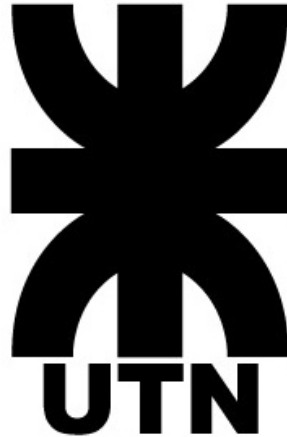


Universidad Tecnológica Nacional

Facultad Regional Córdoba

Ingeniería y Calidad de Software



Trabajo Práctico Evaluable 8:

Release & Sprint Planning

Año: 2024

Curso: 4K3

Integrantes del Grupo 5:

- Nayi Carlos | 90406
- Liendo Agustin | 90080
- Bergesio Genaro | 83464
- Cuello Tomas | 83296
- Floreano Micaela | 72276
- Galiano Tomas | 85824

Docentes a Cargo:

- Covaro, Laura
- Massano, Cecilia
- Gonzalez, Georgina

Fecha de entrega: 11/10/2024

Introducción

En esta evaluación, simulamos un escenario real de desarrollo ágil para profundizar nuestra comprensión de la planificación de sprints y entregas. Partiendo de un backlog de producto y un MVP, nuestro equipo creará un plan de entrega y actas de planificación de sprint. Como somos nuevos en Scrum y no tenemos experiencia con las tecnologías del proyecto, dedicamos un **Sprint 0** a establecer una base sólida, enfocándonos en el diseño de prototipos, la configuración del backlog y la investigación de herramientas clave como Google Maps API.

Consideraciones de Contexto necesarias

Contamos con un equipo de 6 personas, 5 con un seniority junior en cada área de expertise (front, back, testing, análisis, y bases de datos), y 1 siendo semisenior en bases de datos. La capacidad del equipo en horas ideales se estimó tomando en cuenta que es el primer proyecto del equipo trabajando juntos, además no posee experiencia profesional en cuanto a las tecnologías que solicita el usuario para la creación de la aplicación y es novato en el uso de metodología Scrum.

Con un enfoque en las funcionalidades principales, como el registro de usuarios y la búsqueda de taxis, hemos planificado sprints de dos semanas, ajustando nuestras expectativas debido a la falta de experiencia del equipo.

Horas laborales diarias por integrante: 3h

Días laborables: Lunes, Martes, Miércoles, Jueves y Viernes

Para el cálculo de la capacidad tendremos:

- WH = 18 hs
- DA = 10 días
- Capacidad = 180 hs lineales por sprint

En términos de las ceremonias, tendremos la review los días miércoles a última hora (20 a 21), las retrospectivas los Jueves a primera hora (18 a 19), y la planning (que ya cuenta para el sprint siguiente) los Jueves a última hora (20 a 21). Redefinimos estos tiempos debido a que nuestra disponibilidad de horas lineales es acotada, por lo que dedicar más tiempo nos resulta muy restrictivo. Trabajamos de 18 a 21 todos los días (de lunes a viernes), y tendremos la daily a las 18.30. En términos de velocidad, contamos con una velocidad promedio de 5 puntos por sprint, pero que fluctúa en épocas de examen.

Además, históricamente, el equipo ha sido optimista en sus estimaciones, lo que ha llevado a una brecha entre lo estimado y lo entregado.

Como consideración extra, entendemos que al finalizar el día uno está cansado, y no se utilizarán las horas disponibles al 100%, por lo que estimamos un uso de un 65%, lo que nos deja una **capacidad real** de $180\text{hs} \times 0.65 = 117\text{hs por sprint}$.

Se establecen un conjunto de actividades que debemos realizar antes que se comience el proyecto y los sprints. A este lo denominaremos sprint 0 y son tareas obligatorias para que nuestro proyecto prospere.

Tareas Sprint 0:

- Diseño de prototipos
- Establecer el Product Backlog
- Armar esquema de desarrollo
- Crear y configurar repositorio
- Definir lenguajes de programación y hardware a utilizar
- Establecer reglas de buenas prácticas y estilos
- Diseñar un prototipo de estructura de datos.
- Generar ambiente de testing, desarrollo y producción.
- Crear los servidores para testing, desarrollo y producción.
- (SPIKE) Investigación API Google Maps
 - Investigar cómo se integra esta tecnología en el proyecto.
 - Crear prueba de concepto para poder integrar API de Maps.
- (SPIKE) Investigar y realizar prueba de concepto para utilizar geoposicionamiento.
- (SPIKE) Investigar integración con la API de Facebook

Se entiende a las spike en sprint 0 como una investigación dentro de un timebox para ver la factibilidad de incorporar ciertas funcionalidades en el proyecto y reducir el riesgo antes de comprometerlos.

Plan de Release

Cuánto Tiempo

De acuerdo con la velocidad y capacidad del equipo, el primer release con las US que forman al MVP se estima completado en 3 sprints de 2 semanas cada uno, es decir 10 semanas (2 meses y medio).

Cuántos Sprints

Se desarrollará el MVP en el transcurso de 5 sprints de 2 semanas cada uno.

Qué US son parte del Release

Son parte de este release, las siguientes US (en frase verbal):

- Buscar taxis cercanos
- Pedir taxi Loguear taxista
- Notificar a taxista solicitud de taxi
- Ver ubicación del pasajero
- Ocupar taxi
- Liberar taxi
- Loguear taxista

Para determinar qué se trabaja en cada sprint, debemos priorizar las US..

Priorización de las US

1. Loguear taxista
2. Pedir taxi
3. Buscar taxis cercanos
4. Notificar a taxista solicitud de taxi
5. Ver ubicación del pasajero
6. Ocupar taxi
7. Liberar taxi

Nuestro MVP se centrará en permitir que los taxistas se registren y comiencen a recibir solicitudes de servicio, al tiempo que los usuarios puedan solicitar un taxi desde cualquier ubicación.

En total, estimamos la cantidad de 21 SP.

Sprint	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Item	- Loguear taxista (1) - Buscar Taxistas cercanos (3)	- Pedir taxi (5)	- Notificar a taxista solicitud de taxi (3)	- Ver ubicación del pasajero (5)	- Ocupar Taxi (2) - Liberar Taxi (2)
Cant. SP	4	5	3	5	4

Sprint Planning 1

Condiciones de Contexto

Para este sprint no tendremos exámenes, por lo que trabajaremos en normalidad.

Minuta Sprint Planning	
Sprint N°: 1	
Duración del Sprint: 10 días	
Objetivo del Sprint:	
Ofrecer al cliente las funcionalidades relacionadas con las US de Loguear Taxista y Buscar Taxis cercanos. Las mismas deberán ser diseñadas, testeadas, revisadas y aceptadas para autenticar a los conductores y visualizar sus ubicaciones en tiempo real.	
Capacidad del equipo en horas ideales: 180 hs ideales, 117 hs reales	
Equipo Scrum:	
<ul style="list-style-type: none">- Nayi Carlos- Liendo Agustin- Bergesio Genaro- Cuello Tomas- Floreano Micaela- Galiano Tomas	
Definición de Hecho (DONE)	Sprint Backlog
<ul style="list-style-type: none">- Diseño revisado y aprobado por el Product Owner- Código completo:<ul style="list-style-type: none">- Código versionado en el repositorio en la rama correspondiente- Código siguiente el manual de buenas prácticas- Refactorización de código- Revisión de código- Probado<ul style="list-style-type: none">- Pruebas unitarias hechas:<ul style="list-style-type: none">- Camino feliz- Camino alternativo- Cambios de estado- Datos guardados- Pruebas de integración	#Se adjunta más abajo una foto con mayor resolución

Sprint Backlog (con mas resolucio

SPRINT 1

Buscar taxis cercanos 2 SP	Diseñar el prototipo que busque taxis cercanos. 12hs	Realizar pruebas a los prototipos y documentar los resultados. 4hs	Diseñar pantalla de búsqueda de taxis 9hs	Diseñar lógica de negocio para la pantalla de búsqueda. 8hs	66 hs
	Crear íconos de taxi en el mapa - Diseño. 2hs	Generar filtros dinámicos para búsqueda en tiempo real. 8hs	Crear casos de prueba 3hs	Realizar pruebas unitarias. 6hs	
	Realizar pruebas de integración con la aplicación 6hs	Realizar prueba de regresión 7hs	Realizar prueba de aceptación con el cliente 1hs		
Loguear taxista 3 SP	Diseñar el prototipo UI para loguear taxista 4hs	Establecer mecanismo de manejo de sesión 6hs	Utilizar solución de logueo con facebook 8hs	Diseño de interfaz para loguear taxista. 2hs	40 hs
	Lógica de negocio para conseguir taxista 2hs	Crear casos de prueba 2hs	Realizar pruebas unitarias 2hs	Realizar pruebas de integración 4hs	
	Realizar prueba de Regresión 3hs	Realizar prueba de Aceptación 1hs			
					+
5 SP					106 hs

Bibliografía

Rubin, K. S. (2012). Essential Scrum: A Practical Guide to the Most Popular Agile Proces. Schwaber, K., & Sutherland, J. (Noviembre de 2020). La Guía de Scrum.

Obtenido de:

<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-GuideSpanish-Latin-South-American.pdf>