

Resumen extendido.

Hace tiempo, se empezó a pensar que el problema del desarrollo de software es hacer la cosa correcta para los usuarios, que agregue valor a su negocio, y hacerla correctamente, dentro de los plazos y presupuestos; aceptando el natural cambio en los requerimientos y respondiendo positivamente ante estos cambios.

En este marco, surge el agilismo para reducir los problemas clásicos del desarrollo de programas, a la par de dar más valor a las personas que componen el equipo de desarrollo.

Entre estos métodos ágiles, existen algunas prácticas en la que las pruebas pasan a ser una herramienta de diseño del código y, por tanto, se escriben antes que el mismo. De entre estas se destaca TDD, utilizando pruebas unitarias fue la precursora y BDD que propone poner el foco en generar valor para el usuario final utilizando las pruebas de aceptación y redefiniendo el dialogo con los usuarios en la búsqueda de un lenguaje ubicuo (común a todos) para lograr desde el primer momento hacer “la cosa correcta”, es decir un software que agregue valor al usuario final. Con el soporte de las pruebas, las prácticas asociadas y las herramientas estas prácticas también logran “hacer la cosa correctamente”.

El objetivo general de este trabajo es profundizar el estudio sobre las pruebas del software utilizando el “desarrollo dirigido por comportamiento” o BDD (*Behaviour-Driven Development*). Para ello se realizará una revisión bibliográfica sobre este tema específico y se aplicarán los conceptos y las técnicas en un desarrollo de software concreto para llevarlos a la práctica y detectar sus ventajas. Así construir un software multiplataforma para la organización y gestión de premios y certámenes.

La metodología seguida incluyo cuatro etapas: investigación preliminar, concepción, desarrollo de la aplicación y documentación de la experiencia.

El proceso de software para el desarrollo de la aplicación fue iterativo e incremental, adaptando prácticas propuestas por la programación extrema y las propias del desarrollo dirigido por comportamiento.

Las metodologías ágiles se soportan en herramientas para poder mantener la flexibilidad necesaria ante el cambio y disminuir la deuda técnica. Las principales herramientas utilizadas son: PivotalTracker, Cucumber, RSpec, Git, GitHub, Travis-CI, Coveralls.io, Heroku.

Cabe destacar, que se utilizaron en su mayoría herramientas libres o de código abierto y en todo caso de uso gratuito.

Completadas las etapas consignadas en la metodología se lograron los siguientes resultados y artefactos de software: se seleccionó Ruby on Rails y la prácticas ágiles de

BDD y TDD junto con herramientas asociadas compatibles con Ruby. Se obtuvo la especificación de requerimientos de software y las historias de usuario. Luego de la etapa de desarrollo se obtuvo un prototipo funcional para premios (aplicación web); el repositorio público del código; las pruebas automatizadas, la medición automatizada de la cobertura de pruebas; servidor de integración configurado y automatizado; versión de demostración de la aplicación publicada en línea, con datos de prueba. Finalmente el informe del TFA.

Se realizó la revisión bibliográfica, encontrando escasa información en español y abundante en inglés sobre la temática abordada, los conceptos y las prácticas asociadas.

Se aplicaron satisfactoriamente los conceptos teóricos asociados a BDD y TDD a una aplicación específica. Construyendo una aplicación web, multiplataforma, para la organización de premios.

Se pudo apreciar una mejora en el proceso de desarrollo al contar con una “documentación viva” que refleja fielmente y de forma actualizada lo que hace la aplicación; también al contar con la retroalimentación, casi inmediata, de las pruebas automatizadas para realizar cambios y solucionar problemas generando un código modificable más fácilmente.

Este trabajo pretendió en un primer momento aplicar TDD a un caso real de una aplicación web y terminó encontrando en BDD una herramienta ideal para la comunicación con los clientes, que permitió incluso aplicar la práctica de integración continua.

Se consideran varias líneas para trabajo futuro: como profundizar sobre las prácticas ágiles, analizar el impacto de su uso en las personas, entre otras. Se propone también continuar el desarrollo de la aplicación.