito a largo plazo de un sistema de software.

# Describa las etapas del proceso de la IR

La Ingeniería de Requisitos (IR) es un proceso iterativo y dinámico que involucra varias etapas para garantizar que el software desarrollado cumpla con las expectativas de los usuarios y del negocio. A continuación, se describen las principales etapas de este proceso:

**Elicitación de Requisitos**: Esta etapa inicial consiste en recopilar información sobre las necesidades y expectativas de los usuarios, clientes y demás stakeholders involucrados en el proyecto. Se utilizan diversas técnicas para obtener esta información, como entrevistas, encuestas, observación de usuarios y análisis de documentos. El objetivo es comprender a fondo el problema que el software debe resolver y las características que debe poseer. Como señala Sommerville (Sommerville I., 2016), la elicitación de requisitos es un proceso complejo y desafiante que requiere habilidades de comunicación y negociación.

**Análisis y Negociación de Requisitos**: Una vez recopilada la información, se procede a analizarla y a negociar los requisitos con los stakeholders. En esta etapa se identifican los conflictos, se priorizan los requisitos y se resuelven las ambigüedades. Es importante involucrar a todos los stakeholders en este proceso para garantizar que todos estén de acuerdo con los requisitos finales. Además, se debe realizar un análisis de viabilidad para determinar si los requisitos son realistas y alcanzables desde un punto de vista técnico y económico.

**Especificación de Requisitos**: La especificación de requisitos consiste en documentar de forma clara, concisa y precisa todos los requisitos del sistema. El documento de requisitos debe ser comprensible tanto para los técnicos como para los usuarios no técnicos. Existen diversas técnicas y herramientas para documentar los requisitos, como diagramas de casos de uso, tablas de requisitos y prototipos. Como indica Pressman (Pressman, 2010), un documento de requisitos bien elaborado es esencial para la comunicación efectiva entre todos los involucrados en el proyecto.

**Validación de Requisitos**: La última etapa del proceso de IR consiste en verificar que los requisitos especificados sean correctos, completos y consistentes. Se realizan revisiones formales del documento de requisitos para identificar errores, omisiones y contradicciones. Además, se pueden realizar pruebas de concepto para validar la factibilidad técnica de los requisitos. La validación de requisitos es un proceso iterativo que puede requerir múltiples revisiones.

# Bibliografía

Barros, L., Tam, C., & Varajão, J. (2024). Agile software development projects–Unveiling the human-related critical success factors. *Information and Software Technology*, *170*. https://doi.org/10.1016/j.infsof.2024.107432

Chakravorty, G., Ramachandra Reddy, B., & Khan, D. A. (2024). Analyzing the quantitative impact of hybrid work model on critical success factors in scrum-based agile software. *International Journal of System Assurance Engineering and Management*, *15*(11), 5343–5355. https://doi.org/10.1007/s13198-024-02539-w

Copp, C. J., Cabell, J. J., & Kemmelmeier, M. (2023). Plenty of blame to go around: Attributions of responsibility in a fatal autonomous vehicle accident. *Current Psychology*, *42*(8), 6752–6767. https://doi.org/10.1007/S12144-021-01956-5/FIGURES/3

Iqbal, M., Ijaz, M., Mazhar, T., Shahzad, T., Abbas, Q., Ghadi, Y., Ahmad, W., & Hamam, H. (2024). Exploring issues of story-based effort estimation in Agile Software Development (ASD). *Science of Computer Programming*, *236*. https://doi.org/10.1016/j.scico.2024.103114

Pressman, R. S. (2010). *Ingenieria del Software. Un Enfoque Practico*. www.FreeLibros.me

Ramírez, A. M. S., & López, A. E. H. (2021). LOS SISTEMAS DE INFORMACIÓN EN SALUD COMO COMPONENTE PRIMARIO EN LA TELEMEDICINA. *UNESUM - Ciencias. Revista Científica Multidisciplinaria*, *5*(6), 1–12. https://doi.org/10.47230/UNESUM-CIENCIAS.V5.N6.2021.600

Sommerville I. (2016). *Ingenieria del Software 7ma. Ed. - Ian Sommerville*.