NICOLAS ALEJANDRO MARIN VILLEGAS

Mejoras al ciclo de desarrollo de software mortonshoes

[Subtítulo del documento]

**Índice**

1. Introducción
2. Análisis de la Situación Actual  
   2.1 Fortalezas  
   2.2 Debilidades
3. Propuestas de Mejora  
   3.1 Basadas en CMMI  
   3.1.1 Definir y Estandarizar Procesos  
   3.1.2 Automatización y Monitoreo Continuo  
   3.1.3 Mejorar la Gestión de Riesgos  
   3.2 Basadas en TMMI  
   3.2.1 Planificación de Pruebas Integrada  
   3.2.2 Optimización de Pruebas de Regresión  
   3.2.3 Pruebas Basadas en Riesgos  
   3.2.4 Estrategias de Validación y Verificación
4. Beneficios de las Propuestas
5. Conclusión

El propósito de este informe es identificar áreas de mejora en el ciclo de desarrollo de software del proyecto MortonShoes en función de los datos recopilados a través de pruebas funcionales, pruebas unitarias, pruebas de carga y estrés. Además, se presentan sugerencias prácticas para optimizar la calidad y eficiencia del software durante el proceso de desarrollo basadas en los estándares del modelo de madurez CMMI y TMMI.

**2. Análisis de la situación actual**

**2.1 Fortalezas**

• Flujo funcional validado: las pruebas funcionales y modulares muestran que la mayoría de las funciones cumplen con los requisitos establecidos.

• Gestión de riesgos inicial: una matriz de riesgos identificó y priorizó problemas críticos como la seguridad y el rendimiento.

• Pruebas exhaustivas: se realizaron pruebas de carga y estrés, cuyos resultados demostraron una alta utilización de recursos en condiciones normales.

**2.2- Debilidades**

1. Gestión de errores:

Errores críticos como imposibilidad de actualizar archivos de configuración en tiempo real y problemas para acceder a las funciones de gestión después de cerrar sesión.

2. Automatización limitada:

ninguna Las pruebas unitarias y las pruebas de integración se realizan principalmente de forma manual. Esto retrasa el ciclo de prueba y permite errores humanos.

3. Gestión de calidad:

No existe un proceso documentado para garantizar la mejora continua en el desarrollo y las pruebas.

4. Optimización del rendimiento:

Los picos de CPU durante operaciones críticas, como iniciar sesión o comprar, indican la necesidad de optimización.

**Sugerencias de mejora**

**Basado en CMMI**

El modelo CMMI promueve la mejora de procesos en cinco niveles. MortonShoes está en el nivel 2 o 3 (gestión básica y definida). Se recomiendan las siguientes acciones para avanzar:

**1. Definir y estandarizar procesos (Nivel 3)**

- Implementar una estrategia de desarrollo estándar, que incluya revisiones periódicas de código y evaluaciones de defectos.

- Documente procesos repetibles de manejo de errores para garantizar la trazabilidad desde el descubrimiento hasta la resolución.

- Cree pruebas funcionales, plantillas integradas, manifiestos y regresiones.

**2.Automatización y monitoreo continuo (Nivel 4)**

- Automatice las pruebas unitarias utilizando herramientas como Jasmine/Karma integradas en canalizaciones de CI/CD.

- Configure la supervisión automatizada del rendimiento en entornos de desarrollo y producción.

- SonarQube integrado para funcionamiento continuo. Identificar vulnerabilidades y errores de calidad del código.

**3. Mejorar la gestión de riesgos (Nivel 5).**

Cree un sistema proactivo que utilice análisis predictivos basados ​​en datos históricos para identificar y mitigar los riesgos emergentes.

Realizar una revisión de riesgos para evaluar el impacto y la eficacia de las medidas adoptadas.

**Basado en TMMI.**

El modelo TMMI se centra en pruebas de madurez para garantizar su exhaustividad y eficacia.

**1. Plan de Pruebas de Integración (Nivel 2).**

- Desarrolle un plan de prueba que esté alineado con los objetivos comerciales y priorice las historias clave de los usuarios finales.

- Asigne un presupuesto de prueba específico y evalúe su idoneidad en cada iteración.

**2.Optimice las pruebas de regresión (Nivel 3)**

- Cree un conjunto de pruebas de regresión automatizadas para validar rápidamente los cambios y reducir el riesgo.

- Asegúrese de que las pruebas de integración incluyan escenarios del mundo real y sean coherentes con los módulos clave de la aplicación (catálogo, carrito de compras y pagos).

**3. Pruebas basadas en riesgos (Nivel 4)**

- Amplíe la matriz de riesgos para incluir métricas como el costo asociado, el impacto en el usuario y la complejidad técnica.

- Priorice las pruebas en áreas de alto riesgo, como la seguridad de los datos y el rendimiento de la carga.

**4. Estrategia de verificación y validación (Nivel 5)**

- Emplee métodos de verificación formales, como la verificación cruzada, para garantizar que la funcionalidad crítica se implemente correctamente.

- Implementar pruebas exploratorias en una etapa avanzada para identificar defectos no cubiertos por las pruebas tradicionales.

**Beneficios de la propuesta**

* Mejora de la eficiencia y la calidad: la estandarización y la automatización reducirán el tiempo de desarrollo y mejorarán la calidad del software.
* Reducir los defectos: La integración de herramientas de prueba y rastreo reducir Numero de errores de fabricación.
* Satisfacción del cliente: un sistema más estable y una interfaz de usuario simplificada aumentarán la confianza del cliente.

**Conclusión**

• MortonShoes está mejorando sus procesos de desarrollo y prueba. La implementación de las mejoras recomendadas basadas en las recomendaciones de CMMI y TMMI permitirá que el equipo avance a un mayor nivel de madurez. Se recomienda que haga de la automatización de pruebas y la estandarización de procesos su primera prioridad para lograr un rendimiento óptimo.