

Metodología Scrum v/s Cascada

Jhon Lopes-Camilo Martínez

October 16, 2013

1 Metodología Scrum

Primero que nada definiremos lo que es una metodología ágil se entiende como los métodos de la ingeniería del software basados en el desarrollo iterativo e incremental, donde los requerimientos y soluciones evolucionan mediante la colaboración de grupos autos organizados y multidisciplinarios. Ahora scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo uno de sus principales objetivos es poder maximizar el retorno más bien conocido como ROI. En Scrum se caracteriza por tener entregas o avances parciales del producto final. En este sentido esta metodología esta principalmente orientado a entornos de mucha complejidad. Donde se prioriza siempre la innovación, la competencia, flexibilidad y alta complejidad. El proceso de esta en un proyecto se genera en periodos temporales, cortos y fijos. Cada iteración tiene que entregar un resultado completo. Existen formas de planificar cada iteración primero parte mediante la selección de requisitos más prioritarios esta actividad por lo general toma cuatro horas como máximo. Para luego planificar la iteración el equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos que se habían comprometidos. La manera en que se ejecuta la iteración es mediante una pequeña reunión todos los días de quince minutos como máximo en esta reunión cada miembro del equipo responde a tres preguntas básicas las cuales son: ¿Qué he hecho desde la última reunión de sincronización? - ¿Qué voy a hacer a partir de este momento? - ¿Qué impedimentos tengo o voy a tener? Donde también dentro de ese equipo presentan roles bien definido tales como Product Owner, Scrum master, equipos de desarrollo y otros roles menos claves que se conocen como roles auxiliares. Para concluir Scrum es una metodología que permite cumplir fácilmente las expectativas generadas por el cliente, tiene una alta capacidad de ser flexible al cambio, permite entregar una mayor calidad del software, permite tener predicciones de tiempo lo que se logra mediante los llamados sprint que permite estimar cuantos tiempos se demorara el equipo y por ultimo permite reducir radicalmente el riesgo ya que mediante las diferentes reuniones se pueden anticipar las diferentes coyunturas.

2 Metodología cascada

Ciclo de vida clásico o en cascada este enfoque del desarrollo de software ha sido el más utilizado y en la actualidad, pese a la aparición de metodologías ágiles, sigue siendo la solución predominante, si bien, en cada organización o en cada de proyecto se puede llevar a cabo con ciertas variantes. Las fases en que se divide el ciclo de vida clásico, pueden ser diferentes. Pero segun las que estableció Roger Pressman son : Analisis-Diseño-Codificaciones-Pruebas-Mantenimiento. En el ciclo de vida clásico en función de las modificaciones y/o correcciones que se realizan en una etapa será necesario la vuelta a fases previas para hacer coherente el proceso de desarrollo y los modelos. El principal inconveniente que se le ha achacado siempre a este tipo de ciclo de vida es que los usuarios tardan demasiado en ver los resultados, lo que hace que el tiempo transcurrido desde que se define el sistema hasta que está disponible sea lo suficientemente amplio como para que hayan ocurrido muchas cosas: desde que no estén la mayoría de las personas que participaron en la especificación, como cambios en los procesos, cambios de criterio, etc..., lo que provoca a hacer replanteamiento de los requisitos en etapas más tardías del desarrollo con el coste que eso conlleva (un refinamiento de los requisitos es razonable siempre y cuando no se superen unos límites) y a que muy probablemente el sistema finalmente disponible esté alejado de lo que realmente quiere el conjunto de usuarios en estos momentos que es distinto a lo que querían hace meses y a lo que querrán dentro de otros tantos. Otro inconveniente ligado con el anterior es que al tratarse el sistema como un todo, los modelos generados (catálogo de requisitos, casos de uso, etc...) serán lo suficientemente grandes como para no poder ser revisados y comprendidos en toda su magnitud, sobre todo por personal no informático. Esto crea una incertidumbre sobre los requisitos del sistema que más adelante traerá problemas. No todo es malo en este modelo, ya que describe un procedimiento racional y ordenado de desarrollo de software, la clave para su éxito o su fracaso es como se gobierne el mismo y las circunstancias que rodeen al proyecto en el momento de su ejecución. Además, existen variantes que ofrecen una mayor flexibilidad al mismo y que permite reducir sus riesgos.

3 Comparativas entre ambas metodologías

Desarrollo:

- El esquema de desarrollo en cascada se caracteriza por proponer actividades secuenciales bien definidas y planear un análisis intensivo de requerimientos. A diferencias scrum que propone actividades más flexibles donde el equipo tiene reuniones periódicas para ver el avance y cambios que podría tener el sistema, además la metodología scrum permite entregar distintos avances del software a medida que este avanza a diferencia del método de la cascada que solo entrega el software cuando este haya finalizado.

Equipo :

- Cascada plantea una estructura de trabajo más jerárquica y con poca participación del equipo por completo del sistema. En scrum se fijan distintos roles de manera que exista una alta comunicación entre todo el grupo de trabajo.

Costos:

- La metodología cascada es una muy costosa si uno quiere hacerle cambios al sistema una vez que ya está bien definido y el enorme cantidad de documentación que este necesita. A diferencia de Scrum que al trabajar por etapas permite devolverse y modificar constantemente según los requerimientos que el cliente solicite y a diferencia de la otra metodología se documenta lo esencialmente necesario.

Adaptibilidad

- Ambas Formas de trabajo permite la complementación con otras tipos de metodologías ágiles y no ágiles, teniendo muy buenos resultado.

References

[<http://www.northware.mx/2012/11/29/desarrollo-en-cascada-waterfall-vs-desarrollo-agile-scrum/>]

[<http://agile101.net/2009/09/08/the-difference-between-waterfall-iterative-waterfall-scrum-and-lean-in-pic>]

[<http://www.proyectosagiles.org/que-es-scrum>]

[http://es.wikipedia.org/wiki/Desarrollo_en_cascada]