

Metodología de la programación II

Practica I

Emiliano Salvatori

Agosto 2019

1. Practica 1

Metodologías Ágiles

1.0.1. ¿Qué es lo que se conoce como "Desarrollo ágil de software" ?

Se conoce a "Desarrollo ágil de software" a un tipo particular de desarrollo de software, el cual se caracteriza por ser lo menos burocrática y compleja posible; tratando de ser simple, iterativa, e incremental. Con ello, se trata de generar ahorros significativos en tiempo y costes, comparado con los métodos de programación convencionales. Asimismo el trabajo es realizado mediante equipos autoorganizados y multidisciplinarios, basados en una comunicación eficiente. La mayoría de los métodos ágiles dividen las tareas en pequeños incrementos con una planificación mínima y no involucran directamente a la planificación a largo plazo.

^{1 2}

1.0.2. ¿Qué es el Evolutionary Project Management (EVO)? ¿Cuándo y quién lo desarrolló? ¿Cuáles son sus principios?

Se denomina gestión de proyectos evolutiva al conjunto de técnicas y prácticas empleadas para conducir la ejecución progresiva de un proyecto: de forma que genere un mínimo producto viable en el menor tiempo posible, y a partir de esa primera entrega, lo mantenga en mejora e incremento continuo. Fue desarrollado por Tom Glib en el en la década de 1970. ³

1.0.3. Investigue sobre la Alianza Ágil en su sitio web oficial www.agilealliance.org. ¿Qué es? ¿Cuáles son sus objetivos y actividades?

La Alianza Ágil es una organización sin fines de lucro comprometida a apoyar a las personas que exploran y aplican valores, principios y prácticas ágiles para hacer que la creación de soluciones de software sea más efectiva, humana y sostenible. La Alianza también genera actividades, las cuales incluyen:

- Conferencias que unen a la comunidad ágil cara a cara
- Un sitio web con información sobre la metodología Ágil y su Comunidad
- Membresía que brinda acceso a recursos creados por los miembros de la comunidad.
- Iniciativas que abordan áreas específicas de interés en la comunidad ágil y brindan apoyo a grupos de la comunidad local

1.0.4. Según el Manifiesto Ágil, ¿Cuál es la mejor forma de comunicación en un proyecto de desarrollo de software?

Según el Manifiesto Ágil la mejor forma de comunicación en un proyecto de software es el cara a cara. Es por ello que un equipo ágil habrá un representante del cliente, la cual se compromete a estar disponible para atender las consultas de los desarrolladores.

¹<https://es.ryte.com/wiki/Desarrollo\%C3%81gil\de\Software>

²<https://es.wikipedia.org/wiki/Desarrollo\%C3%A1gil\de\software>

³<https://www.scrummanager.net/bok/index.php?title=Gesti\%C3%B3n\evolutiva>

1.0.5. ¿Qué es el "timeboxing"?

Timeboxing se relaciona con determinar unidades de tiempo máximas para una actividad en concreto. Esa unidad de tiempo se denomina "time box."⁴ cuadro temporal. El objetivo del timeboxing es definir y limitar la cantidad de tiempo dedicado a una actividad. La metodología Scrum usa timeboxing para todos sus eventos y como una herramienta para definir concretamente tareas abiertas o ambiguas. Se denomina "timeboxes" a los períodos cortos de las iteraciones que por lo general duran de una a cuatro semanas.⁴

1.0.6. ¿Cuál es la clave de un método de desarrollo adaptativo? ¿Cuál es su opuesto y en qué se diferencian?

El desarrollo adaptativo de software (DAS) lo propuso Jim Highsmith 1998 como una técnica para construir software y sistemas complejos. Los apoyos filosóficos del DAS se enfocan en la colaboración humana y la organización propia del equipo. ASD resalta que las aproximaciones secuenciales en cascada solo funcionan en entornos bien conocidos. Pero como los cambios ocurren frecuentemente en el desarrollo software, es esencial usar un método tolerante a cambios. Los métodos en cascada son más rígidos y menos flexibles a los cambios, y si se adaptan lo hacen de una manera lenta y moderada. Los procesos en estos casos son controlados por políticas y normas estrictas; y casi siempre se adaptan al seguimiento de un estricto plan inicial de desarrollo.⁵

1.0.7. Intestige sobre el modelo y documentación JBGE ("Just Barely Good Enough")

El modelo JBGE apunta dentro de un proyecto de Software a generar documentación eficiente. Es decir, en contrapartida al modelo tradicional cuya estrategia es generar documentación para reducir riesgos, este método considera escribir documentación sólo cuando esa sea la mejor manera de lograr objetivos relevantes, ya que a menudo hay mejores maneras de lograr estos mismos objetivos que escribir documentación estática. Es por ello que se tienen determinados lineamientos a seguir:

- Se prefieren especificaciones ejecutables, entregables, por sobre una documentación estática.
- Se prefiere documentar conceptos estables y no meras especulaciones.
- Mucha documentación suele provocar desperdicio.
- Los desarrolladores rara vez confían en la documentación detallada porque generalmente no coincide con el código.
- Muy poca documentación también puede causar problemas para el mantenimiento, la comunicación y el aprendizaje.

6 7

1.0.8. Detalle algunas críticas negativas a las metodologías ágiles

Algunas cosas negativas que tienen las metodologías ágiles (Scrum y Kanban) son:

- No funcionan en equipos que tienen clientes con otra estructura de trabajo más tradicional (en el caso de la banca por ejemplo).
- La gran mayoría de los "expertos" en metodologías ágiles no tienen conocimientos en áreas de producción de software, léase codificación, testing o área funcional.
- No es sostenible a largo plazo en equipos que tienen una alta rotación de empleados, por lo que tiene a difuminarse el rol de cada uno de los integrantes. Al tratar de que cada una de las personas sea multidisciplinar lo que termina sucediendo es que se pierde tiempo en capacitar a personas que naturalmente están interesadas en realizar otras cosas y no conocer TODO el proceso de producción.
- Con la prontitud con la que se suelen tratar los entregables, terminan habiendo malas prácticas de codificación por lo que se pierde tiempo en realizar una revisión de código exhaustiva. Y termina redundando en una mala calidad de entregables y un pobre diseño de solución.

8

⁴<https://www.scruminc.com/what-is-timeboxing/>

⁵<https://blog.conectart.com/metodologias-agiles/>

⁶<http://tdan.com/best-practices-for-agile-documentation/18936>

⁷<https://linchpinseo.com/the-agile-method/>

⁸<https://simpleprogrammer.com/agile-is-dead-code-review/>

1.0.9. Encuanto a las pruebas ¿cuál es la diferencia entre los métodos ágiles y el de cascada?

Una de las diferencias entre Ágil y Cascada es que las pruebas se llevan a cabo en diferentes etapas del ciclo de vida de desarrollo. En Cascada siempre hay una fase de prueba por separado al final de la fase de implementación. En cambio, en las metodologías Ágiles las pruebas se realizan generalmente de manera concurrente con la codificación.

1.0.10. Para los creadores del Manifiesto Ágil, ¿Cuál es el principal factor de medición de avance de un proyecto? ¿Qué otras métricas de avance se suelen usar tanto en métodos ágiles como tradicionales?

Para los creadores del Manifiesto Ágil, el software funcionando es la principal medida de avance. Otras métricas que suelen utilizarse para medir el avance son:

- **Agility Index Measurements (AIM):** asigna puntaje a los proyectos sobre una serie de factores de agilidad.
- **Agility Measurement Index (AMI):** asigna puntaje a los desarrollos en cinco dimensiones: duración, riesgo, innovación, esfuerzo e interacción.