**Desarrollo ágil de software**

* Es un compromiso entre nada de proceso y demasiado proceso

**Manifiesto Ágil – 4 valores de las metodologías ágiles**

1. **Individuos e interacción** por sobre procesos y herramientas
2. **Software funcionando** por sobre documentación detallada
3. **Colaboración con cliente** por sobre negociación
4. **Respuesta a cambios** por sobre seguimiento de un plan

**Los 12 principios del manifiesto**

1. La prioridad es satisfacer al cliente entregando software funcionando en releases tempranos de manera frecuente
2. Aceptar cambios de requerimientos aun en etapas finales
3. Intervalos de desarrollo cortos (de 2 semanas a 1 mes)
4. Colaboración entre personal técnico y no técnico durante todo el proyecto
5. Trabajar con gente motivada
6. El medio de comunicación por excelencia es cara a cara
7. La mejor métrica es la cantidad de software funcionando
8. Mantener un ritmo regular de desarrollo
9. Atender siempre a la excelencia técnica del software
10. Simplicidad
11. La mejor arquitectura y diseño surge de un equipo auto-organizado
12. Realización periódica del auto-evaluaciones del equipo y ajustes de trabajo

Si bien en el proceso ágil la cantidad de documentación no es tan marcada o exigente, no forma parte de lo más importante que hace a un proceso ágil.

Otra de sus grandes diferencias con procesos no agiles es que son adaptables y no predictivos.

Se orientan hacia la gente en lugar de hacia un proceso

**TOP 3 beneficios de procesos ágiles**

* + Habilidad para manejar cambios en prioridades del 87%
  + Incremento en la producción del equipo de un 85%
  + Mejoramiento de la visibilidad del proyecto en un 84%

**Técnicas ágiles empleadas**

* Daily standups
* Retrospectiva
* Priorización de BackLog
* Planificación de iteraciones
* Iteraciones cortas

¿Cuándo es recomendable aplicar Agile?

Es recomendable cuando los problemas caen en la categoría de “complejos”

Dentro de complejos suelen estar el desarrollo de nuevos productos y knowledge work

**SCRUM**

SCRUM es un framework que permite crear su propio proceso para crear productos nuevos.

SCRUM es simple y puede ser implementado en pocos días, pero puede llevar mucho tiempo perfeccionarlo

SCRUM es empírico. Las metodologías rigurosas se basan en métodos definidos.

El control de SCRUM se basa en inspecciones frecuentes y ajustes pertinentes

**Valores de SCRUM**

* Coraje
* Foco
* Compromiso
* Respeto
* Apertura

**Cimientos de SCRUM**

* Auto-organización – **Proceso donde pequeños grupos de trabajo se organizan entre ellos y se reparten la carga de tareas, enfocados en un objetivo claro y sus restricciones**
* Empírico – **Procesos definidos y planificación detallada son remplazados por por inspecciones frecuentes, revisiones just in time y ciclos adaptativos**
* Time Boxing – **Crea el ritmo que guía al desarrollo**
* Colaboración – **Líderes de SCRUM, diseñadores del producto y clientes colaboran con los desarrolladores, no los dirigen**
* Priorización – **Trabajar en lo más importante y no perder tiempo en lo que no agrega valor**

**Roles de SCRUM**

**Product Owner**

* Controla y gestiona el backlog
* Una persona, no un comité
* Acciones visibles
* Nadie puede decirle al equipo una prioridad diferente
* Crear ítems del product backlog
* Priorizar ítems del backlog
* Hablar con los involucrados
* Participar de los reviews
* Hacer seguimiento del release

**SCRUM Master**

* Responsable de que las prácticas, valores y reglas se realicen
* Es responsable de que se resuelvan impedimentos y aplicar soluciones rápidas
* Nexo entre gerencia y equipo
* Dirige las reuniones diarias, comparando progreso real con planeado
* Couchear equipo

**Developement’s Team**

* 7+-2 personas. Si hay más de 8 personas,
* hacer varios equipos trabajando sobre el
* mismo Backlog.
* Autónomos y auto-organizados
* Compromiso de entregar
* un conjunto de ítems del
* Backlog al final de un Sprint
* Libertad de acción
* Limitado por estándares y
* convenciones
* organizacionales
* No hay roles
* Todos codifican

**Product Backlog**

Es una cola priorizada de funcionalidades técnicas y de negocio, que necesitan ser desarrolladas

**Estimaciones del Backlog**

• Las estimaciones son un esfuerzo colaborativo entre las partes

• Las estimaciones son iterativas

• Si un ítem no puede ser

debidamente estimado,

se debe dividir en el Backlog.

• Los estimados sirven de base

para la funcionalidad en el Sprint.

Backlog del Sprint

• El equipo determina que debe hacerse para cumplir el objetivo del Sprint

• El dueño del producto suele asistir

• Se realiza una lista de tareas que tomen de 4 a 16 horas para

completarse

• El Backlog no se modifica durante el desarrollo

• Si el equipo descubre que el Backlog no puede realizarse, el Scrum

Master y el dueño del producto determinan si:

• Algún ítem puede ser removido del backlog

• Alguna funcionalidad puede eliminarse

Sprint Demo/review

• Reunión informativa de unas 4 horas

• El equipo presenta el incremento desarrollado a gerentes, clientes, usuarios y el dueño del producto

• Informal

• Regla de 2 hs preparación

• No usar diapositivas

• Todo el equipo participa

• Se invita a todo el mundo

• Se reportan los problemas encontrados

• CUALQUIER ítem puede ser agregado, eliminado, re-priorizado

• Se estima el nuevo Sprint y se asignan tareas

Sprint retrospective

• Periódicamente, se echa un vistazo a lo que funciona y lo

que no

• Normalmente 1hr a 4hs

• Se realiza luego de cada sprint

• Todo el equipo participa

• Scrum Master

• Product Owner

• Equipo

• Posiblemente clientes y otros

Story time

• El equipo se reúne con el PO para discutir ítems del backlog de alta

prioridad, determinar su criterio de aceptación y asignar la

priorización a cada nuevo ítem.

• Esto ocurre inicialmente antes del desarrollo y después

iterativamente en cada sprint.

Sprint

• Período fijo de tiempo, sugerido 30 días

• Dentro del Sprint

• No se puede cambiar el alcance

• No se puede agregar

funcionalidad

• No se pueden modificar las

reglas del equipo

Reglas de un Sprint

• Foco en la tarea

• SIN interrupciones

• SIN cambios

• El mismo equipo puede descubrir más trabajo a ser hecho

25

Mecánica de los Sprints

• En un Sprint se congelan 3 de las 4 variables de un proyecto:

• Tiempo: 30 días

• Costo: salarios + ambiente

• Calidad: generalmente un estándar organizacional

• El Equipo puede cambiar funcionalidad siempre y cuando cumpla con

el objetivo del Sprint.

Tareas obligatorias

• En un Sprint hay 2 tareas mandatorias

• Reuniones Scrum diarias: participan TODOS los miembros del equipo

• Backlog del Sprint: debe estar actualizado y con los últimos estimados de los

Desarrolladores

**HERAMIENTAS DE SCRUM**

**Tarjeta**

****

**TABLERO**

• Story

• To Do

• Work In Process

• To Verify

• Done

BURNDOWN Chart

**Diagnóstico?**

\_ No todo lo incluido en el sprint será entregado

en la fecha propuesta. El backlog no fue

modificado para alcanzar la fecha (remover

requerimientos o modificar la fecha)

\_ Posibles causas:

\_ Pobres técnicas de estimación

\_ Mal manejo de riesgos

\_ Rotación del equipo o renuncias

\_ Este gráfico muestra que el Scrum Master (SM)

y el Product Owner (PO) no reaccionaron para

evitar esta situación.

\_ Evite llegar a este estadio!

3 roles

• Scrum Master

• Scrum team

• Product owner

• 3 Entregables

• Product backlog

• Sprint Backlog

• Producto de software potencialmente listo para producción

• 4 Reuniones

• Sprint Planning

• Daily

• Sprint Demo

• Restropectivas

• Story Time/Grooming (opciona

**MÉTRICAS**

Capacidad

• **Capacidad es...**

Una estimación de cuanto trabajo puede completarse en un período de

tiempo dado.

Basado en la cantidad de tiempo ideal disponible del equipo.

• **...Se puede medir en...**

Esfuerzo (horas)

Puntos de Historia (Story Points)

Capacidad: consideraciones

• Individuos calculan capacidad realista

• Aplicar estimaciones honestas a sus tareas

• Considerar una capacidad máxima de 50%-70%

• Comprender la capacidad a largo plazo con la velocidad y los

puntos de historia-

• ¿Cuál es el promedio de finalización de un punto de historia

para un equipo/individuo?

44

87

Velocidad

*Velocidad es...*

• Una observación empírica de la capacidad del equipo para completar el

trabajo por iteración.

• Comparable entre iteraciones para un equipo dado y un proyecto dado

*...y no es...*

• una estimación

• un objetivo a alcanzar

• comparable entre equipos

• comparable entre proyectos