



BECAS NEORIS - TESTING

TRABAJO PRÁCTICO FINAL:

PedidosYa

Índice

1. Rol de cada integrante
2. Resumen
3. Palabras claves del trabajo
4. Análisis del caso
5. Definición del problema
6. Desarrollo del caso
7. Método/s de solución abordado/s
8. Herramientas utilizadas
9. Resultados
10. Recomendaciones o Futuras extensiones al trabajo
11. Conclusiones
12. Bibliografía

1. Rol de cada integrante

Product Owner(PO): Carolina Boschini

Responsabilidades: es el responsable de la recepción de las solicitudes/*stories* del cliente, del manejo del *product backlog*, la gestión de prioridades y todo aspecto referente a la planificación del proyecto así como de la organización de meetings diarios y/o semanales para lograr colaborativamente el cumplimiento de las metas y *sprints* establecidos. Participa en discusiones colaborativas y Sprint Retrospectives para determinar sólo los detalles de alto nivel basados en su conocimiento sobre las necesidades y los requerimientos del cliente. Junto con el resto del equipo, se encarga de desglosar dichos planes para probar funcionalidades reducidas. En este equipo, el PO utiliza tableros Kanban para visualizar cómo avanza el trabajo en tiempo real, ser más flexible, apelar al trabajo colaborativo y aplicar cualquier cambio a los planes si es necesario. El PO analizará también la productividad del equipo, las historias recibidas y se ocupará de generar estrategias conjuntas para incrementar tanto la satisfacción del cliente, la productividad como la motivación del equipo. Analizará también los requerimientos de capacitación y planificación del equipo en el Sprint 0. Acepta o rechaza el producto desarrollado.

Scrum Master: Joaquín Ponte Vilela

Responsabilidades: Coach y facilitador de las reglas del SCRUM, resolución de cualquier problemática vinculada a la asignación/repartición de recursos. Analiza cualquier evento que provoca *stopeo* al cumplimiento de las metas establecidas.

QC Testers: Andrea Gabriella Arguellas, Natalia Tonelli y Jhoandra Astudillo(Seniors)

Responsabilidades: desarrollar, mantener y ejecutar los casos de prueba manuales. Se ocupan también del mantenimiento y realización de versiones de emergencia. Identificar, analizar e informar los resultados de las pruebas, así como de informar, rastrear y monitorear defectos, preparar la documentación de prueba y revisarla con el equipo de desarrollo, diseñar estrategias de prueba y los *sprints* junto con desarrollo.

SDET Testers: Virginia Torrazza (Senior) y Bruno Lategano (Senior).

Responsabilidades: automatizar los casos de prueba, escritura de scripts/herramientas reutilizables que ayuden al equipo a reducir esfuerzos repetitivos, investigar los problemas de los clientes remitidos al equipo de soporte técnico, probar errores y crear y administrar informes de dichos errores. Creación de marcos de automatización para diferentes plataformas (web, mobile, desktop) así como de diferentes escenarios de prueba y pruebas de aceptación. Realizar informes diarios y semanales sobre las pruebas realizadas. Analizar y desintegrar sistemas de software y coordinar con otros equipos para mejorar el diseño general. Gestionar la calidad final en el lanzamiento de producción. Buscarán una retroalimentación continua con los desarrolladores y probadores del equipo siguiendo un enfoque XP. Cada equipo estará formado por un SDET tester, un QC tester (Senior) y dos testers semi-senior.

2. Resumen

Nuestro cliente Pedidos Ya nos solicita realizar el Testeo de su App para dispositivos móviles en su versión actual (5.23.10.1) dado que se encuentra con un fallo de nivel crítico en producción el cual necesitan resolver con suma urgencia. Cabe destacar que mientras se realice el trabajo que nos solicita nuestro cliente, la aplicación solo seguirá funcionando, ofreciendo al usuario final, el método de pago “Efectivo”. Mientras tanto se le recomienda al cliente que a futuro deberá proponer u ofrecer al usuario final diferentes alternativas de pago, como por ejemplo código QR, links de pago o billetera virtual para que no le genere tantas pérdidas en caso de tener un fallo crítico en cualquier otro método de pago.

Nuestro equipo de Testing se compromete a resolver este Hotfix a la mayor brevedad posible, poniendo todo el esfuerzo necesario a disposición.

Para ello utilizará las herramientas adecuadas o que considere necesario para llegar a una rápida solución efectiva que se adecue a las necesidades del cliente. El equipo contemplará, si es necesario, llevar a cabo horas/esfuerzo extra para cumplir en tiempo y forma.

3. Palabras clave

TestLink: es una herramienta de Testing para crear y gestionar casos de prueba y organizarlos en casos de prueba.

HotFix: término que significa “parche caliente” supone solucionar un problema/error de un software en funcionamiento, es decir, una vez terminada su etapa de desarrollo.

Feature Flags: son valores booleanos que permiten mostrar o esconder una funcionalidad o sección en la interfaz de usuario.

Trello: es un software de administración de proyectos con interfaz web y con cliente iOS y Android para organizar y gestionar proyectos.

Automation: las pruebas de automation se realizan utilizando herramientas de software de pruebas automatizadas especiales, es decir, no manuales, para ejecutar un conjunto de casos de prueba.

4. Análisis del caso

Pedidos Ya nos solicita realizar el Testeo de su App, específicamente en “Pago con Tarjeta de Crédito”, para dispositivos móviles en su versión actual (5.23.10.1) dado que se encuentra con un fallo de nivel crítico en producción el cual necesitan resolver con suma urgencia.

Lo que ocurre al intentar abonar con el medio de “Tarjetas de Crédito” es que en la App se presenta el mensaje “No se pudo procesar el pago” debido a un error en el reconocimiento del serial de la tarjeta. Este suceso se presenta constantemente, generando que deban optar por otros métodos de pago, como el efectivo que no es lo que prefieren los usuarios.

5. Definición del problema

El problema planteado por el cliente, se refiere a un bloqueo en la modalidad de pago con tarjetas de crédito el cual le está trayendo serios inconvenientes a nivel negocio ya que es uno de los métodos más utilizados por los usuarios. Este inconveniente es replicado por los usuarios en las redes sociales lo cual también afecta a la imagen de nuestro cliente por lo que requiere inmediata resolución.

6. Desarrollo del caso

Para afrontar esta problemática, nuestro equipo de Testing decide utilizar la herramienta Testlink ya que todos los integrantes tienen amplia experiencia en la misma y eso reducirá el tiempo de ejecución. Respecto a las pruebas, decidimos realizar pruebas de sanidad para probar ciertas funcionalidades a profundidad. También se realizaron pruebas de integración y regresión para corroborar que los cambios efectuados no hayan afectado la funcionalidad del software.

Para trabajar internamente de manera organizada y eficiente, el equipo utilizará la herramienta Trello, un tablero virtual, ya que es una herramienta colaborativa donde todos pueden ver en tiempo real el estado de cada tarea asignada, y se puede llevar a cabo un story mapping. Además, Trello posee una integración con Jira y con el slack que son herramientas que el equipo viene utilizando en otros proyectos y facilitan la tarea.

En esta herramienta decidimos utilizar 4 columnas ya que son suficientes para resolver este problema. Dichas columnas son las siguientes: Backlog, To-Do, Testing y Done.

En cuanto a la comunicación en general tanto dentro del equipo como con nuestro cliente se utilizarán las herramientas Gmail, Meet y Whatsapp según sea la importancia, necesidad y urgencia de lo que se desea comunicar.

7. Método/s de solución abordado/s

El método de solución abordado es *Agile Testing*

8. Herramientas utilizadas

Herramienta para organizarnos como equipo se utilizaran: Trello.

Herramienta para gestionar las ejecuciones y casos de prueba utilizaremos: TestLink.

Herramientas de comunicación: Gmail, Whatsapp, Meet.

9. Resultados

Luego de haber definido la estrategia y metodología para afrontar un fallo crítico de estas características, el equipo de Testing logró resolver el problema satisfactoriamente en el tiempo de 2 (dos) días. Esto fue posible dado que es un equipo sólido, comprometido al 100% y con avanzados conocimientos en testing y sus herramientas.

10. Recomendaciones o Futuras extensiones de trabajo

Se le recomienda a nuestro cliente para el futuro, tener más de un proveedor de pagos con tarjetas para evitar que este fallo vuelva a repetirse e implementar un ambiente de homologación.

También se le recomienda y ofrece un servicio de Automatización de pruebas ya que con esto podrá evitar fallos a nivel productivo que afecten nuevamente su negocio. También se recomienda utilizar herramientas para pruebas funcionales y no funcionales de Web Api: SOAP UI ¹ y Cucumber².

¹SOAP UI permitirá al equipo de SDET agregar validaciones, configurar y/o automatizar pruebas así como probar la/s funcionalidad/es requerida/s utilizando base de datos, reportes excel, scripts de Groovy y mantener un seguimiento de los casos testeados durante los distintos sprints.

²La elección de Cucumber se basa en que se trata de una herramienta de software que admite el desarrollo basado en el comportamiento y la importancia de su analizador de lenguaje (Gherkin) que permite que los comportamientos de

11. Conclusión

Luego de analizar la totalidad del trabajo realizado, consideramos como equipo que podríamos haber acortado el tiempo de resolución aproximadamente en medio día, si hubiésemos utilizado una planilla de Excel para plasmar las pruebas. En cuanto al trabajo en equipo consideramos que nuestra comunicación y colaboración es muy buena, lo que nos permite resolver de manera eficiente cualquier situación que se nos presente.

12. Bibliografía

- https://www.clarin.com/sociedad/-pedi-cosa-traieron-problemas-delivery-rappi-pedidos_0_AoD8l5SrX.html
- <https://www.youtube.com/watch?v=LlvXauqz7Rg>
- <https://tuquejasuma.com/pedidos-ya/reclamos/no-funciona-el-pago-on-line>
- <https://www.pedidosya.com.ar/>
- Future Flags: <https://dev.to/marianocodes/que-son-los-feature-flags-kc7>

software esperados se especifiquen en un lenguaje lógico que los clientes puedan entender. De esta forma, apoyándonos en Cucumber como herramienta, en un futuro, el equipo podrá pasar fácilmente de las historias de usuario brindadas a pruebas de comportamiento automatizables.