



**UNIVERSIDAD
DE ANTIOQUIA**

**ASEGURAMIENTO DE CALIDAD EN EL DESARROLLO DE UN
SITIO WEB CON EL FRAMEWORK SAP COMMERCE**

Autora
Miriam Hasana Arango Vásquez

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería de
Sistemas
Medellín, Colombia
2020



Aseguramiento de Calidad en el desarrollo de una aplicación web con el Framework SAP Commerce

Miriam Hasana Arango Vásquez

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

Profesional de Ingeniería de Sistemas

Asesora externa:

July Andrea Gutiérrez Franco

Psicóloga Organizacional

Asesora interna:

Deisy Yuried Loaiza Berrío

Ingeniera de Sistemas

Línea de Investigación:

Calidad de Software

Universidad de Antioquia

Facultad de Ingeniería, Departamento de Ingeniería de Sistemas

Medellín, Colombia

Colombia.

ASEGURAMIENTO DE CALIDAD EN EL DESARROLLO DE UN SITIO WEB CON EL FRAMEWORK SAP COMMERCE

Miriam Hasana Arango Vásquez
Ingeniera de Sistemas en formación, Universidad de Antioquia

Resumen – El aseguramiento de calidad es un área necesaria que debe ser aplicada en todos los proyectos de sistemas de la información para lograr la satisfacción del cliente al momento de tener el producto en producción. En este informe se describe todos los elementos necesarios para asegurar la calidad durante la creación de un sitio web, con el framework SAP Commerce. Por cada Sprint se obtiene una cantidad de datos que fueron procesados por medio de diferentes métricas, las cuales muestran el desempeño del desarrollo del producto y el equipo de trabajo. Al final se enumeran diferentes situaciones que explican los causantes de los diferentes defectos por cada sprint. Cada caso de riesgo debe ser tenidos en cuenta y evaluado para contribuir en el mejoramiento de futuros proyectos de esta índole

Índice de Términos – aseguramiento de calidad, SAP Commerce, casos de prueba, pruebas funcionales, pruebas de interfaz de usuario, pruebas de regresión, pruebas end to end.

Summary – Quality assurance is very important for each IS project to accomplish the client's requirements and concluding a successful project. This paper describes all necessary elements to accomplish quality assurance when a web site is created with SAP Commerce Framework. For each sprint, we were able to collect some data that were processed for some metrics. This information helps us to know-how is a team performance, which are the risk cases and what we can do for not commit the same mistakes in future projects.

Keywords – Quality Assurance, SAP Commerce, test cases, functional testing, UI testing, regression testing, end to end testing

I. INTRODUCCIÓN

Las dos grandes temáticas que se abordan en este escrito son el Framework SAP Commerce, y las técnicas más relevantes del aseguramiento de calidad utilizadas durante el desarrollo de un sitio web con este framework. La importancia en el Aseguramiento de calidad para los productos creados por cualquier empresa radica en perdurar su producto o su servicio

durante mucho tiempo. Asegurar la calidad significa complacer la necesidad del cliente y tenerlo satisfecho [2]. Además, en la gestión de proyectos existen cinco variables importantes las cuales influyen en el éxito de su realización y culminación, las cuales son: alcance, tiempo, coste, calidad y riesgo [8]. Por medio del aseguramiento de calidad se busca ejecutar una de las variables para que el proyecto al final sea exitoso. En la actualidad la mayoría de los proyectos de tecnologías de la información se consideran que tienden a una culminación en un estado dudoso (no se tiene certeza si se culminaron de manera exitosa o no) como se explica en el informe del caos del 2015 donde se asegura que el 29% de los proyectos son exitosos, el 19% fallido y el 59% en discusión [7].

El aseguramiento de calidad para el desarrollo de sitios es un tema que a tomado mucha importancia desde hace poco tiempo, alrededor de unos 12 años. No todas las empresas tienen los recursos o presupuesto para invertir en un departamento de QA, aunque sea completamente necesario [2]. Al igual que lo expone Dante Carrizo y Andrés Alfaro en su artículo “*Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico*” [2] donde se propone una estrategia por medio de diferentes técnicas, para el aseguramiento de calidad de un sitio web, en este escrito se expone las técnicas utilizadas para asegurar la calidad en la creación de un sitio, sus métricas y resultados.

El objetivo del framework SAP es crear sitios de tipo ecommerce de manera más rápida, con el 80% del trabajo ya realizado y el 20% para ser creado por el equipo, desarrollado en base a los requerimientos y personalizaciones que el cliente desee. La importancia del desarrollo de sitios ecommerce radica en la necesidad que tienen las empresas de establecer sus productos y servicios de manera digital por el uso masivo del internet, existiendo una cantidad muy grande de clientes potenciales en la web.

En la tesis de Maestría en Ingeniería informática titulada “*Implantación de una plataforma eCommerce basada en SAP Hybris para una empresa multinacional del*

sector de servicios”, realizado por Carlos Quer Barbare, se presenta de manera detallada cómo una empresa multinacional del sector de servicios realizó el proceso de cambio de su plataforma electrónica por una plataforma SAP Hybris Commerce. En el desarrollo del escrito los temas más relevantes son las razones por las cuales se escoge este framework de trabajo, se describe el problema y se plantea la solución de manera introductoria. Por medio de un estado del arte se muestra la definición de diferentes frameworks entre ellos SAP Hybris y Magento. Una vez esto claro, se pasa a desarrollar la solución donde se especifican tres fases del desarrollo del proyecto, en la primera fase se encuentra el módulo de QA, el cuál es el eje central de esta propuesta. En esta tesis se concluye que el impacto de la plataforma eCommerce por medio del framework Sap Hybris fue muy positivo y satisfactorio para los objetivos trazados y que a pesar de los problemas existentes y que puedan existir sigue siendo una excelente solución [12].

Las técnicas de aseguramiento de calidad y los resultados que se exponen en este escrito se obtuvieron a partir del desarrollo de dos sitios B2C por medio del framework SAP Commerce para una multinacional francesa que se dedica a proveer servicios de contratación eléctrica.

El alcance de este proyecto se limita a la creación de una excelente experiencia de usuario en la búsqueda de productos, navegación en el sitio y en el proceso de compra o pago.

La metodología utilizada para la creación de estos sitios consiste en una metodología ágil, dividida en 6 sprints, cada uno contiene un bloque de tareas repetitivas realizadas por el equipo de QA, las cuales están descritas en la sección de metodología.

II. OBJETIVOS

A. *Objetivo general.*

Asegurar la calidad en la creación de un sitio B2C por medio del framework SAP Commerce.

B. *Objetivos específicos.*

- Crear casos de prueba funcionales, de interfaz de usuario y de regresión para cada aspecto del sistema.
- Ejecutar los casos de prueba funcionales y de interfaz de usuario de forma manual.

- Ejecutar pruebas de regresión y casos de prueba end to end.

III. MARCO TEÓRICO

El aseguramiento de calidad para tecnologías de la información consiste en prevenir los fallos durante el desarrollo del sitio, para evitar que un producto defectuoso entre en la fase de producción [5]. Este aseguramiento de calidad se hizo para el desarrollo de estos sitios.

Los sitios B2C son los sitios web creadas para las empresas que venden directamente a los consumidores finales del producto [13]. SAP Commerce es un Framework que contiene diferentes aceleradores para la creación de sitios web como B2C, B2B, empresas de telecomunicaciones, de viajes, etc. Estas plantillas permiten crear y mantener una solución de comercio omnicanal con muchas funciones [14].

El equipo de aseguramiento de calidad trabaja a partir de un plan maestro de prueba el cual consiste en una plantilla que contiene varias secciones entre ellas: la introducción, la sección de alcance de las pruebas, donde se describe el tipo de pruebas a realizar y en qué dispositivos y sistemas operativos, la sección de lo que no está en el alcance, la sección del enfoque donde se explica el proceso a realizar de manera recurrente durante todos los sprints, el manejo de defectos y el cronograma [15].

En este proyecto se trabajó por medio de una metodología ágil la cual consiste en la gestión de grupos de trabajo y proyectos, donde se tiene en cuenta la eficiencia en el tiempo y los recursos invertidos en las actividades a realizar [16].

Para poder gestionar el proyecto fue necesario la creación de historias de usuario. Una historia de usuario es una descripción de una funcionalidad que debe tener un sistema de información, la cual es adquirida por medio de los requerimientos [17]. Estas historias de usuario se organizaban en un gestor de tareas de proyectos de software, conocido como JIRA [18].

El equipo de aseguramiento de calidad tuvo que crear casos de pruebas funcionales y de UI (User interface) o estilo. Un caso de prueba es un conjunto de acciones que tienen como objetivo probar de diferentes maneras (si es posible) la funcionalidad de un escenario en específico. En otras palabras, los casos de pruebas especifican el cómo debe probarse un escenario [19]. Ahora un escenario de prueba es una funcionalidad

general a probar que debe ser especificada en uno o varios casos de prueba. En otras palabras, un escenario de prueba especifica el qué debe probarse [19].

Los diferentes tipos de prueba que se ejecutaron fueron funcionales, de interfaz de usuario, de regresión, end to end y se acompañó al equipo durante dos semanas en las pruebas de aceptación de usuario. Las pruebas funcionales son aquellas que se hacen a partir de los requerimientos entregados por el cliente [9]. Las pruebas de interfaz de usuario consisten en verificar el buen funcionamiento de los estilos y componentes en los diferentes dispositivos [10]. Las pruebas de regresión tienen como objetivo verificar las funcionalidades más importantes del sitio, para comprobar que los nuevos cambios no hayan dañado esas funcionalidades ya creadas [11]. Las pruebas end to end son aquellas que tratan de simular un proceso completo en el sitio, en este caso un proceso de compra, desde agregar productos al carrito hasta hacer el pago [11]. Por último, *pruebas de aceptación del usuario* o UAT (User Acceptance testing) Es el proceso de verificación que hace el usuario donde comprueba que la solución desarrollada es la que se requiere [20].

Para la gestión de los procesos realizados por el equipo de QA, como la creación de casos de pruebas y su ejecución se utilizó el Addon testrail para Jira.

IV. METODOLOGÍA

A. Integrantes del equipo de QA.

Los integrantes del equipo de aseguramiento de calidad deben seguir tres pilares para realizar muy bien su trabajo. El equipo de calidad debe conocer muy bien el negocio, debe tener un pensamiento crítico y debe colocarse en los zapatos del cliente [21].

El equipo de aseguramiento de calidad estaba conformado por el líder de QA y dos testers.

El líder es el encargado de comunicar los riesgos actuales existentes en el proyecto al PO (Product Owner) y al Arquitecto. Toda la información es recolectada en las reuniones semanales a través de los testers del equipo. Es el encargado de hacer cumplir el master test plan y de priorizar las actividades que deben ser solucionadas con más urgencia. Es una voz en las decisiones que se tomen de manera general en el proyecto con el PO y el Arquitecto [21].

Los testers tienen como responsabilidad ayudar al equipo y a los clientes a entender los requerimientos, analizar las historias de usuario y crear los casos de prueba, participar en la planeación del sprint, definir los datos que requieren los casos de prueba, ejecutar los casos de pruebas y validarlos y mantener al líder informado de todo lo relevante que suceda en el proyecto [21].

B. Etapas del trabajo del equipo de QA

Para que el trabajo de aseguramiento de calidad se cumpla, se pueden distinguir cinco etapas, entre ellas:

1. *Análisis de requerimientos:* Este es un paso inicial del proyecto el cual el departamento de QA al momento de testear ya lo encuentra planteado y gestionado, en este caso en JIRA.
2. *Elaboración del plan de prueba:* El PO y el líder de QA se encargan de establecer el plan de prueba que va a ser ejecutado por los testers.
3. *Elaboración de casos de prueba:* esta actividad es realizada por los testers, en este proyecto se crearon diferentes casos de prueba funcionales y de interfaz de usuario durante cada sprint. En el último sprint del proyecto se crearon casos de prueba end to end y la suite de regresión.
4. *Ejecución de casos de prueba:* al momento que los desarrolladores terminen una tarea y se encuentre el desarrollo en el ambiente de prueba, se empieza a ejecutar los casos de prueba referente a esas tareas. En este proyecto se ejecutaron pruebas de una feature, o de una parte más grande del sistema la cual esta compuesta por subtareas.
5. *Por último, se genera un reporte:* Al final se mide por medio de unas métricas como fue el desempeño del equipo de trabajo, a que se deben los problemas y en que se puede mejorar [9].

C. Actividades durante cada sprint.

Para poder alcanzar los objetivos propuestos se organizaron las actividades por medio del marco de trabajo scrum de la siguiente manera:

1. Revisar las historias de usuario y los respectivos criterios de aceptación referentes a la creación de páginas de contenido (Página de inicio, etc) para hacer el análisis y diseño de los escenarios y casos de prueba antes de ser programados por el equipo de desarrollo.

Estas actividades fueron ejecutadas en el Sprint 1 y el Sprint 2.

2. Revisar las historias de usuario y los respectivos criterios de aceptación referentes a la creación del módulo de búsqueda y navegación de la página para hacer el análisis y diseño de los escenarios y casos de prueba antes de ser programados por el equipo de desarrollo.

Estas actividades fueron ejecutadas en el Sprint 3 y el Sprint 4.

3. Revisar las historias de usuario y los respectivos criterios de aceptación referentes a la creación del módulo del carrito y del proceso de compras en línea para realizar el análisis y diseño de los escenarios y casos de prueba antes de ser programados por el equipo de desarrollo.

Estas actividades fueron ejecutadas en el Sprint 5 y el Sprint 6.

Las actividades que se realizaron en cada sprint son:

Para la creación de casos de prueba.

- Participar en la planeación de los sprint.
- Analizar las historias de usuarios y requerimientos para la creación de los casos de prueba.
- Definir requisitos de datos de prueba.
- Documentar casos de prueba de manera adecuada y referenciarlos adecuadamente a los tiquetes en Jira por medio del Add-on Testrail.
- En cada sprint se analizó y diseñó los casos de prueba para el siguiente sprint.

Para la ejecución de casos de prueba.

- Se ejecutaron los casos de prueba previamente diseñados y documentados.
- Se crearon tiquetes en JIRA en caso de encontrar un bug, con las pruebas y documentación pertinente para que fuera solucionado por el equipo de desarrollo y debidamente referenciado con el caso de prueba creado en Testrail, cumpliendo con la trazabilidad.
- En caso de que al momento de ejecutar la prueba no hubiera ningún problema, se documentaba en test rail y los tiquetes correspondientes en Jira se pasaban al estado pertinente.

- Una vez se solucionaba el defecto se debía probar nuevamente el escenario o caso de prueba que falló.
- Se mantuvo al líder informado de cómo iban las pruebas y las actividades de calidad.
- Asegurar que las actividades de prueba se completen según las expectativas.

Al final del proyecto

- Se ejecutó la suite de regresión donde se encontraban los casos de prueba funcionales y de interfaz de usuarios más relevantes ejecutados durante el proyecto.
- Se ejecutaron los casos de prueba end to end.

V. RESULTADO Y ANÁLISIS

Las métricas utilizadas para la medición de resultados consisten en los casos de prueba realizados, los casos de prueba que fueron ejecutados, los casos de prueba que no fueron ejecutados (**Untested** – No fueron posibles ejecutarse porque la funcionalidad no ha sido completada) lo casos de prueba que pasaron (**Passed** - Al momento de ser ejecutados cumplieron el requerimiento), los casos de pruebas que fallaron (**Failed** – Alguno de los requerimientos descritos en el caso de prueba no se cumplieron o todos), los casos de prueba que se encuentran bloqueados (**Blocked** – Cuando los defectos de otra funcionalidad bloquea estos casos de prueba, o problemas del ambiente, etc), y por último el estado de re-ejecución (**Re-tested** – Cuando un casao de prueba a fallado mas de una vez y se ha tenido que reportar y re-ejecutar varias veces).

A. Sprint 1

Para el primer Sprint se crearon alrededor de 100 casos de prueba, pero por problemas con la estabilidad de la documentación de los requerimientos y la claridad que no tuvieron los desarrolladores, se tuvo que rehacer mucho del trabajo, al final el equipo de QA solo pudo ejecutar 38 casos de prueba en el primer sprint, los cuales corresponden a los resultados mostrados en la tabla y figura 1.

	Passed	Blocked	Retested	Failed
Casos de prueba	26	0	1	12
Porcentaje %	68	0	3	32

Tabla 1. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 1.

Una gran cantidad de casos de prueba fallaron, el 32%, mas de ¼ de la cantidad total. Solo el 3% se tuvo que ejecutar nuevamente después de haber fallado, lo que corresponde a un solo caso de prueba.

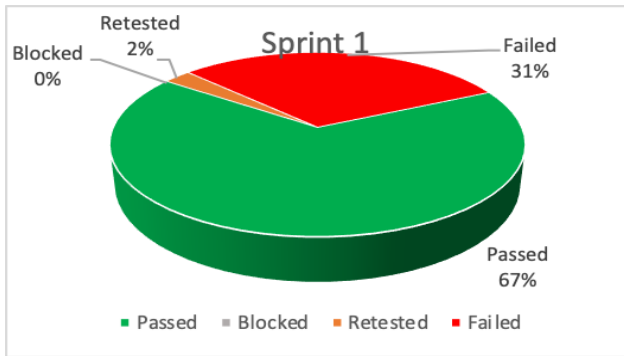


Fig 1. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 1.

B. Sprint 2

Para el segundo Sprint los requerimientos se pudieron ejecutar un total de 89 casos de pruebas de los cuales el 94% pasaron. A pesar de esto el 17% de ese 94% en algún momento fallo, y tuvo que haber sido probado nuevamente. Esto quiere decir que en algún punto del proyecto el 23% de los casos de prueba falló.

	Passed	Blocked	Retested	Failed
Casos de prueba	84	0	15	5
Porcentaje %	94	0	17	6

Tabla 2. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 2.

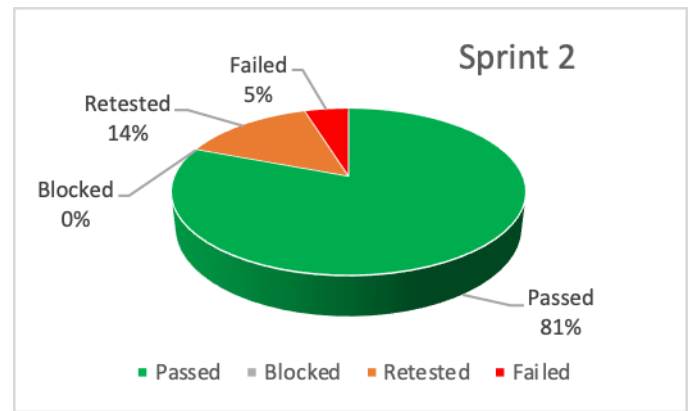


Fig 2. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 2.

Aunque el porcentaje de los casos de prueba fallidos fue menor al primer sprint al finalizar, los casos que se ejecutaron nuevamente mas los casos fallidos suman un 23%, casi ¼ del total los casos de prueba. El problema más relevante de este Sprint es muy parecido a lo sucedido en el primer Sprint, el cual consiste en desconocimiento de los requerimientos. Aunque el desarrollo de las tareas aceleró, se viene con un retraso considerable del primer sprint.

C. Sprint 3

Para el Sprint 3 los casos de pruebas ejecutados equivalen a lo mismo que en el Sprint 2 conincidentalmente, 89 casos de prueba. Para este sprint, los casos fallidos durante el sprint fueron 20%, 3% menos que el sprint anterior, lo que muestra un mejoramiento muy pequeño, pero significativo.

	Passed	Blocked	Retested	Failed
Casos de prueba	86	0	19	3
Porcentaje %	97	0	21	3

Tabla 3. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 3.

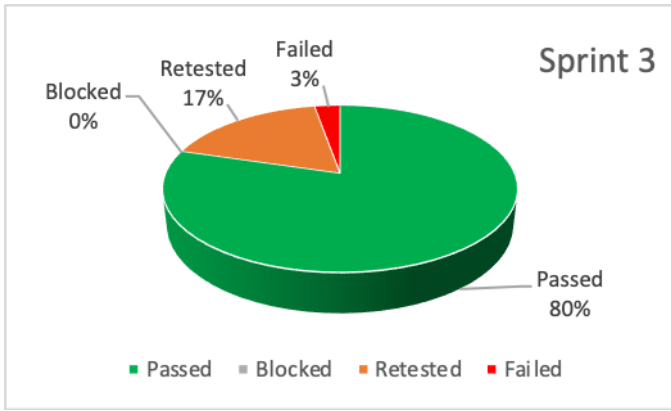


Fig 3. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 3.

Este pequeño mejoramiento puede sugerir un mejor cumplimiento del equipo de trabajo y una mejor definición y entendimiento de los requerimientos.

D. Sprint 4

	Passed	Blocked	Retested	Failed
Casos de prueba	47	0	13	8
Porcentaje %	85	0	24	15

Tabla 4. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 4.

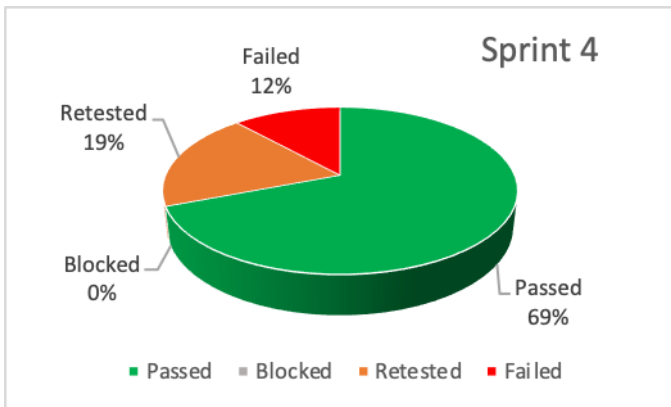


Fig 4 Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 4.

Para el Sprint 4 los casos fallidos durante el proyecto fueron del 39%, pasa el 1/4 de los casos de prueba testeados. El inconveniente más relevante para este Sprint sigue siendo el retraso del proyecto que se acumuló desde el primer sprint.

E. Sprint 5

	Passed	Blocked	Retested	Failed	Untested
Casos de prueba	76	0	36	1	2
Porcentaje %	96	0	46	1	3

Tabla 5. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 5.

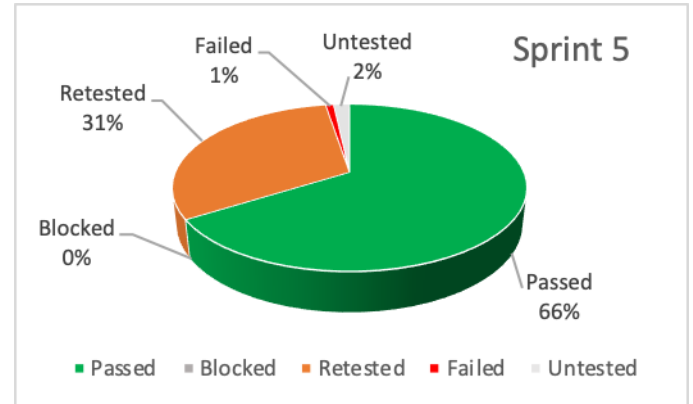


Fig 5. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 5.

En el Sprint 5 y el Sprint 6 fue donde se crearon las funcionalidades más importantes, entre ellas el checkout. Durante este sprint también se puede observar más de 1/4 de los casos de pruebas fallidos durante el sprint, para un total del 47%, casi 2/4 o la mitad de casos de prueba ejecutados. Aunque al final del sprint solo se cerró un caso de prueba, hasta el momento este fue el sprint más crítico en el desempeño del equipo de desarrollo. Para este análisis contamos con una nueva variable que muestra la cantidad de casos de prueba que no se alcanzaron a ejecutar, en este caso fue del 2%.

Como se crearon funcionalidades importantes, y que tienen relación entre sí, al hacer cambios en una se dañaban otras funcionalidades. Esta es la razón por la cual, durante este sprint, casi la mitad de casos de prueba falló.

F. Sprint 6

Los casos de prueba de este último Sprint fueron ejecutados en las semanas de UAT. El 39% de los casos de prueba fallaron en algún momento durante el proceso, el 40% pasaron y el 13% falló. En este punto

del proyecto los defectos se acumulan, porque el usuario ya está verificando.

	Passed	Blocked	Retested	Failed	Untested
Casos de prueba	48	13	31	16	42
Porcentaje %	40	11	26	13	35

Tabla 6. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 6.

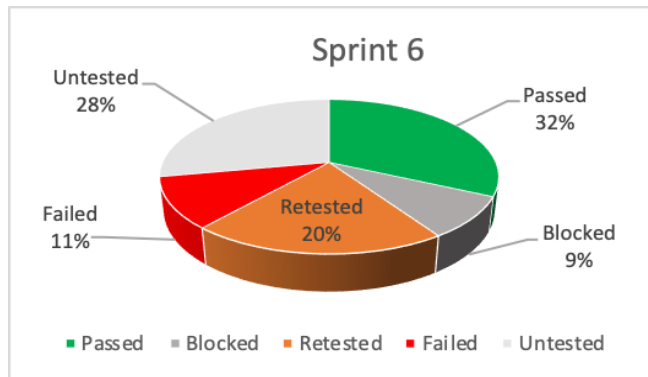


Fig 6. Muestra el resultado de las métricas tomadas para los casos de prueba en el sprint 6.

G. Regresión

Durante el Sprint 5 y el Sprint 6 el mayor problema que tuvo el proyecto fue los daños que ocurrían en otras funcionalidades cuando se hacían cambios en otras funcionalidades. Por tal motivo la ejecución de estos casos de prueba se hizo en la última semana. Para las pruebas de regresión se tuvo un total de 131 casos de prueba de los cuales el 81% cumplieron con los requerimientos, y en algún punto de la ejecución de esta lista el 12% fallaron.

	Passed	Blocked	Retested	Failed	Untested
Casos de prueba	110	4	4	12	5
Porcentaje %	84	3	3	9	4

Tabla 7. Muestra el resultado de las métricas tomadas para los casos de prueba ejecutados en las pruebas de regresión.

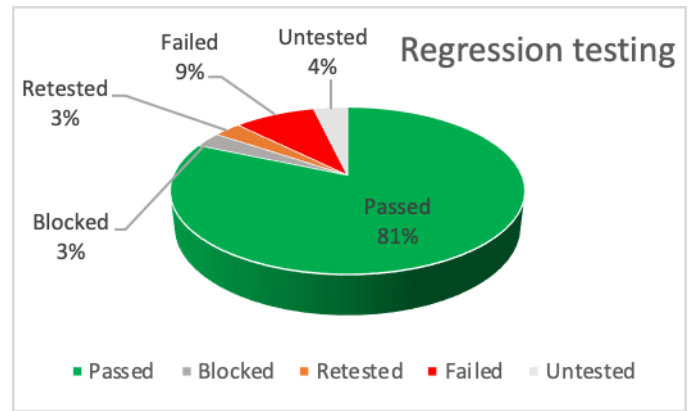


Fig 7. Muestra el resultado de las métricas tomadas para los casos de prueba ejecutados en las pruebas de regresión.

H. End to end

	Passed	Blocked	Retested	Failed
Casos de prueba	16	0	2	0
Porcentaje %	100	0	13	0

Tabla 8. Muestra el resultado de las métricas tomadas para los casos de prueba ejecutados en las pruebas end to end

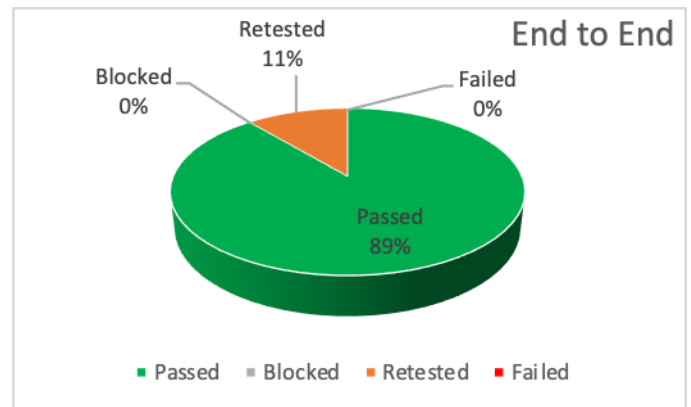


Fig 8. Muestra el resultado de las métricas tomadas para los casos de prueba ejecutados en las pruebas end to end

En el momento que se enecutaron estos casos de prueba, la funcionalidad del checkout ya era aceptable y todos los defectos encontrados ya estaban reportados. Se hicieron un total de 16 casos de prueba de los cuales en algún punto de la ejecución 2 de ellos fallaron.

VI. CONCLUSIONES

Entre las situaciones de riesgo que se encontraron durante el desarrollo de este proyecto se encuentran:

- Al empezar el desarrollo del sitio no se tenían los estilos correctos aprobados por el cliente. Esto causó que se trabajara más de una vez en un mismo requerimiento y atrasó el proyecto desde el principio.
- Muchas veces los desarrolladores terminaban sus tareas sin tener un conocimiento completo de los requerimientos del usuario. Muchos de los defectos levantados fueron por esta razón.
- Durante un sprint hubo una confusión en los cambios de los estados de los tiquetes de Jira por parte del PO y el arquitecto, lo cual atrasó al equipo de QA para el testeó. De este problema surgió una mayor independencia para el equipo de QA, para cambiar los estados de las Features.
- El acoplamiento del equipo. En el primer sprint aparte del problema en los requerimientos, trabajar con personas diferentes requiere de tiempo para tomar confianza y trabajar en equipo.
- El ingreso al equipo de trabajo de algunos desarrolladores sin mucha experiencia.
- Algunos desarrolladores haciendo los cambios necesarios en código para cumplir con sus tareas programadas dañaban una funcionalidad que estaba trabajando.
- Cambios a última hora del cliente.
- Falta de análisis de algunos requerimientos del cliente, los cuales no se adaptaban a la forma en funciona el Framework.

De diferentes formas los aspectos mencionados anteriormente afectaron a que el producto tuviera la cantidad de defectos que tuvo durante todo el desarrollo. Como medidas para mejorar estas situaciones de riesgo se tienen:

- A pesar que trabajamos con metodologías ágiles, es importante que el equipo de desarrollo tenga un grado de experiencia y confianza.
- Antes de empezar el desarrollo se deben tener los requerimientos claros por parte del equipo y plasmados en todas las herramientas pertinentes.
- El equipo que está en contacto con el cliente debe tener un conocimiento muy alto de la tecnología, para poder saber que negociaciones puede acordar con el cliente.

- Al hacer el ingreso de nuevos integrantes al grupo se debe hacer la contextualización pertinente del proyecto.
- En el momento de hacer cualquier cambio en la configuración del gestor de tareas comunicarlos a las áreas pertinentes.

Estos puntos de riesgo y soluciones pueden ser de ayuda para tener en cuenta en otros proyectos, de esta manera se puede mejorar la calidad de los procesos de desarrollo en el área de producción.

REFERENCES

- [1] M. Cubillos and D. Rozo, "El concepto de calidad: Historia, evolución e importancia para la competitividad", *Ciencia.lasalle.edu.co*, 2009. [Online]. Available: <https://ciencia.lasalle.edu.co/cgi/viewcontent.cgi?article=1170&context=ruls>. [Accessed: 20- May- 2020].
- [2] D. Carrizo and A. Alfaro, "Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico", *Scielo*, 2020. [Online]. Available: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-33052018000100114. [Accessed: 20- May- 2020].
- [3] G. Lompfrey and S. Hernández, "LA IMPORTANCIA DE LA CALIDAD EN EL DESARROLLO DE PRODUCTOS DE SOFTWARE", *Fit.um.edu.mx*, 2008. [Online]. Available: <http://fit.um.edu.mx/CI3/publicaciones/Technical%20Report%20COMP-018-2008.pdf>. [Accessed: 20- May- 2020].
- [4] S. Granada, "Colombia a leader in e-commerce during the coronavirus: Report", *The Bogotá Post*, 2020. [Online]. Available: <https://thebogotapost.com/colombia-a-leader-in-e-commerce-during-the-coronavirus-report/46654/>. [Accessed: 21- May- 2020].
- [5] J. Cárdenas, "Diferencias entre garantía de calidad (QA) y control de calidad (QC) - Novanotio", *Novanotio*, 2020. [Online]. Available: <https://www.novanotio.es/diferencias-entre-garantia-de-calidad-qa-y-control-de-calidad-qc/>. [Accessed: 22- May- 2020].
- [6] J. Zavala Ruiz, "¿Por Qué Fracasan los Proyectos de Software? Un Enfoque Organizacional", *ReasearchGate.net*, 2004. [Online]. Available: https://www.researchgate.net/publication/283546859_Por_Qu_e_Fracasan_los_Proyectos_de_Software_Un_Enfoque_Organizacional#pf8. [Accessed: 22- May- 2020].
- [7] J. Gómez, "Informe del Caos 2015 (Chaos Report 2015) o Cómo de bien o mal fueron los proyectos en el año 2015", *El Laboratorio de las TI*, 2015. [Online]. Available: <https://www.laboratorioti.com/2016/05/16/informe-del-caos-2015-chaos-report-2015-bien-mal-fueron-los-proyectos-ano-2015/>. [Accessed: 22- May- 2020].

- [8] T. Watson, "Tres desastrosos fracasos de gestión de proyectos", *ITM Platform | Projects, Programs & Portfolio*, 2016. [Online]. Available: <https://www.itmplatform.com/es/blog/tres-desastrosos-fracasos-de-gestion-de-proyectos/>. [Accessed: 23- May- 2020].
- [9] Kabernun, "¿Por qué son importantes las pruebas funcionales? - Kibernum Chile", *Kibernum Chile*, 2020. [Online]. Available: <https://www.kibernum.com/noticias/por-que-son-importantes-las-pruebas-funcionales-2/>. [Accessed: 23- May- 2020].
- [10] [10]4rSoluciones, "UX y UI: ¿cuáles son las diferencias? | 4R Soluciones | Diseño, Desarrollo y Programación Web & Mobile", *4R Soluciones | Diseño, Desarrollo y Programación Web & Mobile*, 2020. [Online]. Available: <https://www.4rsoluciones.com/blog/ux-y-ui-cuales-son-las-diferencias-2/>. [Accessed: 23- May- 2020].
- [11] Anna, "Test de Regresión en un escenario Ágil.", *Hablamos de calidad del software*, 2018. [Online]. Available: <http://www.hablamosdecalidad.com/test-de-regresion-en-un-escenario-agil/>. [Accessed: 23- May- 2020].
- [12] Quer Barberá, C. (2017). *Implantación de una plataforma eCommerce basada en SAP Hybris para una empresa multinacional del sector servicios* (Maestría en Sistemas de informática). Universitat Politecnica de Valencia.
- [13] C. Díaz, "Qué es un e-commerce: tipos de negocios y pasos para crearlo", *Doofinder Site Search*, 2020. [Online]. Available: <https://www.doofinder.com/es/blog/que-es-e-commerce>. [Accessed: 27- Jan- 2020].
- [14] Talos Digital, *1.02 Installing SAP Commerce.*, 1st ed. Medellin, 2020.
- [15] Talos Digital, *Test Plan*, 1st ed. Medellin, 2020.
- [16] J. alcalde, "Metodologías ágiles (Scrum) - Definición, qué es y concepto | Economipedia", *Economipedia*, 2020. [Online]. Available: <https://economipedia.com/definiciones/metodologias-agiles-scrum.html>. [Accessed: 25- May- 2020].
- [17] Sap Hybris, "Sap Hybris Cloud for Customers Users Stories.", *Help.sap.com*, 2020. [Online]. Available: https://help.sap.com/doc/3b8d1a7fe4654281b0b1a9c0b50502d4/1802/es-ES/C4C_SolutionOverview_ES.pdf. [Accessed: 25- May- 2020].
- [18] E. Romero Frías, "Jira Software - Una aplicación para el seguimiento de errores web - Esteban Romero", *Esteban Romero*, 2020. [Online]. Available: <http://estebanromero.com/herramientas-emprender-desarrollar-proyectos/jira-software-una-aplicacion-para-el-seguimiento-de-errores-web/>. [Accessed: 28- May- 2020].
- [19] M. Vargas, "Casos de Prueba y/o Escenarios", *Los Andes Training*, 2020. [Online]. Available: <https://losandestraining.com/2017/10/19/casos-de-prueba-yo-escenarios/>. [Accessed: 28- May- 2020].
- [20] M. Setter, "User Acceptance Testing – How To Do It Right!", *Usersnap.com*, 2020. [Online]. Available: <https://usersnap.com/blog/user-acceptance-testing-right/>. [Accessed: 24- May- 2020].
- [21] Talos Digital, *Roles and Responsibilities*, 1st ed. Medellin, 2020.