

UNTREF

UNIVERSIDAD NACIONAL
DE TRES DE FEBRERO

Trabajo práctico 2

Patrones de usabilidad

Docentes:

Dr. Díaz Pace, Andrés

Dr. Fontdevilla, Diego

Alumnos:

Campetella, Ramón

Cao, Ezequiel

Cozzi, Gonzalo

Ricci, Santiago

Curso:

Diseño y Evolución de Arquitecturas de Software

Índice

Índice.....	2
Consigna.....	3
Introducción.....	3
Prueba de concepto.....	3
Análisis cuantitativo.....	3
Análisis Cualitativo de Usabilidad.....	5
Ventajas y desventajas.....	5
Trayectorias de Evolución.....	6
a. Mejora de Algoritmos de Predicción.....	6
b. Frecuencia de actualización de los datos ofrecidos al usuario.....	6
Mecanismos para Garantizar el Sostenimiento de la Usabilidad.....	6
a. Pruebas regulares del algoritmo de predicción.....	6
b. Análisis de Feedback del Usuario.....	6
c. Actualizaciones Iterativas Basadas en Datos.....	6

Consigna

- El objetivo es que la prueba de concepto efectivamente se ejecute y que pueda realizarse un análisis de los atributos de calidad de modo cualitativo y cuantitativo.
- El análisis cuantitativo debe aportar métricas y condiciones de medición para caracterizar la solución.
- El análisis cualitativo debe explicitar ventajas y desventajas, trayectorias de evolución y mecanismos para garantizar el sostenimiento del atributo.
- La solución deberá estar documentada mediante un ADR, orientado a una audiencia pública, y no a los participantes de la materia.
- La documentación debe incluir las instrucciones para ejecutar, incluyendo instalación de software de base, etc.
- La entrega debe tener la forma de un repositorio de fuentes (github o similar) con la documentación en el README.
- Pueden elegir la tecnología de implementación que prefieran.
- Se valorará positivamente (puntos adicionales) si la entrega incluye pruebas automatizadas que ilustren los escenarios de aceptación.

Introducción

El presente trabajo tiene por finalidad presentar una implementación del patrón “Predicción de duración de tarea” del reporte técnico “Achieving Usability Through Software Architecture” de Len Bass, Bonnie E. John y Jesse Kates del Instituto de Ingeniería de Software de la Universidad de Carnegie Mellon. Para dicha implementación, se define una métrica que permite evaluarla desde la usabilidad, se presenta un análisis cualitativo y finalmente trayectorias de evolución y mecanismos para garantizar el sostenimiento del atributo

Prueba de concepto

En [este repositorio](#) puede encontrarse una implementación de una prueba de concepto.

Análisis cuantitativo

Dado que este patrón permite al usuario realizar otra tarea mientras que el sistema completa un proceso cuya ejecución demora un tiempo considerable, si el sistema no provee predicciones precisas, el usuario tomará decisiones con información incorrecta lo que puede llevar a que sean erróneas. En consecuencia, se considera que la solución debe ser evaluada desde este punto de vista. Para ello, proponemos como métrica el promedio del error absoluto medio para las predicciones de cada tarea:

$$E = \frac{1}{T} \sum_{t=0}^T E_t$$
$$E_t = \frac{1}{n} \sum_{i=0}^n |p_i - v_i|$$

Donde:

- n : es la cantidad de ejecuciones de una tarea determinada;
- p_t : es la duración predecida para la tarea t para la ejecución i ;
- v_i : es la duración real de la tarea t para la ejecución i ;
- T : es la cantidad total de tareas vistas hasta el momento;
- E_t : es el error absoluto medio para las predicciones de la tarea t ;

Un punto a tener en cuenta es que v_i puede ser afectado por:

- la carga del sistema a lo largo de la ejecución de una tarea t ;
- interacción con recursos externos al host donde se ejecuta, como puede ser un enlace de red;
- cuestiones propias de la tarea t como el tamaño de la entrada.

En consecuencia, se proponen las siguientes condiciones de medición:

- el conjunto de tareas de prueba debe ser siempre el mismo;
- no debe depender de recursos externos al host;
- la entrada de la tarea y el orden de ingreso de la misma debe ser siempre igual;
- el orden de ejecución de la tareas también debe ser el mismo.

Así, el objetivo es minimizar E .

Análisis Cualitativo de Usabilidad

Ventajas y desventajas

Ventajas	Desventajas
Predicción Basada en Datos Históricos Utiliza datos históricos para proporcionar estimaciones de duración, resultando en predicciones más precisas y relevantes. Impacto en Usabilidad: Genera confianza en el usuario, ya que puede basarse en métricas pasadas y tener expectativas realistas sobre la duración de las tareas.	Dependencia de Datos Históricos Si la aplicación carece de datos históricos para una tarea en particular, no podemos estimar su duración, y si los datos son escasos, la duración estimada tendrá una desviación elevada respecto de la duración real. Impacto en Usabilidad: Puede generar frustración en los usuarios que dependen de la precisión de las estimaciones para planificar su tiempo.
Feedback en Tiempo Real Muestra actualizaciones en tiempo real sobre el progreso de la tarea y las estimaciones de duración, permitiendo a los usuarios ajustar sus planes. Impacto en Usabilidad: Permite a los usuarios ajustar sus planes en función de la información actualizada, mejorando la percepción del control y la eficiencia.	Variabilidad en la Duración de las Tareas Las tareas pueden variar en complejidad y duración, lo que puede hacer que las predicciones sean menos fiables en ciertos casos. Impacto en Usabilidad: Puede llevar a expectativas no cumplidas, disminuyendo la satisfacción del usuario y la confianza en la aplicación.

Trayectorias de Evolución

a. Mejora de Algoritmos de Predicción

- **Estrategia:** Invertir en el desarrollo de algoritmos más sofisticados que no solo utilicen datos históricos, sino que también consideren factores contextuales (como el tipo de tarea, el perfil del usuario, condiciones de ejecución, etc.).
- **Resultado esperado:** Predicciones más precisas y adaptadas a las necesidades del usuario.

b. Frecuencia de actualización de los datos ofrecidos al usuario

- **Estrategia:** Se considera la posibilidad de evaluar la frecuencia de actualización de los datos que el usuario recibe, para que dichos datos estén siempre lo más actualizados que sean posibles. Esto implica un *trade off* con la *performance* de ejecución de la aplicación.
- **Resultado Esperado:** Mejora en la precisión de los datos que el usuario recibe, evitando que éste tenga información muy desactualizada.

Mecanismos para Garantizar el Sostenimiento de la Usabilidad

a. Pruebas regulares del algoritmo de predicción

- **Descripción:** Realizar pruebas periódicas basadas en escenarios controlados que permitan identificar desvíos en los resultados esperados de las predicciones. Estas pruebas podrían integrarse en el servidor de integración continua, a fin de garantizar la sostenibilidad del atributo.
- **Beneficio:** Permite recoger feedback directo y frecuente, ajustando la aplicación según la precisión esperada del algoritmo.

b. Análisis de Feedback del Usuario

- **Descripción:** Implementar mecanismos para recopilar y analizar el feedback de los usuarios (encuestas, valoraciones, etc.).
- **Beneficio:** Proporciona información valiosa sobre cómo los usuarios perciben la usabilidad y qué áreas requieren atención.

c. Actualizaciones Iterativas Basadas en Datos

- **Descripción:** Telemetría de la ejecución de la solución en entornos reales.
- **Beneficio:** Garantiza que la aplicación evoluciona para satisfacer las expectativas cambiantes de los usuarios y mantener su relevancia.