

Desarrollo en Cascada (Waterfall) VS Desarrollo Agile-SCRUM

Por Jesus Demetrio Velázquez Camacho

Dentro de las organizaciones de desarrollo de aplicaciones existen dos grandes corrientes para la metodología en el desarrollo de un proyecto.

La que tradicionalmente conocemos como “**desarrollo en cascada o secuencial**” y las nuevas metodologías que proponen la generación de pequeños entregables en un esquema de actividades que se pueden solapar o traslapar, ya sea en forma secuencial o con un enfoque totalmente solapado.

Dentro de estas dos formas de trabajo, aquí analizamos las principales características del desarrollo en cascada con CMMI/RUP y el desarrollo ágil con SCRUM. Focalizaremos nuestra atención en entender un poco más la metodología Scrum y compararla con la metodología de desarrollo en cascada que es la más usada en la actualidad.

Características de Scrum

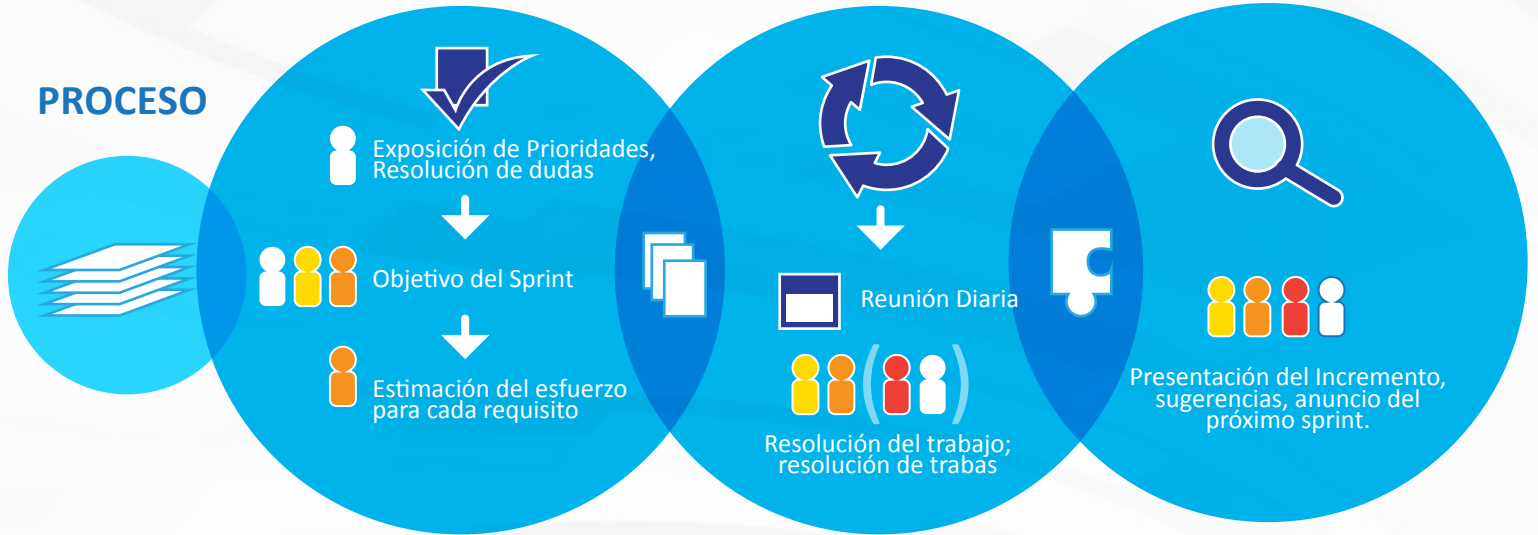
Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el **ScrumMaster**, que mantiene los procesos y trabaja de forma similar al director de proyecto, el **ProductOwner**, que representa a los **stakeholders** (interesados externos o internos), y el **Team** que incluye a los desarrolladores. Scrum está catalogada como una metodología de desarrollo AGILE con ciclos secuencias con solapamiento.

Scrum permite la creación de equipos auto organizado impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto.





Un principio clave de **Scrum** es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado *requirementschurn*), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, **Scrum** adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas "post-it" y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.




Scrum: Ficha Sinóptica






ROLES

-  **PROPIETARIO DEL PRODUCTO**
Determina las prioridades, una sola persona.
-  **SCRUM MANAGER**
Gestiona y facilita la ejecución del proceso.
-  **EQUIPO**
Construye el producto.
-  **INTERESADOS**
Asesoran y observan.

COMPONENTES

-  **PILA DEL PRODUCTO**
Relación de requisitos del producto, no es necesario excesivo detalle. Priorizados. Lista en evolución y abierta a todos los roles. El propietario del producto es su responsable y quien decide.
-  **PILA DEL SPRINT**
Requisitos comprometidos por el equipo para el sprint con nivel de detalle suficiente para su ejecución.
-  **INCREMENTO**
Parte del producto desarrollada en un sprint, en condiciones de ser usada (pruebas, codificación limpia y documentada).

REUNIONES

-  **PLANIFICACIÓN DEL SPRINT**
1 jornada de trabajo. El propietario del producto explica las prioridades y dudas del equipo. El equipo estima el esfuerzo de los requisitos prioritarios y se elabora la pila del sprint. El Scrum Manager define una frase del objetivo del sprint.
-  **REUNIÓN DIARIA**
15 minutos de duración, dirigida por el Scrum Manager sólo puede intervenir el equipo: ¿Qué hiciste ayer?, ¿Cuál es el trabajo para hoy?, ¿Qué necesitas?. Se actualiza la pila del sprint.
-  **REVISIÓN DEL SPRINT**
Informativa, aprox. 4hrs., moderada por el Scrum Manager, presentación del incremento, planteamiento de sugerencias y anuncio del próx. sprint.

SPRINT



Ciclo de desarrollo básico de SCRUM, de duración recomendada de 30 días en el que se desarrolla un incremento del producto.

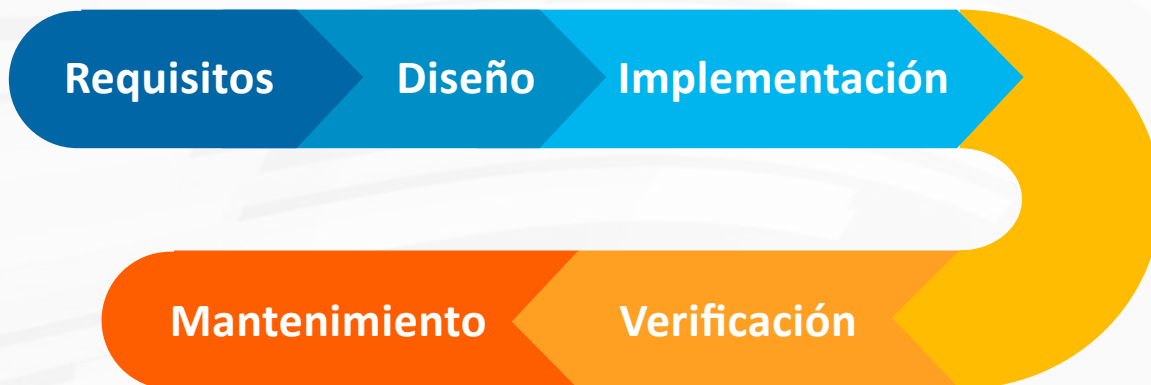
VALORES

- Empowerment y compromiso de las personas
- Foco en desarrollar lo prometido
- Transparencia y visibilidad del proyecto
- Respeto entre las personas
- Coraje y responsabilidad

Características del Desarrollo en Cascada comparadas con SCRUM.

El esquema de desarrollo en cascada se caracteriza por proponer actividades secuenciales, claramente agrupadas dentro de fases o ciclos del desarrollo del proyecto, propone hacer un análisis intensivo de requerimientos y se vuelve complicado volver a etapas previas del proyecto cuando se encuentran diferencias significativas en el alcance definido en etapas tempranas del mismo. El levantamiento de requerimientos es muy riguroso y los Analistas definen a priori todos los requerimientos funcionales y no funcionales relacionados con el proyecto. Normalmente, una fase no puede iniciar sin que la fase anterior haya sido revisada y aceptada por el cliente o usuario final, sin que esto signifique el sistema cumplirá con sus necesidades.

Esquema de Fases del Desarrollo en Cascada



Diferencias en el manejo de Requerimientos

Mientras que en el desarrollo en cascada se exige la aceptación de alcances previamente definidos a través de documentos como “Casos de Uso”, en el que se hacen referencias técnicas a partir de los requerimientos del usuario, en Scrum se propone el esquema de generar “Historias de Usuario” (Userstories) para entender y manejar el requerimiento desde el punto de vista de un usuario final de la aplicación. Para el desarrollo en cascada, normalmente solo se involucran los analistas de sistemas para el levantamiento de requerimientos sin involucrar a otros miembros del equipo de desarrollo (ejemplo: tester).

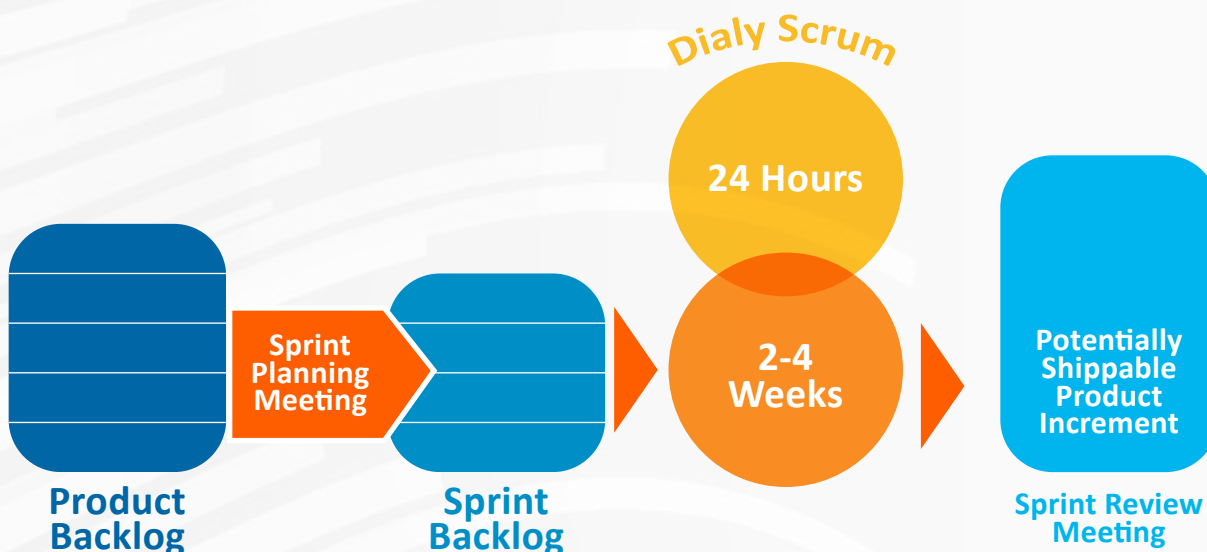
Problemas y características con este tipo de especificación

1. El alcance se congela rápidamente;
2. Se tiene un conocimiento claro de cuándo parar;
3. Aunque los requerimientos evolucionen, el alcance debe ser mantenido hasta que se genere un control de cambios; y
4. Los cambios en los requerimientos normalmente aparecen a lo largo del proyecto.

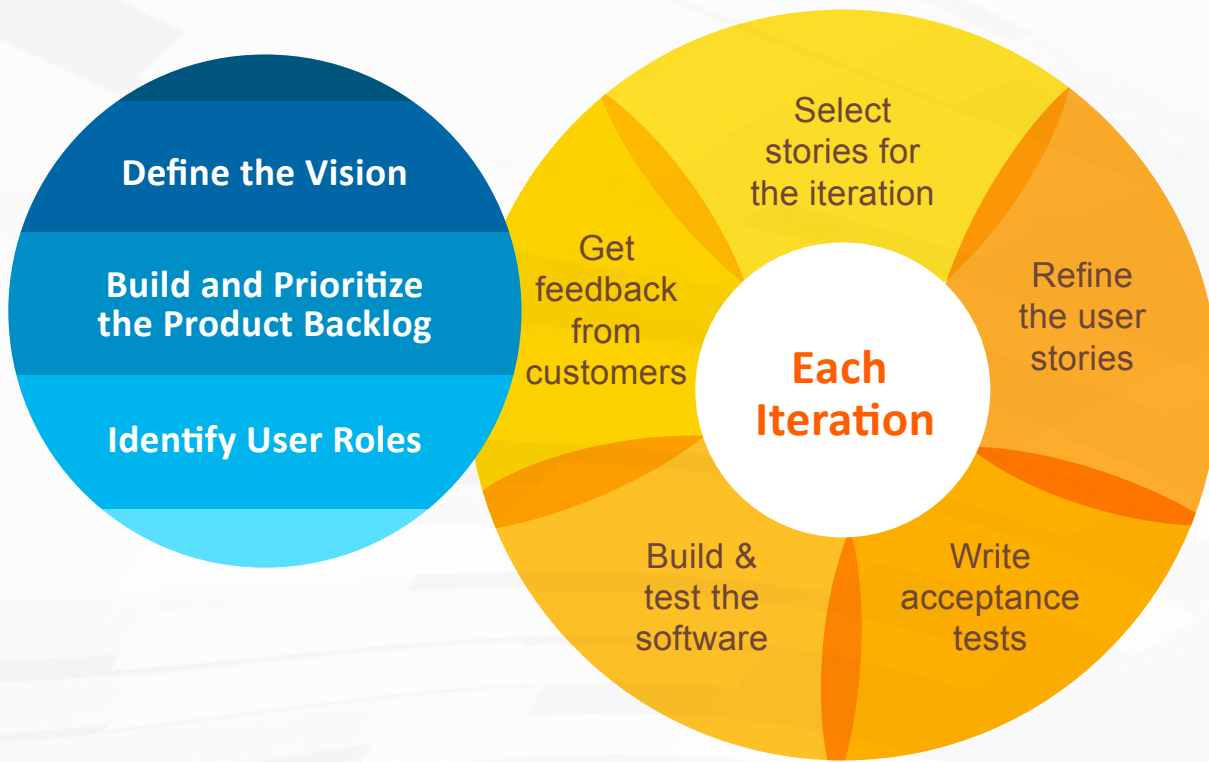
Manejo de requerimientos en Agile

Las metodologías que usan AGILE como Scrum, requieren que TODO el equipo de desarrollo esté involucrado en todas las etapas del proyecto en constante contacto con el usuario final de la aplicación. Los requerimientos Agile son representados como “Historias de Usuario” (Userstories), la cual es una pequeña descripción del requerimiento del usuario descrito en un lenguaje lo más aproximado a sus jerga diaria. Se hace un énfasis muy fuerte a que este tipo de comunicación esté orientada al usuario final y que sea fácil de entender. Estas historias definen alcances fáciles de identificar para poder efectuar planes iterativos para definición, revisión y confirmación de alcances. No se les exige a los usuarios finales que tengan que conocer desde un principio todos sus requerimientos. Sin embargo, sí se pide que no se cambien en los ciclos iterativos (sprints) que se definen para el desarrollo del producto.

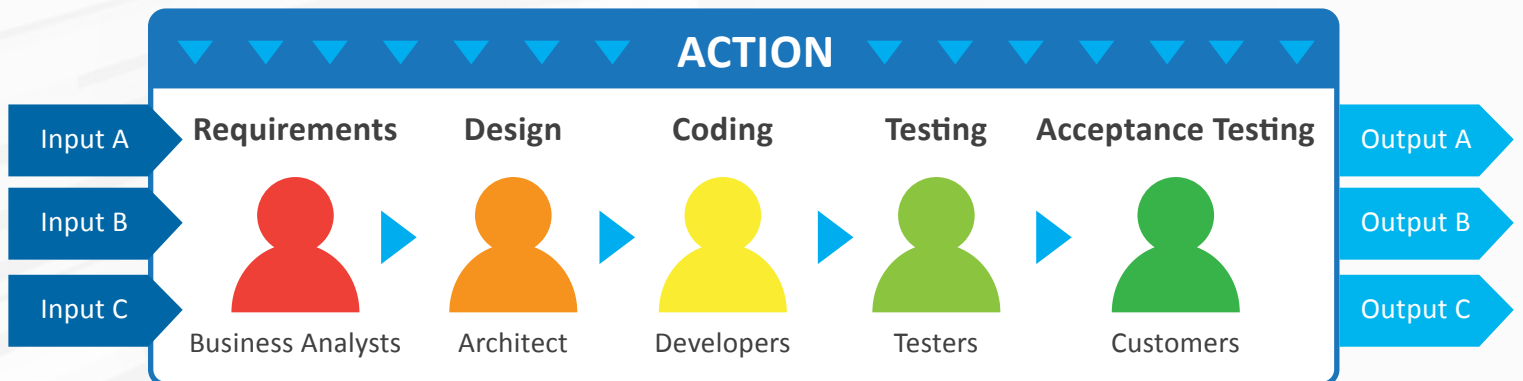
Modelo de iteración de Agile-Scrum

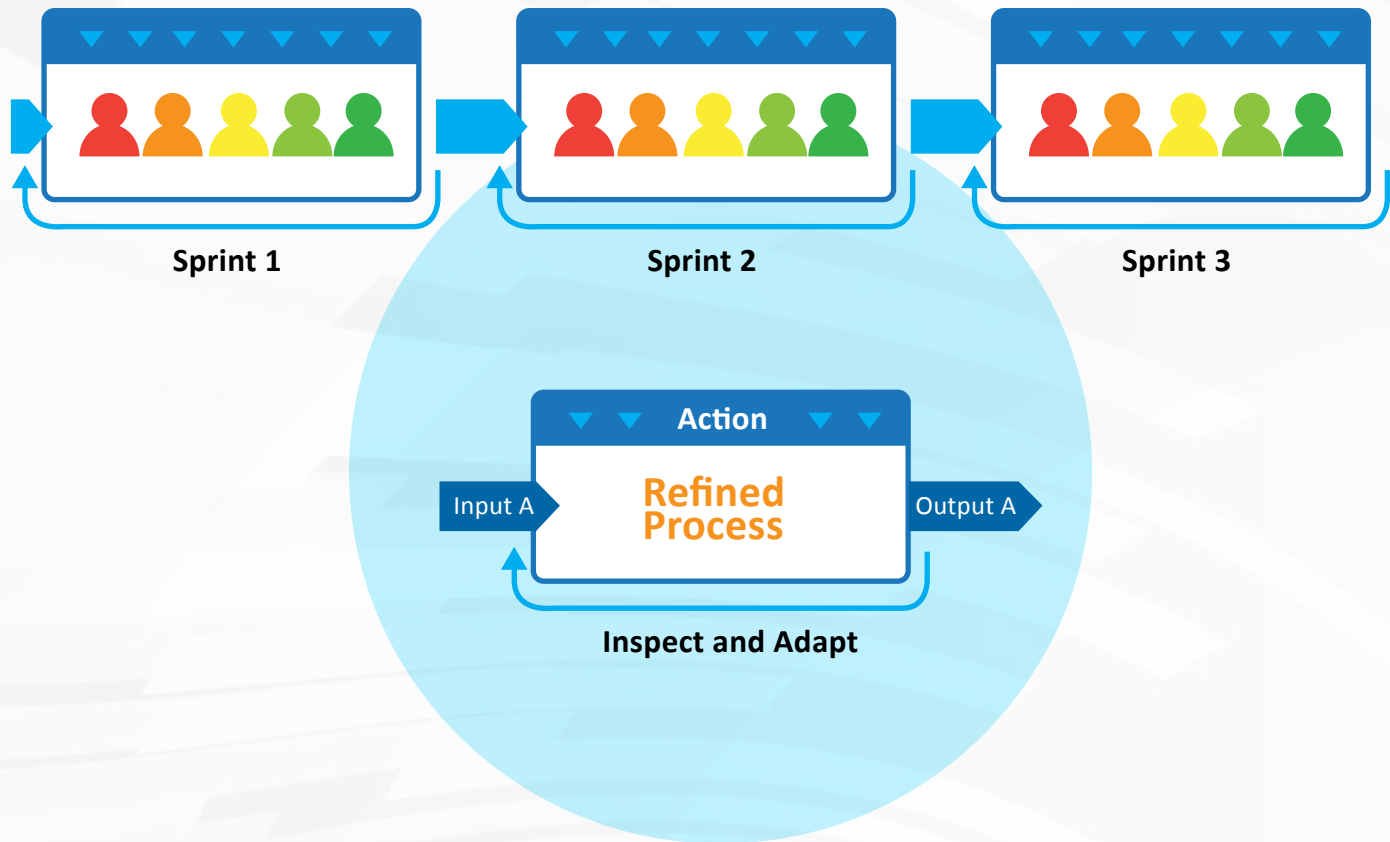


An Agile Requirements Process



Modelo gráfico del desarrollo en cascada comparado con el desarrollo Agile-Scrum





En el modelo que Microsoft propone para el desarrollo iterativo de productos, una de las principales características que sobresale de los métodos iterativos, es que el entendimiento de las necesidades del cliente final se vuelven más claras, se minimizan los malentendidos entre desarrolladores y usuarios finales.

Sin embargo, habrá que recordar que algunas metodologías, independientemente de sus grandes bondades, podrán ser las adecuadas para cierto tipo de usuarios finales y para los desarrolladores involucrados. Mientras que otras metodologías podrán representar un serio reto para su implementación.

Para concluir, diremos que si quieres entregar un producto equivocado, que no cumpla las necesidades de tus usuarios finales, no sigas una metodología; pero si tu intención es acercarte lo más que puedas a resolver una necesidad específica, lo mejor es usar una metodología, probarla, refinarla y adecuarla a tu entorno operativo.

Estoy seguro que siguiendo estas líneas, tendrás un mejor resultado al final de tu proyecto de desarrollo.

Si prefieres recibir **ayuda profesional** y evitar errores en la planeación financiera y de calidad en tu proyecto de desarrollo, te invito a que nos contactes. Somos una empresa especialista en desarrollo de aplicaciones, base de datos y aplicaciones para Iphone/Ipad.

Desarrollamos software basado en Microsoft .net, java, iOS y Android; y para aquellas empresas que sólo requieren la contratación directa de especialistas, proveemos consultores por proyecto, temporales o fijos con experiencia en las tecnologías más avanzadas para apoyar tú estrategia en sistemas de información y desarrollo de software.

Contáctanos

Interior de la República Mexicana
01 800 288 OPEN (6736)

Ciudad e México (D.F.)
(55) 5536 2968

Monterrey, Nuevo León
(81) 8262 1111

Desde Estados Unidos (U.S.A.)
(512) 853 9472

Síguenos en

facebook.com/northware

twitter.com/northwaremx

Nuestro correo electrónico
info@northware.mx



Northware[®]
Software Development