

AUTOMATIZACIÓN DE CASOS DE PRUEBA PARA MEJORAR EL PROCESO
DE CALIDAD DE SOFTWARE.

OLGA LUCIA CÁRDENAS BÁEZ

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.
2016

AUTOMATIZACIÓN DE CASOS DE PRUEBA PARA MEJORAR EL PROCESO
DE CALIDAD DE SOFTWARE.

OLGA LUCIA CÁRDENAS BÁEZ

Trabajo de grado para optar al título de INGENIERO DE SISTEMAS

Director:
AUGUSTO JOSE ANGEL MORENO

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ D.C.
2016

Nota de aceptación

Firma del Presidente del Jurado

Firma del Jurado

Firma del Jurado

Bogotá, Agosto 25 de 2016.

Las directivas de la Fundación Universitaria los Libertadores, los jurados calificadores y el cuerpo docente no son responsables por los criterios e ideas expuestas en el presente documento. Estos corresponden únicamente al autor

DEDICATORIA

A Dios por proteger y bendecir cada día de mi vida.

A mi hijo por infundir en mí ese gran deseo de superación, por su gran apoyo y compañía.

A mis padres quienes me apoyaron siempre con actitud de grandeza y lo más importante por enseñarme a luchar por cumplir mis sueños.

A Pedro Manuel Ceron quien lleno mi vida de alegría y amor en todos aquellos momentos difíciles, por su paciencia y ayuda profesional.

AGRADECIMIENTOS

Quiero ofrecer mis más sinceros agradecimientos a:

El Ingeniero Augusto José Ángel Moreno, quien me apoyó durante todo el proceso pasantía organizacional.

A los docentes de la Fundación Universitaria los Libertadores, quienes formaron las bases de mi conocimiento para desarrollar este proyecto, especialmente al docente Julián Alberto Castillo, por su esfuerzo, conocimientos, paciencia y valiosos consejos logre finalizar mis estudios.

A ATH la empresa que permitió que formara mi perfil profesional desde el inicio de mi carrera con Ingeniera de Sistemas.

A PayU la empresa que me brindó la oportunidad de participar en su proyecto y así colocar en práctica mi formación profesional.

Son verdaderamente muchas las personas que tienen un gran mérito en mi vida profesional gracias a su apoyo, consejos y sobre todo su amistad.

TABLA DE CONTENIDO

	pág.
RESUMEN	9
ABSTRACT	11
1. INTRODUCCION	14
2. PLANTEAMIENTO DEL PROBLEMA.....	15
3. JUSTIFICACIÓN	16
4. ALCANCE Y LIMITACIONES	17
5. OBJETIVOS	18
5.1 <i>Objetivo General</i>	18
5.2 <i>Objetivos Específicos</i>	18
6. MARCO TEORICO.....	19
6.1 <i>Antecedentes Históricos</i>	19
6.2 <i>Pruebas de software</i>	20
6.3 <i>Tipos de Ejecución de Pruebas de software</i>	21
6.4 <i>Automatización</i>	22
7. INGENIERÍA DEL PROYECTO.....	24
7.1 <i>Descripción de la situación actual</i>	24
7.2 <i>Requerimientos de la información</i>	24
7.3 <i>Modelamiento del sistema</i>	25
7.4 <i>Descripción del sistema</i>	31
8. EVALUACIÓN ECONOMICA DEL PROYECTO.....	33
9. BENEFICIOS DE LA IMPLEMENTACIÓN.....	36
9.0 <i>Operacionales</i>	36
9.1 <i>De Gestión</i>	36
9.2 <i>Estratégicos</i>	36

9.3 De Infraestructura	37
9.4 De IT	37
10. PRESUPUESTO	38
10.1 Costo de Infraestructura Física	38
10.2 Costo de Recursos Humanos	38
10.3 Costo Total del Proyecto.....	38
11. CRONOGRAMA.....	39
12. CONCLUSIONES	41
13. RECOMENDACIONES	42
14. GLOSARIO	43
15. BIBLIOGRAFIA	45

LISTA DE IMAGENES

	pág.
Ilustración 1. PayU Pagos Electrónicos	19
Ilustración 2. Modelo V de desarrollo.....	20
Ilustración 3. Marco Scrum	26
Ilustración 4. Roles Scrum	27
Ilustración 5. Tablero de tareas Scrum	29

LISTA DE ANEXOS

Anexo 1. Informes de Pasantía Organizacional

Anexo 2. Código Parcial Scripts Automatizados

RESUMEN

Las empresas que se dedican al desarrollo de software han encontrado un factor en común que ralentiza de gran manera el proceso del mantenimiento sobre aplicaciones altamente acopladas y complejas, este factor es el proceso de pruebas.

Las Pruebas de Software se han venido convirtiendo en el “cuello de botella”, debido a la capacidad de cobertura que puede ofrecer los Tester sobre grandes suites de casos de pruebas, por tanto, es necesario generar procesos eficientes con altos estándares de calidad.

Nuestro medio es cada día más avasallador y exigente, es por esto que las empresas buscan tener ventajas tecnológicas y metodológicas que logren superar a su competencia. Las personas ajenas a el área de la tecnología, exigen soluciones de acuerdo a sus necesidades sin tener en cuenta la complejidad de los procesos. Es por esto que las fábricas de software deben encontrar la manera para poder sobresalir en el competitivo mundo comercial y mantenerse a la vanguardia del mercado.

Una de las necesidades que se puede observar claramente en las solicitudes de software es el tiempo de entrega, y este es el aspecto en que las grandes empresas que han logrado estar en la vanguardia de las tendencias tecnológicas generan estrategias en sus procesos con el fin de dar soluciones rápidas.

En el marco de la industria de los pagos en línea, se están enfrentando día a día a mayores retos de seguridad, fiabilidad, calidad y la eficiencia en procesos transaccionales y cada nueva estrategia de desarrollo técnico y/o funcional trae consigo la responsabilidad de mantener una plataforma adecuada para generar satisfacción en los clientes.

El proceso de pruebas, sobre todo en la fase de regresión, es la mejor manera de evaluar la calidad de cada componente de software, pero así mismo es un proceso que debe tomar un tiempo prudencial para cumplir su objetivo. La automatización de estos procesos de negocio es fundamental para disminuir estos tiempos, generando garantías de estabilidad y funcionamiento de los intercambios de información generados durante los pagos electrónicos.

Durante este proyecto se busca optimizar el proceso de pruebas de regresión de los procesos ya desarrollados y en proceso constante de mantenimiento, por medio de la automatización de casos de prueba funcionales, para herramientas que ya se encuentran estables y productivas.

Palabras Claves: Proceso de pruebas, calidad, tecnología, procesos, seguridad, fiabilidad, eficiencia, pruebas de regresión, automatización, pagos electrónicos, optimizar, pruebas funcionales.

ABSTRACT

Companies dedicated to software development have found a common factor that slows down the process of maintenance on highly coupled and complex applications, this factor is the testing process.

Software testing have been turning in the "bottleneck" because of the ability of coverage that can offer the Tester on large suites of test cases, therefore, it is necessary to create efficient processes with high quality standards.

Our environment is daily, even more domineering and demanding, this is why companies seek to have technological and methodological advantages to achieve get over their opponents. People outside of the technology area, demand solutions according to their needs regardless the complexity of the processes. That is why software factories must find ways to excel in the competitive commercial field and stay ahead of the market.

One of the needs that can be clearly seen in request of software is the release day, and this is the aspect that large companies have consider to be at the forefront of technology trends generate strategies in their processes in order to give quick solutions.

In the industry of online payments, every day they are facing challenges of security, fiability, quality and efficiency in transactional processes and every new strategy of technical development and / or functional brings with it the responsibility to maintain a suitable platform to generate customer satisfaction.

The testing process, particularly in the regression phase, is the best way to assess the quality of each software component, but also it is a process that should take a reasonable time to accomplish its target. The automation of these business processes is essential to reduce these times, generating guarantees of stability and functioning of the exchange of information generated during electronic payments.

In this project the goal is to optimize the process of regression testing processes on developed software and software that are in constantly maintenance process, through automation of functional test cases for tools that are already stable and productive.

Keywords: Testing process, quality, technology, processes, security, fiability, efficiency, regression testing, automation, electronic payments, optimize, functional testing.

1. INTRODUCCION

Las empresas de comercio electrónico son “proveedores de servicios de pagos en línea”¹, un “intermediario” entre el comprador (pagador) y el comerciante (recaudador), con el propósito de generar pagos de manera rápida, sencilla y eficiente mediante diferentes tipos de dispositivos, por tanto, esta empresa de pagos en línea con un gran tiempo en la industria del e-commerce, contribuye al posicionamiento del mercado en la red y la economía digital en Colombia.

Pero detrás de todo este inmenso mundo digital (transacciones, clientes, compradores, dispositivos...) nos encontramos con el trabajo de quienes hacen posible esta solución de pagos. Las herramientas desarrolladas por el área de tecnología en esta multinacional dedicada al e-commerce es la maquinaria que garantiza cada uno de los negocios online de sus comerciantes y para mantener una estrategia comercial en crecimiento debe alinearse a las solicitudes de sus clientes sin dejar atrás las soluciones ya productivas.

El equipo de tecnología se enfrenta a un gran reto día a día para generar soluciones que se adapten fácilmente a los mercados digitales actuales, realizando mantenimiento a los componentes de software ya desarrollados, mejorando su desempeño, incluyendo funcionalidades o migrando a nuevas versiones con más cualidades como usabilidad, portabilidad, fiabilidad entre otras.

La empresa de pagos Online, ha implementado una estrategia que evite que el desarrollo ágil (Scrum) de aplicaciones deje por fuera procesos de gran importancia como el Testing, el cual esta organización ha logrado optimizar por medio de automatización de casos de prueba.

¹ PayU, Líderes alrededor del mundo. [En línea]. [Consultado el 23 de Julio de 2016] Disponible en: <https://www.payu.com.co/sobre-payu>.

2. PLANTEAMIENTO DEL PROBLEMA

En la actualidad la Empresa de pagos online cuenta con un proceso de Testing muy costoso a nivel de tiempos por la gran cantidad de desarrollos nuevos y mantenimientos sobre funcionalidades que requieren pruebas de carga, regresión y aceptación entre otras, y a pesar de que cuentan con una infraestructura ya definida y analistas de pruebas con capacidades para realizar la labor de automatización, no ha logrado encontrar un punto de equilibrio entre el tiempo que debe dedicar para la automatización los casos de prueba de regresión, mantenimiento de la suite automatizada y los casos manuales del Sprint actual, lo cual lleva a:

- Doble Codificación sin control: Las suites automatizadas no se mantienen con los cambios realizados durante el sprint y existen varios casos duplicados.
- Retrocesos en procesos de pruebas: Aquellos casos ya automatizados que no se actualizaron durante los sprint quedan sin funcionar y los analistas de pruebas deben volver al proceso de pruebas manuales.
- Improductividad: Se da por perdido el tiempo y esfuerzo del analista de pruebas o desarrollo que automatizo la prueba si después de un sprint no se actualiza y se retorna a la prueba manual.
- Clientes Insatisfechos: Debido a la gran cantidad de casos de prueba que se deben realizar manuales, algunas funcionalidades quedan sin un cubrimiento de pruebas adecuado y se pueden inyectar errores en los ambientes productivos.
- Perdidas Económicas: Para poner en marcha el proceso de automatización se requirió una inversión organizacional y de infraestructura que debe ser aprovechada.

3. JUSTIFICACIÓN

Las empresas dedicadas al mercado digital fomentan el comercio electrónico garantizando a sus clientes la tranquilidad de realizar transacciones en línea con sus productos bancarios, con costo por transacción de acuerdo a ciertos convenios realizados entre el comercio y la empresa.

Los pagos en línea son un negocio que maneja información confidencial de los vendedores y compradores, y lo más importante su dinero, de ahí que estos esperan un servicio con altos estándares de calidad y seguridad, lo que lleva a esta entidad a generar procesos estrictos de desarrollo y pruebas de estas herramientas suministradas a los comercios para conectarse con las entidades bancarias.

Las empresas de comercio electrónico están obligadas a establecer normas para darle cumplimiento a la ley de protección de datos (PCI), con el fin de asegurar legalmente la información que viaja en las tramas de sus transacciones. Y por tanto todas las áreas de la compañía deben encontrarse alineadas a cumplir y ofrecer estos estándares de seguridad. El área de calidad debe mantener un alto nivel de desempeño para disminuir la posibilidad de ocurrencia de factores que pongan en riesgo la información de sus clientes.

Actualmente los procesos de prueba se alternan entre los casos manuales y automatizados, pero el enfoque de la metodología es mantener el 80% de la suite de regresión automatizada y las pruebas manuales sean de aquellas funcionalidades en proceso de desarrollo y casos de prueba de gran complejidad. Este cubrimiento en el proceso de aseguramiento de calidad lograría mantener la confianza que brinda a sus clientes.

Para lograr este objetivo, los equipos del área de tecnología deben incluir puntos dentro de su planeación para automatizar las pruebas de regresión y evitar que la deuda técnica de casos continúe en crecimiento, por mi parte en conjunto con un equipo de analistas de pruebas planteamos una estrategia para mantener todos aquellos casos que ya se encuentran automatizados, pero presentan fallos e iniciar el proceso de disminución de la suite de regresión manual.

Con este plan emergente se reactivará y fomentará el progreso de las pruebas automatizadas, la capacidad del grupo de calidad y la velocidad de los equipos Scrum dentro de la compañía. Con el gran aporte de confianza que brindan las pruebas de regresión a los procesos de software, y al encontrarse migradas a un sistema automatizado en su mayoría y complementadas con pruebas complejas manuales se lograra una cobertura bastante amplia.

4. ALCANCE Y LIMITACIONES

El proyecto de automatización en la empresa de Pagos Online ya se encuentra en curso por tanto nuestro alcance radica en:

1. Identificar y mantener los casos de prueba que actualmente tienen errores durante la ejecución. Se tiene un total de 25 casos ya mezclados en la rama principal, pero con inconvenientes en la ejecución y que actualmente se realizan manualmente.
2. Disminuir la deuda técnica de la suite de Regresión del equipo dedicado a las integraciones con las redes dentro de la empresa (Payments Methods). En la actualidad se tiene un atraso de 76 casos de prueba pendientes por automatizar únicamente para este equipo.

Los eventos que podrían convertirse en limitaciones para este proyecto son:

1. Los analistas dedicados al proyecto tienen la responsabilidad de realizar las pruebas del Sprint y adicionalmente la automatización de pruebas. A pesar de tener una planeación definida, si alguno de los desarrollos comprometidos en el Sprint requiera varios ciclos de pruebas se verá afectado el tiempo de la automatización ya que para la empresa es prioridad asegurar los componentes en procesos.
2. Los casos de prueba que se encuentra funcionando inadecuadamente requieren demasiado tiempo para el análisis, lectura y mantenimiento del código.
3. Posibles inconsistencias en los datos utilizados para los casos de prueba automatizados, el análisis y revisión de estos errores demandan una gran cantidad de tiempo.

5. OBJETIVOS

5.1 Objetivo General

Implementar una estrategia para mantener los casos de prueba automatizados y disminuir el número de casos de la suite de regresión que actualmente se ejecutan de forma manual.

5.2 Objetivos Específicos

- Analizar un plan de acción que permita optimizar la automatización de casos de prueba dentro del transcurso del Sprint.
- Implementar una estrategia para mantener los casos de prueba automatizados que actualmente presentan fallas.
- Definir un estándar de codificación de casos de prueba automatizados para analizar y mantener con facilidad.
- Analizar la complejidad de los casos de prueba de la suite de regresión con el fin de seleccionar los casos que se automatizaran.
- Realizar los procesos de automatización de manera ordenada y dentro de los tiempos establecidos
- Automatizar casos de prueba de la suite de regresión.
- Generar una disminución de la deuda técnica en cada uno de los Sprint.

6. MARCO TEORICO

6.1 Antecedentes Históricos

Este proveedor de pagos en línea. Es una multinacional con “presencia en 16 mercados de alto crecimiento, posicionándolos como expertos en los mercados locales”², esta constituye la división del grupo Naspers fundado desde 1915 el cual tiene presencia en 133 países, el negocio de PayU consiste en el comercio electrónico dando gran importancia a la manera en que los consumidores compran y pagan para ofrecer un método de pago rápido, sencillo y seguro, ubicada en el sector norte de la ciudad de Bogotá (Chico). Cuenta con una estructura administrativa y tecnología robusta con un enfoque diferencial frente al ambiente laboral. Es una gran empresa con una infraestructura física para sus áreas administrativas, comerciales y de tecnología



Ilustración 1. PayU Pagos Electrónicos

Tomado: DigitalBox. Como recibir pagos electrónicos. 2016. <http://digitalbox.com.co/como-recibir-pagos-electronicos/>

PayU Actualmente brinda diferentes servicios para utilizar el api de pagos la cual realiza cobro de comisión por transacción, los productos³ que ofrece a los comercios son:

- Integración API.
- Integración Web Checkout.
- SDK - Software Development Kits.
- Botones de Pago.
- Carritos de Compra.
- Tokenización.
- Terminales Virtuales.

² PayU, Líderes alrededor del mundo. [En línea]. [Consultado el 23 de Julio de 2016] Disponible en: <https://www.payu.com.co/sobre-payu>.

³ PayU, Developers. [En línea]. [Consultado el 23 de Julio de 2016] Disponible en: http://developers.payulatam.com/es/solutions/payment_request.html.

6.2 Pruebas de software

Corresponde al conjunto de actividades dedicadas a realizar aseguramiento de calidad sobre componentes de software, determinando su correcto funcionamiento de acuerdo a lo requerido. “Este proceso puede suponer el 40% del coste total del de desarrollo del proyecto”⁴.

Existen varios tipos de pruebas de acuerdo a cada una de las etapas del modelo de desarrollo, para el Modelo V se tienen en cuenta las siguientes:

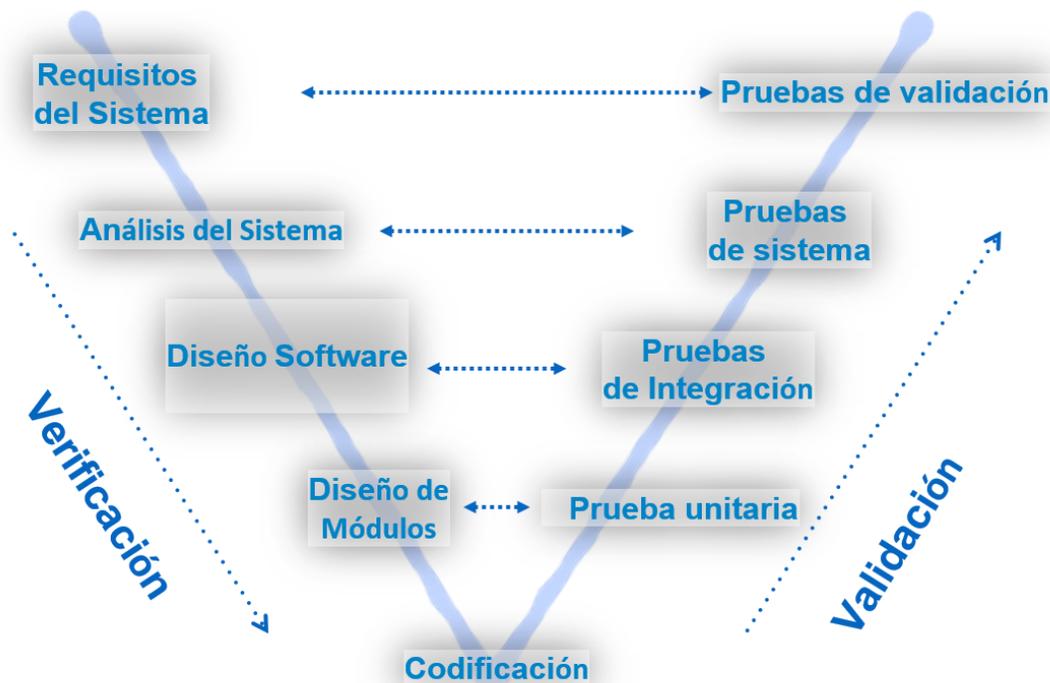


Ilustración 2. Modelo V de desarrollo

Tomado: Tutorialspoint. Software - Ciclo de Vida de Desarrollo.
http://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm

- Pruebas de Aceptación: “Pruebas de aceptación realizadas por usuarios/clientes en sus respectivas dependencias, con el objeto de determinar si un componente o sistema satisface las necesidades de usuarios/clientes y se ajusta al proceso de negocio, incluyendo normalmente tanto software como hardware.”⁵

⁴ ALONSO, Fernando. MARTINEZ Loic. y SEGURA Fco. Javier. Introducción a la Ingeniería de Software. Madrid. 2005 p. 87

⁵ SSTQB, Glosario Estándar de Términos Utilizados en Pruebas de Software. [En línea]. [Citado el 07 de julio de 2008] Disponible en: http://www.sstqb.es/ficheros/sstqb_file94-5205ea.pdf

- Pruebas de Sistema: Pruebas realizadas desde el punto de vistas del usuario y con un entorno de prueba igual o similar al productivo, dentro de estas se realizan una serie adicional de pruebas basadas en los riesgos y/o funcionalidades especificadas en los requerimientos.

En la fase de pruebas de sistemas debemos ejecutar algunas suites de pruebas que garantizan el entorno de los componentes modificados.

Pruebas de Regresión: “Son pruebas de un programa previamente probado que ha sufrido modificaciones, para asegurarse que no se han introducido o descubierto defectos en áreas del software que no han sido modificadas como resultado de los cambios realizados. Se realiza cuando el software o su entorno han sido modificados.”⁶

- Pruebas de Integración: “Pruebas ejecutadas con el objeto exponer defectos en las interfaces e interacciones entre elementos integrados.”⁷
- Pruebas unitarias: Corresponden a la validación de una pieza, componente, modulo o subprograma específico del software.
- Pruebas de Integración: Verifica la correcta interacción entre dos o más componentes que se desarrollan aisladamente.

6.3 Tipos de Ejecución de Pruebas de software

De acuerdo al tipo de ejecución existen dos tipos de pruebas: Manuales y Automáticas.

- Pruebas Manuales.

Son pruebas realizadas por los analistas de desarrollo o de pruebas que verifican cada funcionalidad del sistema.

⁶ SSTQB, Glosario Estándar de Términos Utilizados en Pruebas de Software. [En línea]. [Citado el 07 de julio de 2008] Disponible en: http://www.sstqb.es/ficheros/sstqb_file94-5205ea.pdf

⁷ SSTQB, Glosario Estándar de Términos Utilizados en Pruebas de Software. [En línea]. [Citado el 07 de julio de 2008] Disponible en: http://www.sstqb.es/ficheros/sstqb_file94-5205ea.pdf

- Pruebas Automatizadas.

Se realizan a través de herramientas que permiten de manera controlada comparar los resultados de un sistema vs los resultados esperados en el diseño de casos de prueba. Estas herramientas se denominan frameworks como Junit, el cual permite la programación de scripts con lenguaje Java.

6.4 Automatización

“El término automatización se refiere a una amplia variedad de sistemas y procesos que operan con mínima, incluso sin intervención, del ser humano”⁸. con el fin de que sean más rápidos, eficientes y precisión.

La automatización cobra importancia ya que reducen el costo de la mano de obra directa, evitando retrasos por fallas humanas.

Existen varios tipos de automatización como:

- Control Automático de Procesos, se refiere a los procesos químicos o físicos.
 - Automatización Fija, Sistemas lógicos o controladores programables.
 - Control numérico computarizado, Robots industriales “Celdas de manufactura flexibles”
 - Automatización flexible, Series de estaciones de trabajo dedicadas al proceso de almacenamiento y manipulación de materiales controlados por computadoras.
 - Proceso Electrónico de Datos, “Frecuentemente relacionado con las tecnologías de la información, centros de costos, etc”⁹. por medio de la obtención, análisis y registro de datos a partir de interfaces de computadoras y pueden re-categorizarse de acuerdo a sistemas automatizados:
1. Sistemas en línea. Acepta datos de entrada y salida de acuerdo a su codificación.

⁸⁸ QuimiNet, ¿Qué es la automatización?. [En línea]. [Citado el 22 de febrero de 2008] Disponible en: <http://www.quiminet.com/articulos/que-es-la-automatizacion-27058.htm>

⁹ UNAD, Tipos de Automatización. [En línea]. [Consultado el 20 de Junio de 2016] Disponible en: http://datateca.unad.edu.co/contenidos/208009/Contenido%20en%20linea/23_tipos_de_automatizacion.html

2. Sistemas en tiempo real. Recibe datos los procesa datos con una rapidez esperada.
3. Sistemas de apoyo a decisiones. Sirven de apoyo para tomar decisiones importantes en una entidad.
4. Sistemas basados en el conocimiento. Sistema experto especializados a una tarea que pueden presentar inconvenientes debido a que es demasiado especializado, y por tanto, no adaptable a cambios de circunstancias, adicionalmente entre más crece el sistema automatizada, más costoso es su mantenimiento.

7. INGENIERÍA DEL PROYECTO

7.1 Descripción de la situación actual

El área de tecnología actualmente cuenta con una división de equipos de trabajo ordenados de acuerdo a la metodología de desarrollo ágil Scrum. Es importante para la organización seguir las buenas prácticas que ofrece dicha metodología para brindar herramientas que se alineen a los avances tecnológicos y necesidades de sus clientes.

Cada equipo cuenta con un analista de pruebas el cual debe encargarse de las pruebas funcionales, no funcionales y automatización de casos de la regresión dentro del tiempo que cursa el sprint.

Sin embargo, debido a la gran cantidad de proyectos en curso en el equipo de Payments Methods se ha venido dejando de lado la tarea de automatización de casos de prueba y por tanto a la fecha se identificó una gran cantidad de casos de prueba funcionales pendientes por automatizar en la suite de regresión.

La idea principal de la inversión en infraestructura y procesos organizacionales para automatización de pruebas era evitar que los analistas de pruebas dedicaran la mayor cantidad de su tiempo en verificar funcionalidades de esos componentes acoplados que realmente no fueron modificados, y por el contrario concentren todos sus esfuerzos en identificar problemas técnicos (rendimiento, seguridad, escalabilidad, portabilidad entre otras)

Es importante identificar una estrategia para aprovechar los recursos adquiridos por la empresa y disminuir la deuda técnica referente a la automatización de pruebas.

7.2 Requerimientos de la información

1. Mantenimiento de Casos de prueba Automatizados.

Se debe realizar el seguimiento del código de cada uno de los casos de prueba automatizados y actualizarlo de acuerdo al proceso y resultado esperado actual de la funcionalidad.

2. Automatización de casos de Prueba

Realizar la automatización de cada uno de los pasos descritos en los casos de prueba funcionales incluidos en la suite de regresión.

3. Reporte de Ejecución de Caso Automatizado

En caso de que el caso de prueba corresponda al envío de tramas se realizara un reporte que contenga los datos del request enviado y el response de la aplicación, con su respectivo resultado de la ejecución.

Por el contrario, si el caso de prueba se realiza sobre acciones de una aplicación se requiere que el reporte contenga un título que describa el caso y el respectivo resultado de la ejecución.

4. Reporte general de Ejecuciones.

La ejecución de cada una de las suites automatizadas genera un reporte consolidado que contiene una métrica de resultados y una tabla con el resultado de cada caso de prueba. Asegurar que todos los casos de prueba implementados registren su estado en el reporte.

7.3 Modelamiento del sistema

Marco Scrum.

- Manifiesto por el Desarrollo Ágil de Software

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre procesos y herramientas
Software funcionando sobre documentación extensiva
Colaboración con el cliente sobre negociación contractual
Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.¹⁰

Kent Beck	Alistair Cockburn	James Grenning	Ron Jeffries	Robert C. Martin
Mike Beedle	Ward Cunningham	Jim Highsmith	Jon Kern	Steve Mellor
Arie van Bennekum	Martin Fowler	Andrew Hunt	Brian Marick	Ken Schwaber
Dave Thomas				

¹⁰ AGILE Manifiesto, Manifiesto por el Desarrollo Ágil de Software. [En línea]. [Citado 2001] Disponible en: <http://www.agilemanifesto.org/iso/es/manifesto.html>

Corresponde a un “marco de gestión para el desarrollo incremental de productos”¹¹, dentro de las metodologías ágiles de software aplicado mediante los valores del manifiesto ágil, que utiliza un conjunto de directrices que deben ser cumplidas por un Sprint Team (equipo de trabajo), mediante 3 roles, 5 reuniones y 3 artefactos.



Ilustración 3. Marco Scrum

Tomado: IKRITWEB. Desarrollo de Software. 2014. <http://ikrit.com.ar/>

- Roles¹²

1. Product Owner

Este rol se hace responsable de la visión del producto, es aquella persona que conoce el negocio, por tanto, es quien realiza la priorización de Historias de Usuario en el Backlog de acuerdo a las deliberaciones con los stakeholders (interesados) y adicionalmente realiza el proceso de aceptación.

2. Scrum Master

El Scrum Master es aquel miembro del equipo que tiene altos conocimientos técnicos y del marco de desarrollo Scrum, para facilitar todo el proceso de desarrollo, resolviendo impedimentos, generando un ambiente de bienestar y

¹¹ SCRUM Reference Card, Acerca de Scrum. [En línea]. [Consultado el 25 de Julio de 2016] Disponible en: <http://scrumreferencecard.com/reference-card-de-scrum/>

¹² SCRUM Reference Card, Acerca de Scrum. [En línea]. [Consultado el 25 de Julio de 2016] Disponible en: <http://scrumreferencecard.com/reference-card-de-scrum/>

orden para su equipo, protegiéndolo de distracciones o interferencias externas, y es responsable de mantener y realizar seguimiento sobre los artefactos Scrum. Sin embargo, no tiene autoridad en la gestión del equipo.

3. Scrum Team (Equipo de Desarrollo de Scrum)

El Scrum Team debe ser multidisciplinario, auto-dirigido y auto-gestionado con un alto nivel de responsabilidad por el cumplimiento y trabajo en equipo, por esto, incluye a miembros con diferentes perfiles que soportan el proceso de desarrollo (analistas: técnicos, de desarrollo, de pruebas, de negocio, de arquitectura entre otros), tiene la capacidad de negociar compromisos con el Product Owner y cumplir con los objetivos propuestos durante cada Sprint.

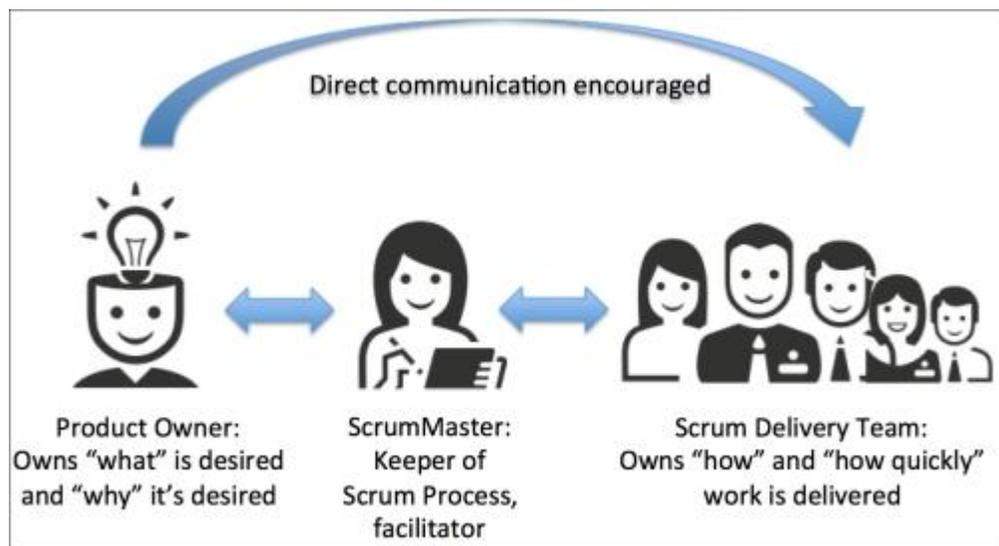


Ilustración 4. Roles Scrum

Tomado: SAFARI. Scrum Roles. <https://www.safaribooksonline.com/library/view/the-professional-scrummasters/9781849688024/ch01s04.html>

- Reuniones

El facilitador de todas las reuniones es el Scrum Master, sin ejercer poder de tomar decisiones.

1. Sprint Planning (Reunión de Planificación de Sprint)

Al iniciar cada una de las iteraciones (Sprint), se reúnen el Product Owner, Scrum Master y Scrum Team, con el fin de negociar la inclusión de historias de usuario debidamente priorizadas dentro del Backlog en el Sprint actual. El Product Owner expone sus prioridades y el equipo determina la velocidad (Puntos a realizar) y realiza la puntuación de cada una de las historias.

Al finalizar la reunión se establece el compromiso de finalización de las historias incluidas dentro del número de puntos incluidos en la velocidad que actualmente maneje el equipo

2. Scrum Daily (Reunión Diaria)

Corresponde a una reunión que se realiza a diario por un tiempo no mayor a 15 minutos, en el que se reportan 3 cosas: Lo que se realizó el día anterior, lo que se está realizando hoy y lo que se hará mañana, además de los impedimentos que ha estado enfrentado cada integrante (el impedimento no se soluciona en la reunión únicamente se expone), es importante que sea a la misma hora y en el mismo lugar (de pie), con todos los miembros del Scrum team, el Scrum Master y el product owner con el fin de que apoye posibles soluciones o ambigüedades que pueda evidenciar el equipo.

3. Sprint Review (Reunión de Revisión del Sprint)

Al finalizar el Sprint, se realiza la respectiva revisión mediante una reunión con una demostración en vivo para con el Scrum Master, Product Owner y a los demás interesados, con el fin de determinar si los compromisos establecidos durante la reunión de planeación se cumplieron. En caso de tener no conformidades la historia se devuelve al Backlog con su respectiva prioridad.

4. Scrum Retrospective (Reunión de Retrospectiva del Sprint)

La reunión de retrospectiva es una reunión que se realiza al finalizar el Sprint para reflexionar sobre el proceso por parte del equipo, verifican los temas buenos y los temas por mejorar, adicionalmente se toman los puntos por mejorar de las reuniones de Retrospectiva anteriores y se verifica si hay mejoras en cada uno de los ítems.

La participación del Scrum Master en esta reunión es crucial ya que dedica su esfuerzo en generar alternativas para crear un ambiente de seguridad psicológica que disminuya inconvenientes entre los participantes.

5. Sprint Backlog Grooming (Reunión de Refinamiento del Backlog)

En esta reunión el equipo estima la cantidad de esfuerzo que podría invertir en cada ítem referenciado en el Product Backlog proporcionando información importante tanto funcional como técnica para su priorización. Se logran identificar ambigüedades o alcances de la solicitud o de los criterios mínimos de aceptación. dentro de las historias de usuario para modificarlas o en su defecto moverlas entre varias Epics (Agrupación de Historias de Usuario).

- Artefactos

1. Product Backlog (Backlog del Producto)

El Backlog del Producto se refiere a la lista ordenada y priorizada de ítems de producto, y de esta salen las agrupaciones de historias de usuario que se convierten en el compromiso de cada sprint.

Esta lista esta compuesta de ideas, características, mejoras, bugs, requerimientos de documentación, cada uno de estos elementos del Backlog del Producto incluye una descripción y una estimación con un único responsable de su mantenimiento, el producto Owner., estos Items pueden originarse a partir del Dueño del Producto, de los miembros del equipo, o de otros interesados y deben tener las siguientes características.

- Escrita en forma de Historia de Usuario.
- Contiene una definición de terminado (DOD)
- Criterios de aceptación específicos.
- Enfoca su contenido en el que hacer, por encima del cómo.

2. Sprint Backlog (Backlog del Sprint)

Artefacto construido con información de la reunión Sprint Planning, corresponde a la elección de ítems del product backlog (PBIs) que se desarrollaran durante el Sprint en curso, adicionalmente contiene la estimación y el alcance comprometido por el equipo del trabajo. Con este artefacto terminado se podrán iniciar las tareas por parte del Scrum Team.

Normalmente se representa como un taskboard (Tablero de tareas) físico, por ejemplo:

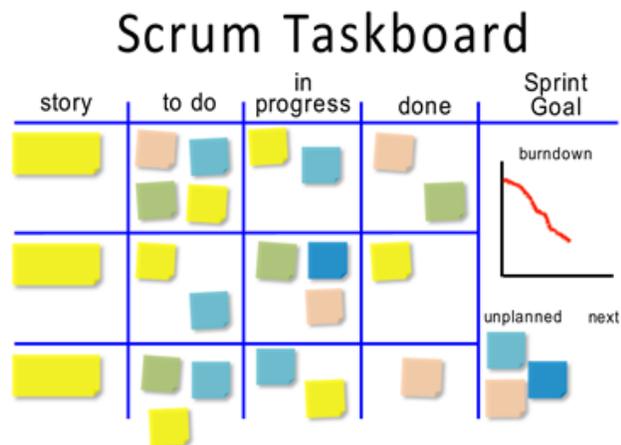


Ilustración 5. Tablero de tareas Scrum

Metodología del Proyecto - Estrategia.

Para incluir el proyecto de automatización dentro del marco Scrum, se realiza la siguiente propuesta metodológica como estrategia, para incluir en cada Sprint los siguientes pasos:

- **Definición de Pruebas:** Se busca ordenar y priorizar los casos de prueba que se encuentran pendientes por automatizar en la suite de regresión manual, Se realiza una evaluación de complejidad de ejecución manual de cada prueba.
Roles: Líder de Pruebas, Scrum Master del Equipo, Analistas de Pruebas.
Soporte: Se diseña un documento que especifica las funcionalidades, criticidad y complejidad de los casos funcionales pendientes por automatizar.
Etapa Sprint: Sprint Planning
- **Definición de los Procedimientos de Prueba:** Después de la priorización de casos de prueba, se busca definir dentro de que suites se ubicarán, lo que implica definir también los scripts que conformarán las pruebas automatizadas y sus dependencias y resultados esperados.
Roles: Líder de Pruebas, Scrum Master.
Soporte: Documento de estructura de suites automatizadas.
Etapa Sprint: Sprint, Tarea de Diseño
- **Generación de scripts:** Se debe realizar la creación y verificación del funcionamiento del código que construyo cada uno de los scripts. Se deben integrar los casos de prueba automatizados a la suite correspondiente debidamente analizada en el paso anterior.
Roles: Analista de Pruebas.
Soporte: Clases que contienen los nuevos casos de prueba a versionar dentro de las suites correspondientes.
Etapa Sprint: Sprint, Tarea de Desarrollo

Para mejorar la codificación de los casos de prueba se define el siguiente estándar (convenciones de código):

- Todos los script de casos de prueba deben contener en el inicio un comentario en el que se lista el autor, fecha, Descripción corta del caso de prueba.

```
/*  
 * @autor <a href= "email del usuario que automatizo el caso"> Nombre del usuario, </a>  
 * @since aa/mm/dd  
 * @throws Throwable  
 * This test case performance: <br/>  
 * Descripción en ingles corta del proceso que realiza el caso de prueba.  
*/
```

- La primera línea del caso de prueba debe contener el identificador y nombre del caso en la suite de la aplicación Test Link usada por la empresa.

@Test(description = "Identificador : Nombre del caso de prueba")

Después de esta línea se inicia la codificación del método que realizará la ejecución de la prueba automatizada.

- Cuando una expresión de código no sea visible en una línea, romperla después de una coma o de un operador.
- Identación del código para que sea fácil de leer.
- **Ejecución de suites:** Se realizará un proceso de pruebas mediante la ejecución de la suite automatizada con el fin de verificar su correcto funcionamiento con respecto al proceso automatizado. Este proceso se realiza dentro de un entorno estable y similar al productivo
Roles: Analista de Pruebas.
Soporte: Reporte de resultados de ejecución, actualización del documento generado en el análisis.
Etapas Sprint: Sprint, Tarea de Pruebas, Sprint Review
- **Integración de Ramas:** Actividad que comprende la integración de la rama que contiene los casos de prueba en el proyecto general de automatización. Se realiza el commit evitando conflictos entre las versiones de las clases.
Roles: Analista de Pruebas.
Soporte: Componentes integrados en el proyecto general de automatización.
Etapas Sprint: Trabajo Terminado, puesta en producción.

7.4 Descripción del sistema

Herramientas utilizadas

Selenium.

Open QA ofrece distintas soluciones para automatización de casos de prueba entre ellas:

- Selenium Core, herramienta que permite por medio de una aplicación de tipo web desarrollado en HTML y JavaScript permite realizar automatización de pruebas sobre diferentes navegadores.

- Selenium IDE, Extensión del navegador Mozilla Firefox que permite grabar flujos de navegación (Script de Prueba), para luego modificar y ejecutar las pruebas automatizadas en Selenium Core
- Selenium Remote Control, herramienta de automatización que brinda interfaces de programación (APIs) para diferentes lenguajes como Java, .NET, Perl, Python y Ruby.

Selenium Remote Control es la herramienta que se utiliza dentro de la empresa para la automatización de casos de prueba ya que permite la ejecución de la interfaz web sobre diferentes navegadores (Mozilla Firefox e Internet Explorer).

Eclipse

Eclipse es la solución de un entorno de desarrollo para la construcción, modificación y ejecución de los scripts, seleccionado por su facilidad al mejorar la experiencia en automatización de pruebas mediante plug-ins que logran la integración con otras herramientas como herramientas de automatización (Selenium), lenguajes de formato a documentos (HTML, XML, XHTML), funcionalidades para la creación y depuración de scripts (Firebug, XPath Checker y XPather), manejador de versiones (GitHub), integración continua (Jenkins).

Versión Eclipse Luna Service Release 2 (4,4.2)

Versión Java 1.7

Git Hub

Plataforma que realiza el control de versiones sobre proyectos por medio del alojamiento de un repositorio de código, permite clonar repositorios externos (folk) y realizar actualizaciones desde ramas propias a la rama por defecto (master), adicionalmente permite realizar el seguimiento sobre los commit (actualizaciones de código) realizadas sobre las ramas.

Versión 2.8.2.windows.1

Maven Versión 3.3.9

8. EVALUACIÓN ECONOMICA DEL PROYECTO

A partir de la matriz de riesgos realizada durante el proyecto se pueden evidenciar los siguientes riesgos asociados a cada una de las etapas del ciclo de vida del software.

Procesos: Automatización de casos de prueba				
Área Principal: Tecnología				
Equipo: Calidad				
Fase	Riesgo	Causa	Probabilidad de Ocurrencia	Impacto
Análisis	Cambios no contemplados	Modificaciones en proceso no contempladas en las funcionalidades en etapa de análisis de casos de prueba.	Media	Medio
	Definición errada de Complejidad	Asociación equivocada en la complejidad de los casos de prueba analizados.	Baja	Bajo
	Selección incorrecta	Selección inadecuada de casos de prueba debido al costo en tiempo del proceso de automatización.	Baja	Alto
Diseño	Diseños incompletos	Los diseños de casos de prueba pueden estar incompletos con respecto a las entradas y salidas (resultados esperados).	Media	Medio
Codificación	Diseños ambiguos	Casos de prueba manuales mal documentados que puedan generar retrasos o ambigüedades en la construcción del script.	Media	Medio
	Daños de infraestructura	Daños de infraestructura que no permitan avanzar a los analistas en el proceso de generación de scripts.	Baja	Alto

	Recursos asignados a Multitareas	Los analistas de pruebas pueden ser asignados temporalmente a otro proyecto de acuerdo a las necesidades de la empresa, disminuyendo la velocidad del equipo.	Alta	Alto
Pruebas	Documentación errada	Problemas de documentación en las fases de diseño y codificación, que pueden generar errores y por tanto re-procesos.	Media	Alto
	Casos ambiguos	Uso de información ambigua que no permite la ejecución de casos de prueba exitoso.	Media	Bajo
	Inestabilidad de ambientes	Uso de ambientes y datos inestables de Calidad que requieren tiempo de análisis profundo para identificar el problema.	Alta	Alto
Implementación	Conflictos de versionamientos	Conflictos en los versionamientos de las ramas del proyecto sobre la rama master o principal.	Alta	Medio
	Documentación inexistente	Errores en la ejecución de los casos de prueba por parametrizaciones sin documentar.	Baja	Bajo
	Versionamientos de Tecnología	Modificaciones en la tecnología que generar errores de compatibilidad.	Baja	Alto

Mantenimiento	Casos de prueba sin actualizar	Demoras en la actualización de los casos de prueba que sufren cambios en su funcionalidad.	Medio	Bajo
	Inconvenientes en Desarrollo	Inconvenientes al realizar la actualización de los casos de prueba automatizados por malas prácticas de desarrollo.	Medio	Alto

9. BENEFICIOS DE LA IMPLEMENTACIÓN

9.0 Operacionales

- *Ejecución rápida y efectiva.* Cada vez que dedicamos un tiempo en la automatización de un caso de prueba que se ejecuta repetidamente el resultado a nivel de rapidez y efectividad es evidente, debido a que no se requieren esfuerzos de adecuación de data y navegabilidad manual.
- *Reutilización.* Cuando realizamos pruebas de combinatorias sobre formularios con distintas entradas suele ser demasiado costoso, al implementar algoritmos de repetición con diferentes agrupaciones de datos en el proceso de automatización mejoramos la eficiencia del caso de prueba.
- *Datos Consistentes.* La automatización de casos de prueba requiere que los datos se adecuen para facilitar el proceso y se diseñan estrategias para evitar la modificación de estos. En las pruebas manuales el control es inexistente por el volumen de datos creado durante cada uno de los ciclos de pruebas.
- *Mayor Productividad.* Los procesos autónomos permiten minimizar las pérdidas de tiempos en los procesos productivos generadas por fallas humanas.

9.1 De Gestión

- *Reporte de Resultados.* El resultado de las ejecuciones de la suite de pruebas automatizadas se genera de acuerdo a la necesidad y en tiempos reales, al realizar las pruebas manualmente para adquirir el resultado del proceso debía registrarse en un sistema adicional la información y/o confiar en el reporte dado por el analista de pruebas.
- *Adaptable a Metodologías Agiles.* El marco de Scrum permite que la automatización de casos de prueba sea óptima para el desempeño de los equipos de desarrollo, por la constante necesidad de realizar regresiones en cada iteración, finalización de Sprint (En Pagos Online cada 2 semanas).

9.2 Estratégicos

- *Retorno sobre la Inversión (ROI).* El alto costo de infraestructura y tiempo que se invierte en los proyectos de automatización, se verán reflejados el largo plazo, debido al incremento de la velocidad de los equipos para generar nuevas funcionalidades ya que el tiempo en procesos de pruebas de regresión se redujo.

- *Confianza de Clientes.* La automatización de casos de prueba permite que los servicios ya productivos mantengan su calidad identificando errores fácilmente. Cuando un producto se encuentra estable en cuanto a sus funcionalidades y el cliente lo evidencia, puede deducir que sus procesos están alineados a sus necesidades incrementando su confianza en la empresa.
- *Competitividad.* La eficiencia que provoca el proceso de automatización, lleva a las empresas a un alto estándar de innovación y competitividad.

9.3 De Infraestructura

- *Aumento de Rendimiento.* La automatización de casos de prueba de regresión permite que su ejecución sea rápida y eficiente y por supuesto que permite su ejecución en horarios diferentes al proceso de desarrollo y pruebas funcionales manuales, esto permite la optimización de tiempos y estabilidad con respecto al rendimiento de los servidores.

9.4 De IT

- *Avances Tecnológicos.* Este proyecto permite la modernización de los esquemas de pruebas de software, dejando atrás patrones manuales para iniciar la era de las herramientas de automatización asegurando procesos de alto nivel tecnológico.

10. PRESUPUESTO

10.1 Costo de Infraestructura Física

Concepto	Cantidad	Costo Unitario	Costo Total
Equipo Personal	5	\$ 1.200.000	\$ 6.000.000
Servidor	1	\$ 13.000.000	\$ 13.000.000
<i>Costos de Infraestructura</i>			19.000.000

10.2 Costo de Recursos Humanos

Perfil	Salario mensual	Valor por hora	Salario mensual+ factor prestacional
Gerente de Proyecto	\$ 5.500.000	\$ 22.917	\$ 8.352.043
Arquitecto	\$ 5.000.000	\$ 20.833	\$ 7.592.767
Analista de Automatización	\$ 3.000.000	\$ 12.500	\$ 4.555.660
Analista de Pruebas	\$ 1.800.000	\$ 7.500	\$ 2.733.396
<i>Costos de Recursos Humanos</i>			\$ 23.233.866
<i>Costo Total (15 Meses estimados de Proyecto)</i>			\$ 348.507.990

10.3 Costo Total del Proyecto

Resumen de Costos

<i>Costos de Infraestructura</i>	\$ 19.000.000
<i>Costos de Personal</i>	\$ 348.507.990
<i>Costos de Operación (Resolución de Fallos)</i>	\$ 956.250
<i>Otros Gastos</i>	\$ 500.000
Costo Total del Proyecto	\$368.964.240

11. CRONOGRAMA

- Cronograma general.

	Nombre	Duración	Inicio	Terminado	Predecesores	Iniciales del recurso
1	PROYECTO AUTOMATIZACIÓN PRUEBAS	328 days	7/09/15 08:00 AM	7/12/16 05:00 PM		
2	Fase de Formulación	5 days	7/09/15 08:00 AM	11/09/15 05:00 PM		
3	Definición Meta y Objetivos	5 days	7/09/15 08:00 AM	11/09/15 05:00 PM		GP
4	Fase de Planeación	15 days	14/09/15 08:00 AM	2/10/15 05:00 PM	2	
5	Definición de Costos	5 days	14/09/15 08:00 AM	18/09/15 05:00 PM		GP
6	Definición del modelo de Riesgos	5 days	21/09/15 08:00 AM	25/09/15 05:00 PM	5	AA;LP;GP
7	Cronograma	3 days	28/09/15 08:00 AM	30/09/15 05:00 PM	6	GP;LP
8	Incertidumbre Fase de Planeación	2 days	1/10/15 08:00 AM	2/10/15 05:00 PM	7	AA;LP;GP
9	Fase de Análisis	10 days	5/10/15 08:00 AM	16/10/15 05:00 PM	4	
10	Definición de Requisitos Funcionales	5 days	5/10/15 08:00 AM	9/10/15 05:00 PM	4	LP
11	Definición de Requisitos NO Funcionales	3 days	12/10/15 08:00 AM	14/10/15 05:00 PM	10	GP;LP
12	Incertidumbre Fase de Análisis	2 days	15/10/15 08:00 AM	16/10/15 05:00 PM	11	GP;LP
13	Fase de Ingeniería	12 days	19/10/15 08:00 AM	3/11/15 05:00 PM	9	
14	Diseño de Arquitectura	10 days	19/10/15 08:00 AM	30/10/15 05:00 PM		A;LP
15	Incertidumbre Fase de Diseño	2 days	2/11/15 08:00 AM	3/11/15 05:00 PM	14	A;LP
16	Fase de Desarrollo y Pruebas	286 days	4/11/15 08:00 AM	7/12/16 05:00 PM	13	
17	Construcción del Modelo	30 days	4/11/15 08:00 AM	15/12/15 05:00 PM		A;AA
18	Acoplamiento (MiddleWare)	10 days	10/12/15 08:00 AM	23/12/15 05:00 PM		A;AA
19	Construcción Scripts Fase I S.	20 days	24/12/15 08:00 AM	20/01/16 05:00 PM	18	AA;AP
20	Incertidumbre Fase I	5 days	21/01/16 08:00 AM	27/01/16 05:00 PM	19	AA;AP
21	Construcción Scripts Fase II S.	20 days	28/01/16 08:00 AM	24/02/16 05:00 PM	20	AA;AP
22	Incertidumbre Fase II	5 days	25/02/16 08:00 AM	2/03/16 05:00 PM	21	AA;AP
23	Construcción Scripts Fase III S.	20 days	3/03/16 08:00 AM	30/03/16 05:00 PM	22	AA;AP
24	Incertidumbre Fase III	5 days	31/03/16 08:00 AM	6/04/16 05:00 PM	23	AA;AP
25	Construcción Scripts Fase IV S.	20 days	7/04/16 08:00 AM	4/05/16 05:00 PM	24	AA;AP
26	Incertidumbre Fase IV	5 days	5/05/16 08:00 AM	11/05/16 05:00 PM	25	AA;AP
27	Construcción Scripts Fase V S.	20 days	12/05/16 08:00 AM	8/06/16 05:00 PM	26	AA;AP
28	Incertidumbre Fase V	5 days	9/06/16 08:00 AM	15/06/16 05:00 PM	27	AA;AP
29	Construcción Scripts Fase VI S.	20 days	16/06/16 08:00 AM	13/07/16 05:00 PM	28	AA;AP
30	Incertidumbre Fase VI	5 days	14/07/16 08:00 AM	20/07/16 05:00 PM	29	AA;AP
31	Construcción Scripts Fase VII S.	20 days	21/07/16 08:00 AM	17/08/16 05:00 PM	30	AA;AP
32	Incertidumbre Fase VII	5 days	18/08/16 08:00 AM	24/08/16 05:00 PM	31	AA;AP
33	Construcción Scripts Fase VIII S.	20 days	25/08/16 08:00 AM	21/09/16 05:00 PM	32	AA;AP
34	Incertidumbre Fase VIII	5 days	22/09/16 08:00 AM	28/09/16 05:00 PM	33	AA;AP
35	Construcción Scripts Fase IX S.	20 days	29/09/16 08:00 AM	26/10/16 05:00 PM	34	AA;AP
36	Incertidumbre Fase IX	5 days	27/10/16 08:00 AM	2/11/16 05:00 PM	35	AA;AP
37	Construcción Scripts Fase X S.	20 days	3/11/16 08:00 AM	30/11/16 05:00 PM	36	AA;AP
38	Incertidumbre Fase X	5 days	1/12/16 08:00 AM	7/12/16 05:00 PM	37	AA;AP

Iniciales de los recursos (Perfiles)

	Lider de Pruebas	LP
	Analista de Automatización	AA
	Analista de Pruebas	AP
	Gerente de Proyecto	GP
	Arquitecto	A

El proceso de pasantía empresarial se realizó en el periodo comprendido entre 8 de febrero y el 15 de marzo, sobre la fase de Desarrollo y Pruebas con perfil de Analista de Pruebas.

Se entregaron 3 informes¹³ a la universidad donde se relacionó el avance de acuerdo al siguiente cuadro:

<i>Informe</i>	<i>Periodo</i>	<i>Casos Automatizados</i>
1	08 Febrero – 15 Marzo	10
2	15 Marzo – 15 Abril	15
3	18 Abril – 13 Mayo	12

¹³ Ver Anexo 1. Informes de Pasantía Organizacional

12. CONCLUSIONES

Se realizó el mantenimiento y depuración de los casos de prueba correspondientes al equipo de Payments Methods, sobre la suite de pruebas de regresión, eliminando todos los casos duplicados. Adicionalmente se avanzó en la automatización de 37 casos de prueba que se encontraban pendientes, esto mejoro los tiempos de regresión y la disminución de la deuda técnica.¹⁴

Durante el proceso de automatización de casos de prueba se evidenciaron mejoras que pudieron realizarse, por ejemplo, el estándar de manejo de excepciones, las cuales ahora describen en que momento el script genera un fallo para realizar un análisis y mantenimiento más certero, y la generación de datos en ambientes alternos para que la contingencia de la ejecución de la suite automatizada no sea demasiado costosa.

Los equipos de Testing adoptaron prácticas enriquecedoras en sus procesos y aislaron las dependencias con los analistas desarrolladores de aquellos casos de prueba que presenten fallos. La adaptabilidad de los equipos a las mejoras implementadas sufrió bastantes discusiones acerca de los tiempos que podría tomar incluirlas como obligatorias en el proceso de automatización, a pesar de los inconvenientes se logró argumentar los beneficios y establecerlas.

Se evidencio una mejora positiva en los procesos de pruebas en la empresa, y por tanto, se ha dado la posibilidad de incluir más puntos en la velocidad del equipo para automatizar y generar nuevas funcionalidades.

¹⁴ Anexo 3. Código Scripts Automatizados

13. RECOMENDACIONES

Se recomienda que los analistas de pruebas dedicados a la disminución de la deuda técnica dediquen su tiempo completamente al diseño de scripts ya que la ejecución de múltiples tareas impide especializarte, concentrarte y agilizar la tarea.

Se sugiere iniciar el proceso de automatización de suites de pruebas adicionales que toman una cantidad de tiempo importante en ejecutar de manera manual:

- Pruebas de Carga: Automatizar este tipo de pruebas genera mayor efectividad y probabilidad de fallos humanos al procesar grandes cantidades de datos y/o archivos en tiempos específicos. Un robot automatizado puede brindar mayor seguridad con respecto a los intervalos entre una y otra carga.
- Pruebas de Performance: La optimización de pruebas de performance requiere un análisis efectivo, de primera mano podríamos evaluar y automatizar casos que midan el nivel de latencia de la aplicación.

14. GLOSARIO

A continuación, encontrará las definiciones relevantes con respecto al mercado electrónico y la ingeniería de sistemas.

API: Application Programming Interface, servicios (procedimientos y funciones) que el Sistema Operativo ofrece al programador.

ECONOMÍA DIGITAL: o economía web, se refiere al aprovechamiento de las tecnologías de la información para realizar pedidos y pagos electrónicos de bienes y servicios.

ESCALABILIDAD: habilidad de un sistema para reaccionar ante el crecimiento potencial de trabajo.

FIABILIDAD: capacidad de que los productos de software funcionen bajo condiciones y un período fijo.

IDE: un entorno de desarrollo integrado que brinda herramientas para un proceso de desarrollo amigable.

INFRAESTRUCTURA: conjunto de servicios o elementos requeridos para el funcionamiento eficiente de una entidad.

IT: Information Technology, se refiere a las tecnologías de la información y la comunicación (TIC, TICs).

LATENCIA: el tiempo de retraso entre el momento en que se inicia una transacción y el momento en que se completa.

MERCADO: contexto en el que se realiza intercambio de productos o servicios, intervienen la oferta(vendedores) y la demanda(compradores).

NAVEGADORES: software utilizado para acceder a internet (Mozilla Firefox, Internet Explorer, Google Chrome, entre otros).

NEGOCIOS: relaciona procesos de producción, distribución y venta de bienes y servicios.

ONLINE: en línea, se refiere a conectividad eficiente.

PAGADOR: cliente, persona que adquiere un servicio o producto y realiza un pago en línea.

PAYU: empresa Colombiana que presta servicios de pagos en línea.

PLUGINS: o Complemento que añade una funcionalidad o característica a un software.

PORTABILIDAD: propiedad del software para ser ejecutado en diferentes sistemas operativos y/o plataformas.

RAMA: versión de una copia puntual de archivos con el estado actual, que puede ser modificada y/o reutilizada.

RECAUDADOR: vendedor, persona o entidad que recibe el pago en línea por la venta de un servicio o producto.

REQUEST: petición de solicitud de información con una estructura determinada.

RESPONSE: respuesta a una petición de objetos y/o información con una estructura determinada.

SCRIPT: se refiere a un archivo de procesamiento por Lotes que realiza una tarea específica.

SDK: Software Development Kit: Paquete de herramientas que permiten crear aplicaciones.

SPRINT: iteración de entre 1 y 4 semanas que atiende un subconjunto de historias de usuario contenidas en un Backlog.

SUITE DE PRUEBAS: agrupación de casos de prueba.

USABILIDAD: facilidad con la que un usuario puede utilizar cierta herramienta.

TESTER: analista que se dedica a realizar la comprobación de componentes de software.

TESTING: se refiere a las pruebas de software que buscan proporcionar un nivel de calidad a un producto.

TRAMAS: unidad de envío de datos (bits).

15. BIBLIOGRAFIA

ALONSO, Fernando. MARTINEZ Loic. y SEGURA Fco. Javier. Pruebas del Software. En: Introducción a la Ingeniería de Software. 1 ed. Madrid. 2005 p. 87-90

APICA. Automated Testing vs Manual Testing: Which Should You Use, and When?. [en línea]. <https://www.apicasystem.com/blog/automated-testing-vs-manual-testing/>. [citado en 07 de Noviembre de 2014]

AUTOMATIZACIÓN y CONTROL. Beneficios. [en línea]. http://autycontrol.com/?page_id=7.

BASE 36 Smart Solutions. Automated vs. Manual Testing: The Pros and Cons of Each. [en línea]. <http://www.base36.com/2013/03/automated-vs-manual-testing-the-pros-and-cons-of-each/>.

BERENJEL GOMEZ, José Luis. Pruebas de Software. En: Desarrollo de aplicaciones web en el entorno servidor. 1 ed. Madrid. Ediciones Nobel 2016. p. 34-46

DIMENSIONAL RESEARCH. Análisis de Riesgos. En: La automatización de procesos de TI en la vida real: los clientes de HP comparten sus experiencias. [en línea]. <http://www8.hp.com/h20195/v2/GetPDF.aspx%2F4AA4-4419ESE.pdf9>.

DIMES, Troy. Conceptos Básicos de Scrum. En: Conceptos Básicos de Scrum: Desarrollo de Software Ágil y Manejo de Proyectos Agile. 1 ed. Babelcube 2015. p. 10-33

DOS Ideas. Personas y Software. Todo Scrum en 1 sola página. [en línea]. <http://www.dosideas.com/noticias/metodologias/981-scrum-en-1-sola-pagina.html>. [citado en 9 de Mayo de 2013]

FORD, Sara. Automation Testing versus Manual Testing Guidelines. [en línea]. <https://blogs.msdn.microsoft.com/saraford/2005/02/07/automation-testing-versus-manual-testing-guidelines/>. [citado en 07 de Febrero de 2005]

GIT HUB. Documentation. [en línea]. <https://git-scm.com/>.

GRANADOS LA PAZ, Rafael Luis. Pruebas de Software. En: UF 1844: Desarrollo de aplicaciones web en el entorno servidor. 1 ed. Málaga. IC Editorial, 2014.

JAMES, Michael. Acerca de Scrum. [en línea]. <http://scrumreferencecard.com/reference-card-de-scrum/>.

MADERNA, Alejandro. Pruebas Manuales vs. Automatizadas: ventajas y desventajas. [en línea]. <http://www.kanoah.com/blog/pruebas-manuales-vs-automatizadas-ventajas-desventajas/>. [citado en 20 de Junio de 2016]

MANCISIDOR RUEDA, Ana López. La automatización de pruebas como una inversión. En: ¿Por qué invertir en la automatización de pruebas Software?. [en línea]. 2014. <http://www.ati.es/IMG/pdf/IBM.pdf>. [citado en 15 de Febrero de 2019]

MANDADO PÉREZ, Enrique. MARCOS ACEVEDO, Jorge. FERNÁNDEZ SILVA, Celso. y ARMESTO QUIROGA, José I. Confiabilidad de los Sistemas Electrónicos. En: Autómatas Programables y Sistemas de Automatización. 2 ed. Marcombo S.A. 2009. p. 934-938

NARVÁEZ ROSERO, María del Pilar. Análisis de Riesgos. En: Gestión de Riesgos en la Fase de Diseño para Proyectos de Construcción Utilizando la Guía PMBOK. [en línea]. 2014. <https://git-scm.com/>. [citado en 23 de Abril de 2014]

SOFTWARE Testing Genius. Pros and Cons of Automated Testing. [en línea]. <http://www.softwaretestinggenius.com/pros-and-cons-of-automated-testing>.

STEFANINI Informática & Tecnología. Conoce los Beneficios de las Pruebas de Software Automatizadas. [en línea]. <http://www.stefaninicolombia.com/conoce-los-beneficios-de-las-pruebas-de-software-automatizadas/>. [citado en 01 de Marzo de 2016]

TESTNBUG. Test Automation vs. Manual Testing in Software Development. [en línea]. <http://www.testnbug.com/2015/06/test-automation-vs-manual-testing-in-software-development/>. [citado en 29 de Junio de 2015]

TUYA, Javier. RAMOS ROMÁN, Isabel. y DOLADO COSÍN, Javier. Técnicas y Practicas en las Pruebas del Software. En: Técnicas Cuantitativas Para la Gestión en la Ingeniería del Software. 1 ed. La Coruña. NetBiblo, S.L. p. 45-67