

Un modelo de gestión de requerimientos para minimizar el porcentaje de incumplimiento

A requirements management model to minimize the percentage of non-compliance

Victor Alfaro

RESUMEN

Desde la aparición de las metodologías de desarrollo de software, los procesos de desarrollo tienen como base la metodología padre: “El desarrollo en cascada”(Ian Sommerville, 1997). Actualmente se presentan diversas metodologías de desarrollo de Software, las metodologías ágiles, ahora bien ¿Por qué la vigencia actual de una forma de trabajo que aparece en el año 1990?, los resultados a corto plazo y el alcance de los objetivos propuestos(Pérez-Virgen, Salamando-Mejía, & Valencia-Ayala, 2013a). Según la tesis el objetivo de Scrum(Qumer & Henderson-Sellers, 2008) es logrado mediante la optimización del proceso de desarrollo mediante la identificación de las tareas, la gestión del tiempo más eficaz. Se establece como la Metodología en Cascada como la base, como el padre de todas las metodologías. El problema que se identifica en la presente investigación es la replanificación de los requerimientos en la fase de Planificación de desarrollo de software, como consecuencia se produce un alto índice de replanificaciones. La crítica que se realiza es la falta de artefactos que garanticen la viabilidad del proyecto desde la fase de Planificación. Como solución en la presente investigación se presentan filtros, entregables en cada etapa de Planificación, de tal manera que se garanticen la calidad de reuniones, entendimiento, checklists. Por último, la demostración se realizará al término de todo el proyecto, en la fase de cierre, evaluando y analizando los controles de cambios, el tiempo y la puesta en marcha de la postproducción del proyecto.

Palabras clave: Scrum, software, planificación, desarrollo

ABSTRACT

Since the emergence of software development methodologies, development processes are based on the parent methodology: “Cascade development”(Ian Sommerville, 1997). Currently there are several software development methodologies, agile methodologies, now why the current validity of a form of work that appears in the year 1990? The short-term results and the scope of the proposed objectives(Pérez-Virgen et al., 2013a). According to the thesis the goal of Scrum is achieved by optimizing the development process by identifying the tasks, the most effective time management. It is established as Cascade Methodology as the basis, as the father of all methodologies. The problem that is identified in this research is the replanning of the requirements in the software development planning phase, as a consequence there is a high rate of replanning. The criticism that is made is the lack of artifacts that guarantee the viability of the project from the Planning phase. As a solution in the present investigation, filters are presented, deliverables in each stage of Planning, in such a way as to guarantee the quality of meetings, understanding, checklists. Finally, the demonstration will be carried out at the end of the whole project, in the closing phase, evaluating and analyzing the change controls, the time and the start-up of the post-production of the project.

Keywords: Scrum, software, planning, development

1. Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos. Lima, Perú
Ingeniero de Sistemas e Informática. Egresado Maestría Ingeniería de Sistemas e Informática. Consultor independiente. Email: victor.alfaro@unmsm.edu.pe



INTRODUCCIÓN

En el mundo, desde la aparición de las herramientas para realizar desarrollos de software, se presentan inconvenientes que provocan retraso sistemático en los plazos establecidos, ocasionando de esa forma que se exceda en el presupuesto, se entregue con una alta tasa de errores y su utilidad sea inferior a la esperada (Arruzazabala, Dapozo, & Thomas, 2012). En cuanto magnitud esta problemática es atribuible a defectos en los procesos utilizados para recoger, documentar, acordar y modificar los requisitos del sistema (Aларcon & Sandoval, 2011).

Una gran variedad de investigaciones coincide en que la mayor parte de los errores de un sistema se originan al comienzo del ciclo de vida, y que entre el 40% y 60% de los defectos en un proyecto de software están relacionados con los requerimientos del sistema (Sánchez & Velthuis, 2012).

Según Franck Agger (Aggeri & Labatut, 2010) en su investigación demuestra la importancia de la gestión de requerimientos dentro del proceso de desarrollo de software y se verifica la actividad de la revisión de los requerimientos del software como un factor clave dentro del ciclo de vida.

Una de las más preponderantes investigaciones realizadas a múltiples organizaciones en el mundo, que mide los distintos factores que interfieren en el desarrollo de soluciones, que es elaborada por "The Standish Group" (The Standish Group, 1995), identificó que la gestión de requerimientos sigue siendo la etapa de mayor problema para la mayoría de las organizaciones y que las prácticas asociadas a los requisitos son deficientes en el 74% de todas las empresas (Borrell, 2006).

Dicha investigación ha identificado que la causa principal de la baja calidad, alto coste y problemas de entrega del producto sigue siendo por factores asociados a la gestión de requerimientos y entre esos factores se encuentra: La deficiencia en entender las necesidades del cliente, una in-

completa especificación de los requerimientos y la falta de una correcta gestión de cambios de los mismos.

Ahora bien, ya citada la importancia de la gestión de requerimientos en el desarrollo de una solución de software, se verifica un modelo para la mejora del proceso de gestión de Requerimientos, que ha sido probada en las organizaciones (6), es utilizar modelos de buenas prácticas, tales como, CMMI-DEV, ISO-15504 (2003; 2004; 2008; 2012), ISO-12207 (2008). Se presentan demostraciones de casos de éxito en donde la implementación y el uso de estos modelos ha permitido disminuir los problemas de entrega y calidad de un producto software (Peruana, 2004).

Prosiguiendo con la situación en Latinoamérica, en una investigación realizada por Borrell Caridad Racero, en Medellín, Colombia, se presenta la necesidad y la problemática idéntica a Europa. La omisión de pasos importantes en el ciclo de vida de desarrollo del software, entre tantas, la correcta definición de requerimientos. Lo que se pretende con la investigación es cubrir los requisitos de la fase de especificación y sus principales actividades durante el proceso de desarrollo de software (Borrell, 2006).

En el Perú, según la Norma Técnica Peruana (Peruana, 2004), existe una proliferación de marcos de trabajo referenciales para el desarrollo de software. Para la gestión de requerimientos menciona principalmente el proceso de Adquisición donde conviene que los requisitos del sistema incluyan requisitos de negocio, organizativos, de usuario, así como de seguridad física y de acceso y otros requisitos críticos.

La Norma Técnica peruana proporciona la etapa de Adquisición y sus fases para la gestión de requerimientos: Inicio, Preparación de la solicitud de propuestas, Preparación y actualización del contrato, Seguimiento del proveedor, aceptación y finalización (9). De acuerdo con la Oficina Nacional de Gobierno Electrónico e Informática brinda la puesta en marcha de la Norma Técni-

ca Peruana “NTP- ISO 12207:2008 Tecnología de Información, Proceso del ciclo de vida del desarrollo de software, 1era edición” para que se aplique(Irrazabal, Vásquez, Díaz, & Garzás, 2011).

MÉTODOS Y MATERIALES

A. Metodología en Cascada

En Ingeniería de software el desarrollo en cascada, también llamado modelo en cascada (denominado así por la posición de las fases en el desarrollo de esta, que parecen caer en cascada “por gravedad” hacia las siguientes fases), es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior(Ian Sommerville, 1997).

Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase. Este modelo fue el primero en originarse y es la base de todos los demás modelos de ciclo de vida.

La versión original fue propuesta por Winston W. Royce en 1970 y posteriormente revisada por Barry Boehm en 1980 e Ian Sommerville en 1985 (Letelier, Canós, Sánchez, & Penadés, 2003).

Un ejemplo de una metodología de desarrollo en cascada es:

- Análisis de requisitos.
- Diseño del Sistema.
- Diseño del Programa.
- Codificación.
- Pruebas.
- Verificación.
- Mantenimiento.

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamen-

te al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo. La palabra cascada sugiere, mediante la metáfora de la fuerza de la gravedad, el esfuerzo necesario para introducir un cambio en las fases más avanzadas de un proyecto.

La presente investigación enfoca las dos primeras fases; Análisis de requisitos y diseño del sistema. Las mismas que engloban la etapa de Planificación. En dicha etapa se presente el volcamiento de la información, reuniones de trabajo, uso de cronograma de actividades, checklists.

B. Metodologías de desarrollo Ágiles, Scrum

En la actualidad y en los últimos años varias iniciativas internacionales orientadas específicamente para armar pequeñas y medianas empresas, los procesos y los métodos ágiles han sido identificados.

Del mismo modo, diferentes estudios han identificado la correlación entre las metodologías ágiles y modelos de procesos de desarrollo de software como CMMI-DEV e ISO / IEC 12207, pero los estudios relacionados con la norma ISO / IEC 12207(ISO, IEC, & IEEE, 2008) se basan en la versión de 1995(Irrazabal et al., 2011). Por lo tanto, este trabajo se centra en la relación entre las prácticas ágiles, especialmente scrum(Cockburn & Highsmith, 2001), y un subconjunto de proceso de la versión 2008 de la norma ISO / IEC 12207(Irrazabal et al., 2011).

SCRUM(Pernstål, Gorschek, Feldt, & Florén, 2015) es uno de los métodos más populares y ágiles es un proceso iterativo incrementales. Estas dos características hacen dividir el proyecto en fases o iteraciones y la entrega incremental del proyecto.

Las relaciones indicadas en el presente trabajo se obtienen a través de trabajos previos y experiencia en consultoría a 25 empresas que cumplen con los resultados estándar de aplicación de me-

metodologías ágiles(Irrazabal et al., 2011).

El objetivo principal del estudio es conocer el grado en que las prácticas ágiles ayudan en la realización de las prácticas indicadas en este modelo de proceso(Irrazabal et al., 2011).

La metodología scrum es uno de los métodos más usados en el mundo, particularmente en el Perú, su llegada ha sido tardía, existen diversos esfuerzos para establecer buenas prácticas en las organizaciones, pero el tema cultural es una gran barrera que distorsiona el objetivo principal del marco de trabajo, que es el de brindar valor en etapas tempranas.

Existen diversos estudios del marco de trabajo, uno de ellos del creador del marco Jeff Sutherland(Verbruggen, 2018) que indica diversas prácticas en cuanto a la velocidad del equipo, en cuanto al manejo adecuado del marco de trabajo.

Asimismo existen esfuerzos separados por países para establecer los pilares del marco en las organizaciones: grupos de trabajo, comunidades.

C. Norma Técnica Peruana ISO 12207: 2008

En el Perú, según la Norma Técnica Peruana(Peruana, 2004), existe una proliferación de marcos de trabajo referenciales para el desarrollo de software.

Para la gestión de requerimientos menciona principalmente el proceso de Adquisición donde conviene que los requisitos del sistema incluyan requisitos de negocio, organizativos, de usuario, así como de seguridad física y de acceso y otros requisitos críticos.

La Norma Técnica peruana proporciona la etapa de Adquisición y sus fases para la gestión de requerimientos: Inicio, Preparación de la solicitud de propuestas, Preparación y actualización del contrato, Seguimiento del proveedor, aceptación y finalización(Arruzabala et al., 2012).

De acuerdo con la Oficina Nacional de Gobierno Electrónico e Informática brinda la puesta en marcha de la Norma Técnica Peruana “NTP- ISO 12207:2008 Tecnología de Información, Proceso del ciclo de vida del desarrollo de software, 1era edición” para que se aplique(Chaves, 2011).

RESULTADOS

El artefacto a presentar va permitir analizar y diagnosticar el rendimiento productivo de la metodología propuesta para el manejo de requerimientos de gran complejidad.

La metodología scrum(Irrazabal et al., 2011) cuenta con una estructura secuencial y repetitiva que se caracteriza por presentar ciclos reducidos, cortos en alcance, con el fin de concretar entregables visibles para el usuario final, de tal manera que la percepción del trabajo sea constante y la validación de los entregables se realicen al final de cada ciclo, con la evaluación de las actividades realizadas, con aprobaciones de los entregables y las metas establecidas.

Cada ciclo de trabajo se llama Sprint, donde cada sprint se puede comparar como un pequeño entregable, el cual posee alcance, estimación, tiempos.

Según se observa en la Figura N° 1, cada sprint es un ciclo de trabajo reducido, que presenta entradas (inputs), salidas (outputs) y fases internas de análisis y desarrollo. En dicha figura se representan como 5 fases por cada sprint.

Según la Figura N° 1, expuesta en la introducción, existe una correlación del marco metodológico de Scrum(Irrazabal et al., 2011) con la metodología de desarrollo Cascada.

El artefacto a desarrollar va modificar la primera parte que es Planificación, detallando un entregable checklist que se encargará de realizar el diagnóstico y viabilidad de la planificación.



Figura N° 1. Diagrama de la Metodología scrum desplegado bajo la metodología en cascada
Fuente: Elaboración propia.

Si realizamos la función de una lupa imaginaria podemos expandir cada uno de los sprints. De acuerdo a la Figura N° 2, Se observa que a modo de ejemplo se toman 3 entradas: Entrada A, Entrada B y Entrada C, y luego se presentan 3 Salidas: Salida A, Salida B y Salida C. En el ínterin se presentan 5 fases de actividad que son: Requerimientos, diseño, codificación, pruebas de aceptación. Cada una comprende de manera singular a las etapas propias de la metodología de desarrollo de software en Cascada: Desarrollo de Requisitos, diseño de solución, desarrollo y pruebas.

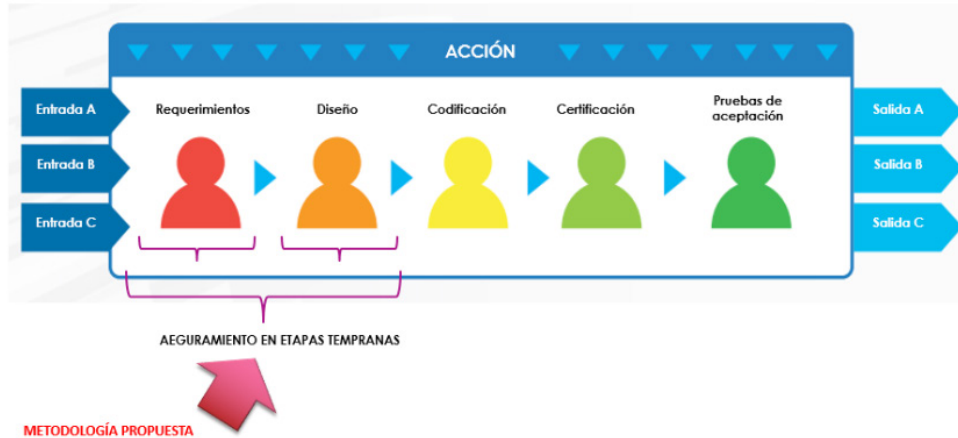


Figura N° 2: Metodología Scrum Híbrida – Cascada

Entonces, las dos primeras etapas que son Requerimientos y Diseño son las dos únicas fases que para esta investigación comprenden en cuanto al alcance. En vista que en ellas se presenta el establecimiento de los requerimientos y el diseño de la solución.

Ahora bien, realizamos nuevamente la función imaginaria de la lupa y expandimos nuevamente las dos primeras etapas antes expuestas: Requerimientos y Diseño. Según la Figura N° 3, el modelo propuesto presenta, y expone artefactos de medición de la viabilidad de cada una de las etapas.

Para la etapa de Requerimientos se cuenta con:

- Programación de reuniones
- Cuestionario desarrollo de requisitos
- Checklist Aseg-Doc – Alcance
- Reporte de hallazgos Aseg-Doc – Alcance
- Seguimiento y cierre de hallazgos Aseg-Doc- Alcance
- Seguimiento de pendientes

Para la etapa de Diseño se cuenta con:

- Seguimiento de pendientes
- Checklist Aseg. Doc Análisis
- Seguimiento y cierre de hallazgos Aseg. Doc – Análisis

El checklist que se genera en la etapa de desarrollo de requerimientos va a ser el medidor de viabilidad para pasar a la siguiente fase que el Diseño de Solución.



Figura N° 3: Diagrama de la etapa de Planificación: requerimientos y diseño con sus respectivos artefactos. Fuente: Elaboración propia.

DISCUSIÓN

La presente solución comprende la evaluación de los documentos de alcance en la etapa de desarrollo de requisitos y diseño de soluciones de acuerdo a la metodología a través de los entregables a presentar, de tal manera que ejecutarán un filtro para poder evaluar la viabilidad de dichos documentos.

Se presenta a través de un flujo de actividades según la Figura N° 4 que se encarga de realizar un seguimiento.

Se inicia con la fase de Planificación, donde se genera el cronograma de actividades y un archivo de reuniones, documentos que serán utilizados para poder verificar el cumplimiento de las fechas establecidas para los compromisos.

Luego se inicia con la fase de relevamiento, donde se generan las reuniones de entendimiento, utilizando un cuestionario de preguntas y un seguimiento de pendientes para poder registrar los pendientes desde el inicio de dicha fase. Prosiguiendo con las actividades en el control de avances, paso previo a la aprobación se genera el documento de Alcance de desarrollo de requisitos.

Es en este documento el cual se aplica el filtro a través del checklist de alcance (Sánchez & Velthuis, 2012).

Este documento comprende un conjunto de preguntas alineadas con el documento de alcance, el cual tiene como finalidad determinar, en cuanto a una escala de niveles, la garantía en cuanto a calidad de información. Luego se genera un promedio ponderado, utilizando ratios para cada pregunta.

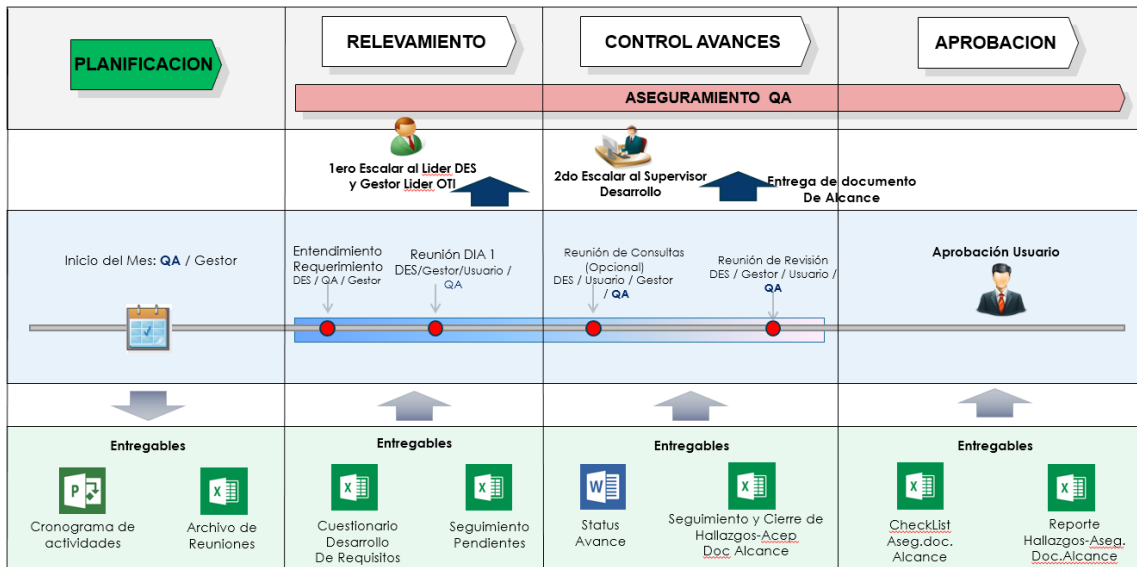


Figura N° 4. Diagrama de la etapa de Planificación: Requerimientos y diseño con sus respectivos artefactos

Fases del modelo propuesto

Desarrollo de requisitos:

Según la Figura N° 4, La fase de desarrollo de requisitos o Requerimientos(Pérez-Virgen, Salamando-Mejía, & Valencia-Ayala, 2013b) comprende 4 actividades: Planificación, Relevamiento(Ian Sommerville, 1997) Control de avances y Aprobación(Ali & Lai, 2016).

La primera actividad Planificación inicia con la elaboración del cronograma de actividades y la programación de reuniones del mes. Estos dos documentos servirán como control de actividades a lo largo del tiempo establecido.

El cronograma de actividades tendrá el registro de todas las reuniones que posteriormente se realizarán, los entregables, algunas actividades a la interna, coordinaciones, etc. La segunda acti-

vidad Relevamiento, se genera el cuestionario de desarrollo de requisitos y el seguimiento de pendientes.

Con estos dos documentos se busca garantizar la continuidad e integridad de la información que se genera en las reuniones. Como tercera actividad se encuentra el Control de Avances, donde se verifica un status de avance, el mismo documento de seguimiento de pendientes que se presenta en la etapa de relevamiento actualizado y sobre todo el Checklist Aseg doc Alcance.

Este documento será el medidor principal de viabilidad que se propone en el modelo propuesto. Para finalizar la actividad de Aprobación se realizará al finalizar el sprint y contará con la aprobación del usuario.

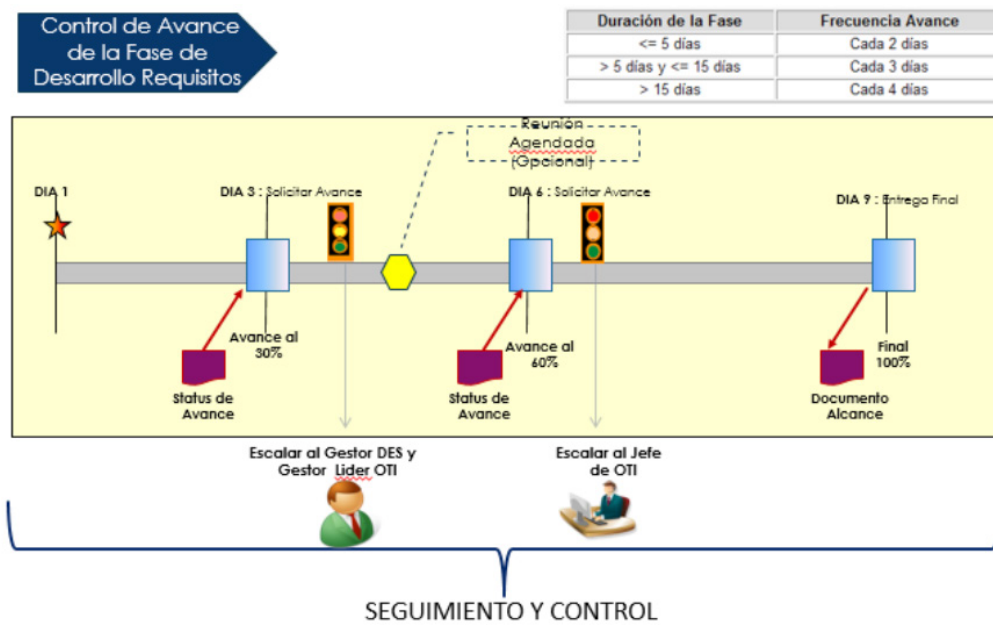


Figura N° 5. Modelo propuesto - Etapa Desarrollo de requisitos detalle

Diseño de Solución:

La fase de diseño de Solución o Diseño comprende, según la Figura N° 6, las actividades: Control de Avances y la Aprobación. Durante la actividad de control de avances se generan los documentos Status de avance, Seguimiento de pendientes, Reporte de hallazgos, Checklist (Pedraza-Jiménez, Banco, Codina, & Cavaller, 2013) de Aseguramiento de Diseño y Seguimiento de cierre de hallazgos.

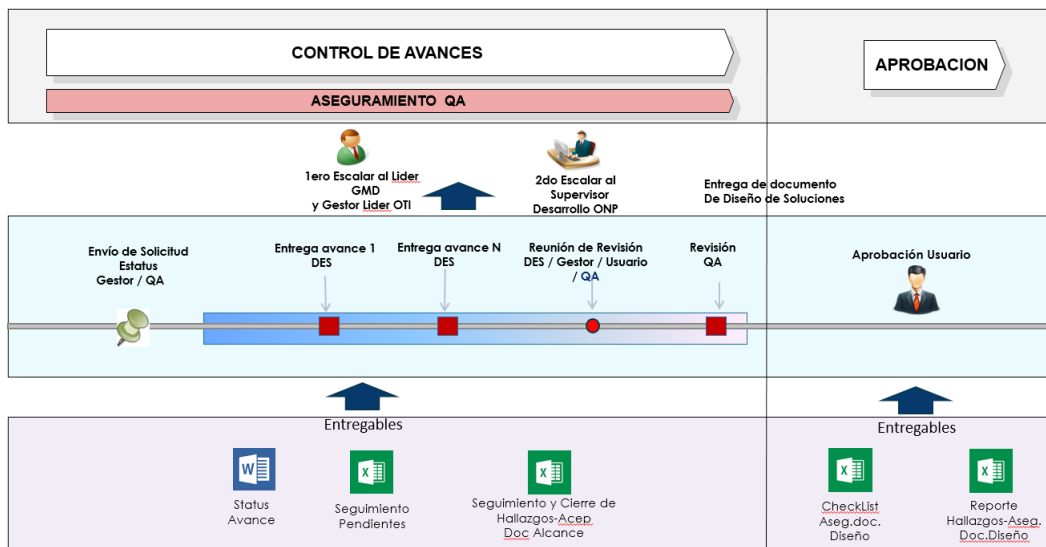


Figura N° 6. Diagrama completo de las etapas de diseño

Demostración del modelo propuesto:

Para la demostración de la presente investigación, se realizó una implementación en tiempo real en la empresa Efact y la empresa estatal ONP (Información, n.d.), aplicando el modelo propuesto, que consisten en las herramientas expuestas en el punto a y b de la presente sección. Se obtuvo como

resultados los siguientes datos de validación y de disminución de los porcentajes de incumplimiento como corresponde a la aplicación del problema. El objetivo del modelo propuesto se satisface con la disminución de los porcentajes de incumplimiento.

Etapa	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Setiembre	Octubre	Noviembre	Diciembre	Promedio
Desarrollo Requisitos	10%	10%	20%	12%	11%	9%	2%	0%	10%	12%	7%	8%	9%
Diseño Soluciones	11%	9%	8%	12%	12%	10%	2%	8%	7%	4%	5%	10%	7%

Figura N° 7. Cuadro estadístico de los resultados obtenidos en el periodo 2015- 2017 ONP

Se demuestra objetivamente, en base a una comparación de gráficos, la disminución del porcentaje de incumplimiento en las fases de desarrollo y diseño, por ende la solución al problema principal.

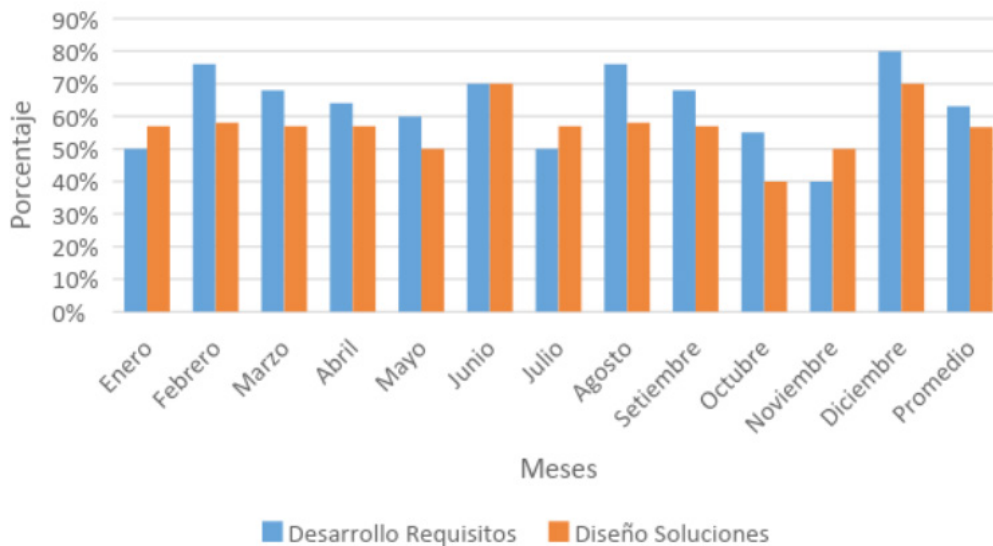


Figura N° 8. Tabla d los resultados del periodo 2015 - 2017 ONP

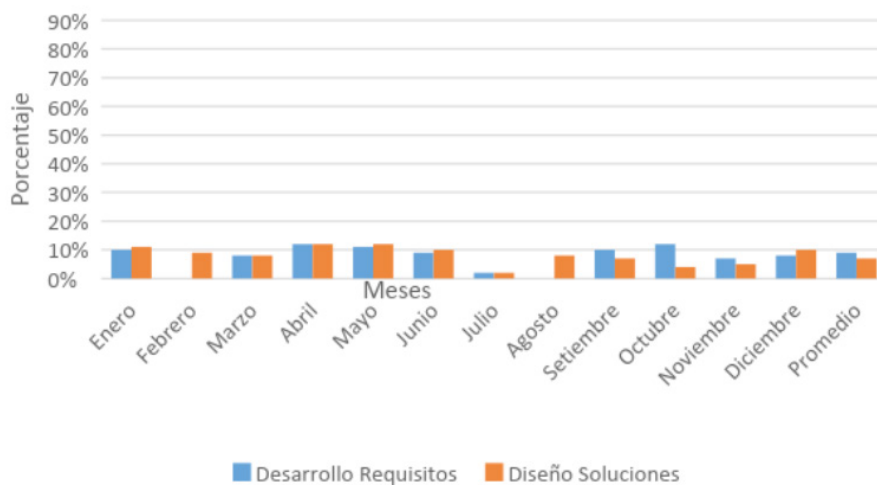


Figura N° 9: Muestra de resultados del periodo 2018 – ONP

CONCLUSIONES

Con la adopción de estas herramientas para poder garantizar la calidad de la información se espera resultados óptimos en postproducción. Los mismos que generan satisfacción al usuario final y por ende se aminora y se suprime el problema principal que es el retraso garantizado en la etapa de planificación, que es el porcentaje de incumplimiento en la fase de planificación en cuanto a los requerimientos dentro de una organización. Con el modelo propuesto se espera que el porcentaje disminuya, se está realizando la validación del modelo propuesto a través de medición de checklist y recorrido de datos.

REFERENCIAS BIBLIOGRÁFICAS

- Alarcon, A., & Sandoval, E. (2011). **Herramientas CASE para ingeniería de Requisitos**. *Cultura Científica*, 0(6). Retrieved from <http://www.revistasjdc.com/main/index.php/ccient/article/view/37>
- Arruzazabala, M. C., Dapozo, G., & Thomas, P. (2012). Certificación ISO 9001:2008: **Impacto en el Proceso de Ingeniería de Requerimientos**. In *CACIC 2012 Anales del XVIII Congreso Argentino de Ciencias de la Computación* (pp. 744–753).
- Borrell, C. R. (2006). **Importancia de la ingeniería de requerimientos dentro del ciclo de desarrollo de software**. (Spanish). *Importance of Requirements Engineering in the Software Development Life Cycle*. (English), (3), 52–56. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&db=fua&AN=34162106&lang=es&site=ehost-live>
- Chaves, M. A. (2011). **La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software**. *InterSedes*. Retrieved from <http://revistas.ucr.ac.cr/index.php/intersedes/article/view/790>
- Ian Sommerville. (1997). **Ingeniería De Software. Informática Industrial**, 687. Retrieved from [http://books.google.com/books?hl=en&lr=&id=L5tVdqFU3jcC&oi=fnd&pg=PA45&dq=ingenieria+de+software&ots=Dgx7drEqNa](http://books.google.com/books?hl=en&lr=&id=L5tVdqFU3jcC&oi=fnd&pg=PA45&dq=Ingenieria+de+Software&ots=Dgx7drEqNa&sig=Kkm4_vRo_scAHzqrSZo9Dd_pyOw%5Cn)
- Información, O. de T. de. (n.d.). **PLAN_82_Plan Operativo_Informático_2013_2013 ONP.pdf**. ISO, IEC, & IEEE. (2008). *Systems and software engineering — Software life cycle processes (ISO/IEC 12207:2008(E); IEEE Std 12207-2008)*. ISO/ IEC/ IEEE.
- Letelier, P., Canós, M., Sánchez, E., & Penadés, M. (2003). **Métodologías Ágiles en el Desarrollo de Software**. Valencia, Valencia, España, 1–8. Retrieved from http://www.carlosfau.com.ar/nqi/nqifiles/XP_Agil.pdf
- Pedraza-Jiménez, R., Banco, S., Codina, L., & Cavaller, V. (2013). **Diseño conceptual y especificación de requerimientos para el desarrollo y rediseño de sitios web**. *El Profesional de La Información*, 22, 74–79. <https://doi.org/10.3145/epi.2013.ene.10>
- Pérez-Virgen, H. L., Salamando-Mejía, C. A., & Valencia-Ayala, L. S. (2013a). *Levantamiento De Requerimientos Basados En El Conocimiento Del Proceso*. **Revista Científica**, (16), 42–51. Retrieved from <http://revistas.udistrital.edu.co/ojs/index.php/revcie/article/view/4022>
- Pérez-Virgen, H. L., Salamando-Mejía, C. A., & Valencia-Ayala, L. S. (2013b). *Levantamiento De Requerimientos Basados En El Conocimiento Del Proceso*. **Revista Científica**. Retrieved from <http://revistas.udistrital.edu.co/ojs/index.php/revcie/article/view/4022>
- Peruana, N. T. (2004). **Norma Técnica Peruana De Tecnología De La Información Ciclo De Vida Del Software**, (18). Retrieved from <https://www.pnp.gob.pe/normas legales/ONGEI/RM N° 179-2004-PCM Uso Obligatorio de Norma tecnica Peruana 12207-2004 Procesos de Ciclo de Vida del Software.pdf>
- Qumer, A., & Henderson-Sellers, B. (2008). **A framework to support the evaluation, adoption and improvement of agile methods in practice**. *Journal of Systems and Software*. <https://doi.org/10.1016/j.jss.2007.12.806>
- Sánchez, C. M. F., & Velthuis, M. G. P. (2012). **Modelo para el gobierno de las TIC basado en las normas ISO**.
- The Standish Group. (1995). *The standish group report. Chaos*, 49, 1–8. <https://doi.org/10.1145/1145287.1145301>
- Verbruggen, F. (2018). **Process Efficiency – Adapting Flow to the Agile Improvement Effort Process Efficiency – Adapting Flow to the Agile Improvement Effort**, (September).