

# Metodologías de desarrollo de software (RUP- SCRUM)

Prof. Mg. Marisa Daniela Panizzi

# Proceso Unificado

- Es una metodología de desarrollo de software genérica que puede utilizarse en gran variedad de sistemas.
- Utiliza el Lenguaje Unificado de Modelado (UML) para diagramar los esquemas necesarios en cada etapa del desarrollo.

# Proceso Unificado: Perspectivas

Los procesos genéricos de desarrollo contemplan solo un enfoque del proceso.

El Proceso Unificado se describe desde tres perspectivas:

- Dinámica (fases sobre el tiempo)
- Estática (actividades – flujos de trabajo)
- Práctica, sugiere buenas prácticas para el proceso

# Proceso Unificado - Claves

- Dirigido por Casos de Uso.
- Centrado en la Arquitectura.
- Iterativo e Incremental.

# Proceso Unificado: Dirigido por casos de uso

- Debe conocerse lo que el usuario (algo o alguien – persona o sistema) necesita y desea.
- Son un fragmento de funcionalidad del futuro software.
- El conjunto de casos de uso (modelo de CU) describe la funcionalidad total del sistema.

**¿Qué debe hacer el sistema... para  
cada usuario?**

Además de representar la funcionalidad del software guían su diseño, implementación y prueba.

**LOS CASOS DE USO GUÍAN EL PROCESO DE DESARROLLO  
COMPLETO, SON LA BASE PARA LA CONSTRUCCIÓN DE MODELOS  
DE DISEÑO E IMPLEMENTACIÓN.**

# Proceso Unificado: Centrado en la arquitectura

La arquitectura:

- Se describe desde diferentes vistas.
- Enuncia aspectos estáticos y dinámicos del sistema: necesidades de usuarios, plataforma, rendimiento, fiabilidad, software reutilizable, etc.
- Es una vista de diseño completo con las características más destacadas, sin detalles.

## Proceso Unificado: Casos de uso y arquitectura

- Los casos de uso representan la función del sistema mientras que la arquitectura representa su forma.
- Deben interaccionar y equilibrarse.
- Ambos deben evolucionar en paralelo.

# Proceso Unificado: Iterativo e Incremental

- Propone definir el trabajo en partes de menor tamaño: miniproyectos.
- Cada miniproyecto es una iteración, su resultado es un incremento (crecimiento del producto).
- Las iteraciones deben estar «controladas»: seleccionarse y ejecutarse de una forma planificada



# Proceso Unificado: Iterativo e Incremental

¿Cómo se selecciona lo que se implementará en una iteración?

- Se seleccionan los casos de uso (funcionalidades) que amplíen la utilidad del producto.
- Trata los riesgos más importantes

Al ser miniproyectos comienzan con casos de uso, que continúan en análisis, diseño, implementación y prueba, finalizando en código ejecutable.

# Proceso Unificado: Iterativo e Incremental

Si una iteración no cumple sus objetivos, deben revisarse las decisiones previas. ➔ AUMENTO DE COSTO DE PROYECTO.

## Beneficios del proceso iterativo controlado:

- Ante el error en una iteración se pierde el costo solo de ese fragmento y no del producto entero.
- Reduce el riesgo de entrega retrasada, los errores se detectan al comienzo cuando hay tiempo para solucionarlos.
- Acelera el ritmo de desarrollo, hay obligaciones de entrega a corto plazo.
- Reconoce que no pueden conocerse de antemano todas las necesidades del usuario, maneja más fácilmente los cambios.

# El ciclo de vida del Proceso Unificado

- El proceso unificado repite una serie de ciclos que se constituyen el ciclo de vida del sistema.
- El resultado de cada ciclo es una nueva versión del producto.
- Cada ciclo se divide en cuatro fases: INICIO – ELABORACIÓN – CONSTRUCCIÓN – TRANSICIÓN.
- Cada fase se divide en iteraciones.

# El ciclo de vida del Proceso Unificado



# El producto

Cada ciclo produce una nueva versión del sistema. Está compuesto por:

- Para el usuario: código fuente que pueda compilarse y ejecutarse + manuales
- Pero además deben incluirse: requerimientos, casos de uso y pruebas, arquitectura y modelo visual que incluye todos los artefactos modelados con UML.

**¿Por qué y para quienes deben incluirse?**

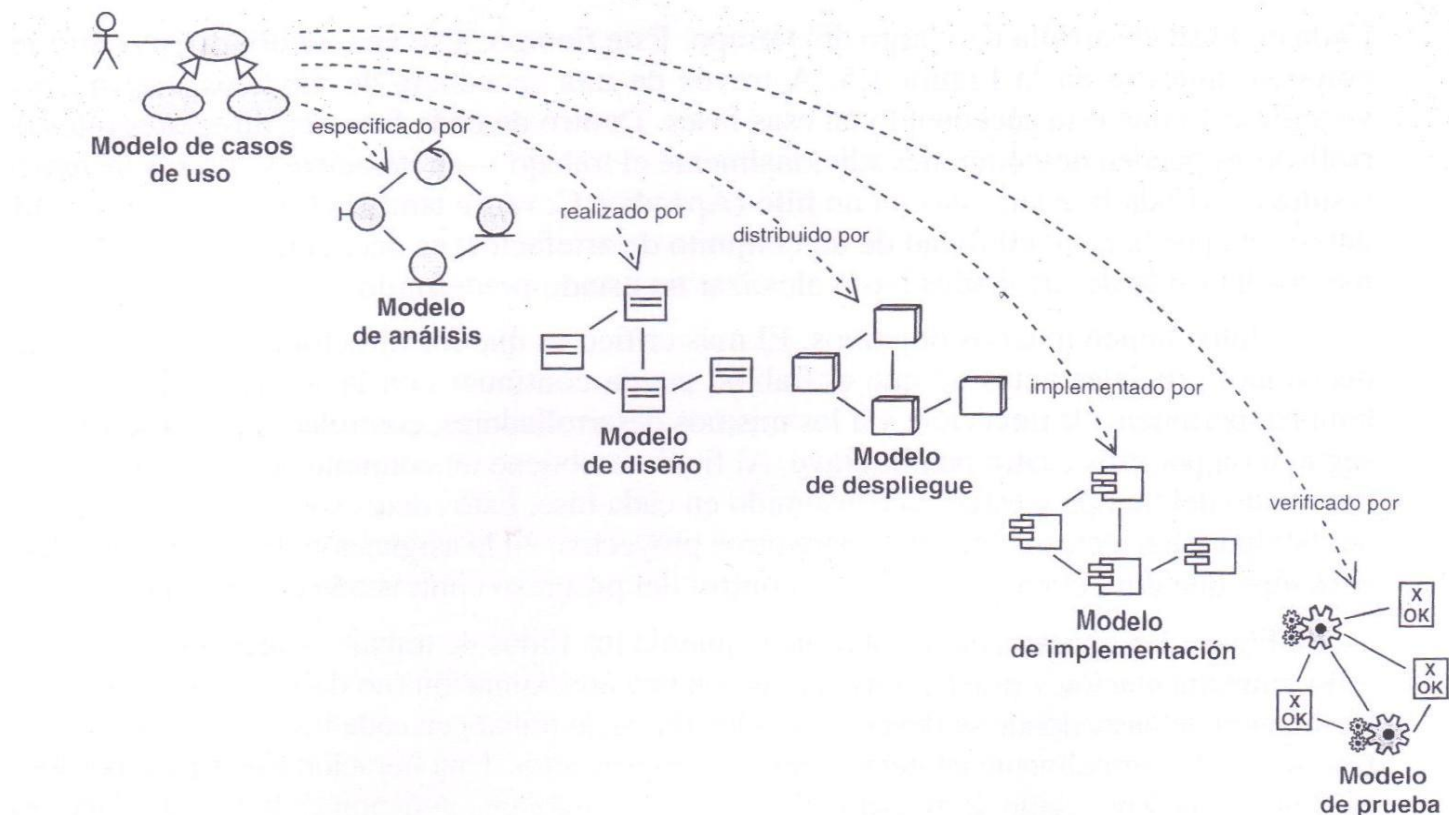
# Artefactos

- Término general utilizado para cualquier información creada, modificada o utilizada durante el proceso de desarrollo del sistema.
- Ejemplos: diagramas UML y sus textos, bocetos de interfaz de usuario, prototipos, procedimientos de pruebas.
- Existen artefactos: de ingeniería y de gestión

# Los modelos del Proceso Unificado

- Son el artefacto más interesante del Proceso Unificado.
- Cada modelo representa una vista del sistema.
- También puede agregarse un modelo de negocio o de dominio que describa el contexto en que funciona el sistema.
- Todos los modelos están relacionados.
- Juntos, representan el sistema como un todo.
- Los modelos poseen dependencia de trazabilidad.

# Los modelos del proceso unificado

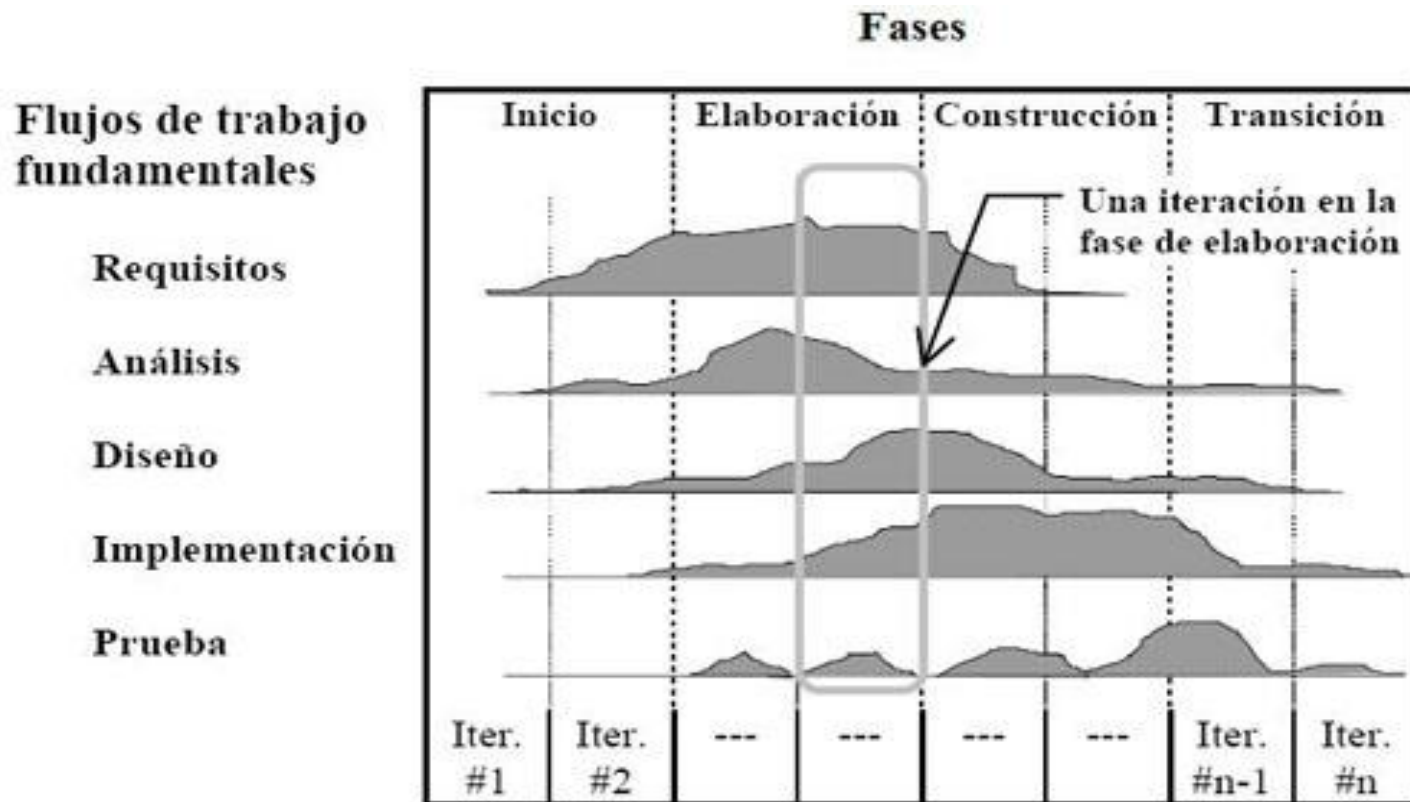




# Fases dentro de un ciclo

- Cada ciclo se desarrolla en el tiempo y se divide en cuatro fases.
- Cada fase puede descomponerse en iteraciones con sus productos resultantes.
- Cada fase termina con un **HITO**, este se determina por encontrarse disponibles ciertos artefactos.
- Los hitos permiten: tomar decisiones antes de pasar a la fase siguiente y controlar el progreso del trabajo. También permiten obtener datos para estimaciones de futuros proyectos.

# Fases dentro de un ciclo



## Fases dentro de un ciclo

- **Fase de inicio:** descripción del producto final y análisis de negocio para el producto.
- **Fase de elaboración:** Se especifican casos de uso y se diseña la arquitectura del sistema. Existen todos los modelos de los casos de uso más críticos. Se obtiene una línea base de la arquitectura. Puede planificarse actividades y estimar recursos para finalizar el proyecto.

## Fases dentro de un ciclo

- **Fase de construcción:** Se crea el producto. La arquitectura es estable pero aún sujeta a mejoras. Se utilizan la mayoría de los recursos. Se contemplan todos los casos de uso acordados con cliente.
- **Fase de transición:** Se obtiene una versión beta que prueban un grupo reducido de usuarios, se identifican defectos. Se corrige y se prueba con la totalidad de usuarios. Incluye formación del cliente, ayuda, asistencia y corrección de errores tras la entrega.

# Un proceso integrado

- Basado en componentes.
- Utiliza modelado visual (UML).
- Se sostiene sobre tres ideas.
- Es un proceso polifacético: ciclos, fases, flujos de trabajo, etc. integrado en un marco de trabajo.

# Metodologías Ágiles

## Un poco de historia.....

- A fines de la década de los '90, aparece un nuevo movimiento conformado por profesionales de la industria, denominado **movimiento heterodoxo**.
- Su propuesta para el desarrollo de software se basaba en:
  - Un **modelo de desarrollo incremental** (pequeñas entregas con ciclos rápidos),
  - **Cooperativo** (desarrolladores y usuarios trabajan juntos en estrecha comunicación),
  - **Directo** (el método es simple y fácil de aprender) y
  - **Adaptativo** (capaz de incorporar los cambios)

# Metodologías Ágiles

## Los 12 Principios que rigen a la comunidad MAs:

1. Nuestra prioridad más alta es satisfacer al cliente a través de la entrega temprana y continua de software valioso.
2. Los requerimientos cambiantes son bienvenidos, incluso cuando llegan tarde en el desarrollo. Los procesos ágiles se pliegan al cambio en procura de una ventaja competitiva para el cliente.
3. Entregar con frecuencia software que funcione, desde un par de semanas hasta un par de meses, con preferencia por las escalas de tiempo más breves.
4. La gente de negocios y los desarrolladores deben trabajar juntos cotidianamente a través de todo el proyecto.

# Metodologías Ágiles

- Principios que rigen a la comunidad MAs:
  - 5. Construir proyectos en torno de individuos motivados. Darles la oportunidad y el respaldo que necesitan y procurarles confianza para que realicen la tarea.
  - 6. La forma más eficiente de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
  - 7. El software que funciona es la medida primaria de progreso.
  - 8. Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante indefinidamente.
  - 9. La atención continua a la excelencia técnica enaltece la agilidad.



# Metodologías Ágiles

- Principios que rigen a la comunidad MAs:

- I 0. La simplicidad (el arte de maximizar la cantidad de trabajo que no se hace) es esencial.
- I 1. Las mejores arquitecturas, requerimientos y diseños emergen de equipos que se auto-organizan.
- I 2. A intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo, y ajusta su conducta en consecuencia.

## ***¿Quiénes eran los otros o el otro movimiento?.....***

### **Movimiento ortodoxo o las metodologías ortodoxas**

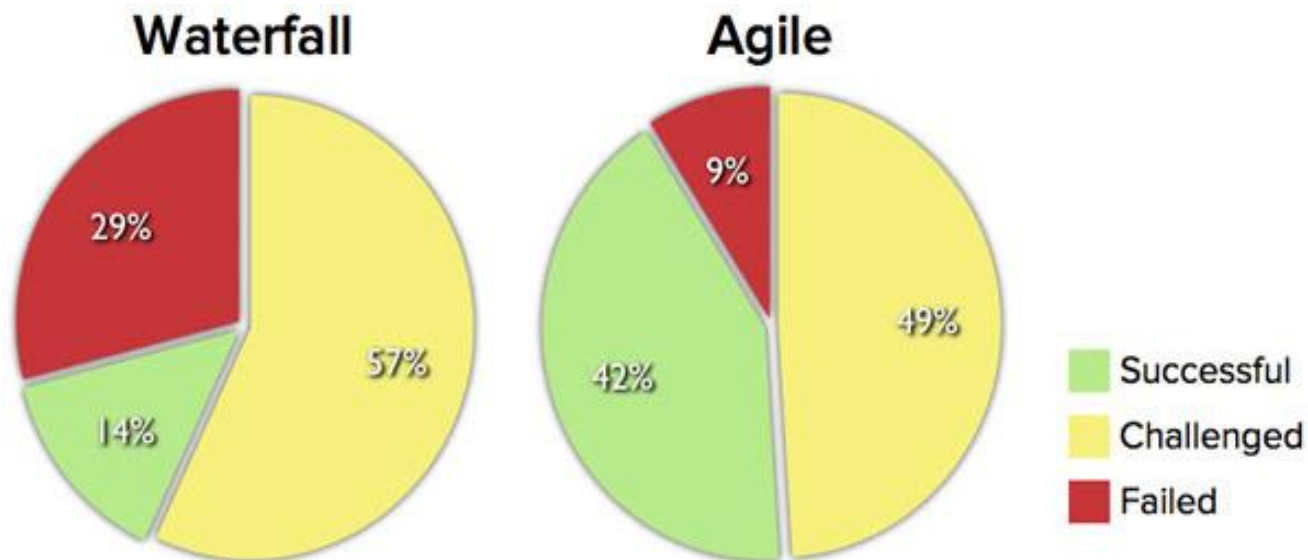
- Comienzan con la elicitación y el análisis completo de los requerimientos del usuario.
- Después de un largo período de intensa interacción con usuarios y clientes, los ingenieros establecen un conjunto definitivo y exhaustivo de rasgos, requerimientos funcionales y no funcionales
- Esta información se documenta en forma de especificaciones para la segunda etapa.
- El diseño, en el que los arquitectos, trabajando junto a otros expertos en temas puntuales (como ser estructuras y bases de datos), generan la arquitectura del sistema.
- Luego los programadores implementan ese diseño bien documentado y finalmente el sistema completo se prueba y se despacha.

# Metodologías Ágiles

- El Manifiesto Ágil propiamente dicho planteaba:
- Estamos poniendo al descubierto formas mejores de desarrollo de software, haciéndolo y ayudando a otros a que lo hagan. A través de este trabajo hemos llegado a valorar:
  - Los individuos y la interacción por encima de los procesos y herramientas.
  - El software que funciona por encima de la documentación abarcadora.
  - La colaboración con el cliente por encima de la negociación contractual.
  - La respuesta al cambio por encima del seguimiento de un plan.

<http://agilemanifesto.org/>, 2001.

# El porqué de las Metodologías Ágiles?



Source: The CHAOS Manifesto, The Standish Group, 2012.

# Visión General de Scrum

- Scrum: Un marco de trabajo dentro del cual las personas pueden afrontar complejos problemas adaptativos, a la vez que entregan productos del máximo valor posible de forma productiva y creativa.
- Scrum es:
  - Ligero
  - Fácil de entender
  - Extremadamente difícil de llegar a dominar

# Visión General de Scrum

- Scrum: no es un proceso o una técnica para construir productos.
- Es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas

# Marco de trabajo de Scrum

- Consiste en los Equipos Scrum y en sus roles, eventos, artefactos y reglas asociadas.
- Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso.

*Nota: Las reglas de Scrum vinculan a los eventos, roles y artefactos, rigiendo las relaciones e interacciones entre ellos.*

# Fundamentos teóricos de Scrum

- Se fundamenta en la teoría empírica de control de procesos, o empirismo.
- El empirismo asegura que el conocimiento procede de la experiencia y *de tomar decisiones basándose en lo que se conoce*.
- Scrum emplea una aproximación *iterativa e incremental* para optimizar la predictibilidad y controlar el riesgo.



# Fundamentos teóricos de Scrum

- Tres pilares soportan toda implementación del control empírico de procesos:
  - Transparencia,
  - Inspección y
  - Adaptación

# Fundamentos teóricos de Scrum

## ■ **Transparencia:**

- Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado.
- La transparencia requiere que dichos aspectos sean definidos por un estándar común, de modo que los observadores compartan un entendimiento común de lo que se está viendo.
- Ejemplo: Todos los participantes deben compartir un lenguaje común para referirse al proceso.

# Fundamentos teóricos de Scrum



## ■ Inspección:

- Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones no deseables.
- Su inspección no debe ser tan frecuente como para que interfiera en el trabajo.
- Las inspecciones son más beneficiosas cuando son realizadas de forma diligente por inspectores expertos, en el mismo lugar de trabajo.

# Fundamentos teóricos de Scrum

## ■ **Adaptación:**

- Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados.
- Dicho ajuste debe ser realizado cuanto antes para minimizar desviaciones mayores.

# Fundamentos teóricos de Scrum

- Scrum prescribe formalmente cuatro oportunidades para la inspección y adaptación:
  - Reunión de Planificación del Sprint (Sprint Planning Meeting)
  - Scrum Diario (Daily Scrum)
  - Revisión del Sprint (Sprint Review)
  - Retrospectiva del Sprint (Sprint Retrospective)

# El Equipo Scrum (Scrum Team)

## ■ Los roles son:

- Dueño de Producto (Product Owner),
- El Equipo de Desarrollo (Development Team)
- Scrum Master



# El Equipo Scrum (Scrum Team)

- Los equipos de Scrum son autoorganizados y multifuncionales.
  - **Equipos autoorganizados:** eligen la mejor forma de llevar a cabo su trabajo, en lugar de ser dirigidos por otros externos al equipo.
  - **Equipos multifuncionales:** tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otros que no son parte del equipo.
- El modelo de equipo en Scrum está diseñado para optimizar la *flexibilidad*, la *creatividad* y la *productividad*.

# El Equipo Scrum

## El Dueño de Producto (Product Owner)

- Es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo.
- El cómo se lleva esto a cabo puede variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos.
- El Dueño de Producto es la única persona responsable de gestionar la Pila de Producto (Product Backlog).



# El Equipo Scrum

## El Equipo de Desarrollo (Development Team)

- Consiste en los profesionales que desempeñan el trabajo de entregar un Incremento de producto “Hecho”, potencialmente utilizable, al final de cada Sprint.
- Sólo los miembros del Equipo de Desarrollo participan en la creación del Incremento.
- Los Equipos de Desarrollo se estructuran y reciben poderes por parte de la organización para organizar y gestionar su propio trabajo.

# El Equipo Scrum

## El Equipo de Desarrollo (Development Team)

- Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas o áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.
- Los Equipos de Desarrollo no contienen sub-equipos dedicados a dominios concretos como pruebas o análisis de negocio.
- **Tamaño del equipo:**
  - El tamaño óptimo del Equipo de Desarrollo es lo suficientemente pequeño como para permanecer ágil
  - Lo suficientemente grande como para completar una cantidad de trabajo significativa.
  - Tener más de nueve miembros en el equipo requiere demasiada coordinación.
  - Los Equipos de Desarrollo grandes generan demasiada complejidad como para que pueda ser gestionada mediante un proceso empírico.

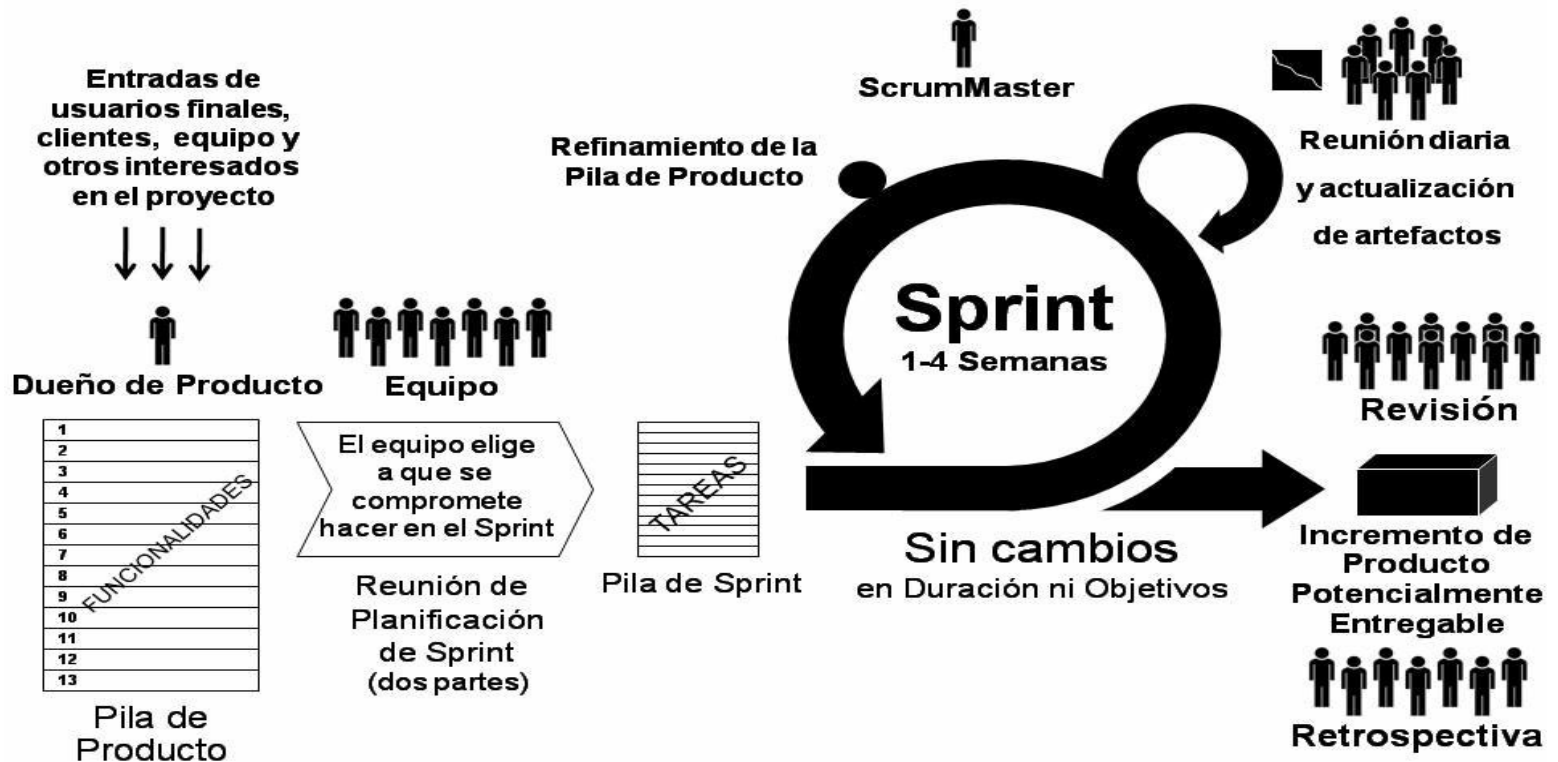
# El Equipo Scrum

## El Scrum Master

- Es el responsable de asegurar que Scrum es entendido y llevado a cabo, hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.
- Es un líder servil, al servicio del Equipo Scrum.

# Scrum

## Roles, Artefactos y Eventos principales



# Scrum

## Eventos principales

- Se utilizan eventos en la forma de bloques de tiempo (**timeboxes**), de modo que todos tienen una duración máxima.
- Esto asegura que se emplee una cantidad apropiada de tiempo en la planificación, de forma que no se admita desperdicio en este proceso de planificación.
- Además del propio Sprint que es un contenedor del resto de eventos, cada uno de los eventos de Scrum constituye una oportunidad para la inspección y adaptación de algún aspecto.

# Scrum

## Eventos principales

### El Sprint

- El corazón de Scrum es el Sprint, un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Hecho”, utilizable y potencialmente entregable
- Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo
- Contienen y Consisten:
  - Reunión de Planificación del Sprint (Sprint Planning Meeting)
  - los Scrums Diarios (Daily Scrums)
  - el trabajo de desarrollo
  - la Revisión del Sprint (Sprint Review)
  - la Retrospectiva del Sprint (Sprint Retrospective).

# Scrum

## Eventos principales

### Durante el Sprint:

- No se realizan cambios que afectarían al Objetivo del Sprint (Sprint Goal)
- La composición del Equipo de Desarrollo se mantiene constante;
- Los objetivos de calidad no disminuyen
- El alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de Desarrollo a medida que se va aprendiendo más del negocio.

# Scrum

## Eventos principales

### Reunión de Planificación de Sprint (Sprint Planning Meeting).

- Se planifica el trabajo a realizar durante el Sprint.
- Este plan es creado mediante el trabajo colaborativo del Equipo Scrum al completo.
- La Reunión de Planificación de Sprint está restringida a una duración de ocho horas para un Sprint de un mes. Por ejemplo, los Sprints de dos semanas tienen una Reunión de Planificación de Sprint de cuatro horas.
- La Reunión de Planificación de Sprint consta de dos partes:
  - **Primera Parte. ¿Qué será entregado en el Incremento resultante del Sprint que comienza?**
  - **Segunda Parte. ¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?**



# Scrum

## Eventos principales

- **Primera Parte. ¿Qué será entregado en el Incremento resultante del Sprint que comienza?**
- Equipo de Desarrollo trabaja para hacer una predicción de la funcionalidad que será desarrollada durante el Sprint.
- **Entrada:** Pila de producto. Presentada por el dueño del producto.
- El número de elementos de la Pila de Producto seleccionados para el Sprint depende únicamente del Equipo de Desarrollo. Sólo el Equipo de Desarrollo puede evaluar qué es capaz de llevar a cabo durante el Sprint que comienza.
- El equipo Scrum elabora un **Objetivo del Sprint (*Sprint Goal*)**: es una meta que será conseguida durante el Sprint mediante la implementación de la Pila de Producto.

# Scrum

## Eventos principales

- **Segunda parte: ¿Cómo se conseguirá completar el trabajo seleccionado?**
- El Equipo de Desarrollo decide cómo construirá esta funcionalidad durante el Sprint, para formar un Incremento de producto “Hecho”.
- *Los elementos de la Pila de Producto seleccionados para este Sprint + el plan para entregarlos = Pila de Sprint (Sprint Backlog)*

# Scrum

## Eventos principales

### **Scrum Diario (Daily Scrum).**

- Es una reunión restringida a un bloque de tiempo de 15 minutos, para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas.
- Se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una predicción acerca del trabajo que podría ser completado antes del siguiente.
- Se realiza a la misma hora y en el mismo lugar todos los días
- Cada miembro del Equipo de Desarrollo explica:
  - ¿Qué se ha conseguido desde la última reunión?
  - ¿Qué se hará antes de la próxima reunión?
  - ¿Qué obstáculos se encuentran en el camino?

# Scrum

## Eventos principales

### **Scrum Diario (Daily Scrum). Algunas Reglas:**

- El Scrum Master asegura que participen sólo los miembros del Equipo de Desarrollo
- Los límites del bloque de tiempo de 15 minutos
- Los Scrum Diarios mejoran la comunicación, eliminan la necesidad de mantener otras reuniones
- Constituye una reunión clave de *inspección y adaptación*.

# Scrum

## Eventos principales



# Scrum

## Eventos principales

### **Revisión de Sprint (Sprint Review):**

- Se lleva a cabo al final de un Sprint
- Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se ha hecho durante el Sprint
- Se trata de una reunión informal
- Se trata de una reunión restringida a un bloque de tiempo de cuatro horas para Sprints de un mes. Para Sprints más cortos, se reserva un tiempo proporcionalmente menor

**El resultado de la Revisión de Sprint es una Pila de Producto revisada, que define los elementos de la Pila de Producto posibles para el siguiente Sprint**

# Scrum

## Eventos principales

### **Retrospectiva de Sprint (Sprint Retrospective):**

- Es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo, y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.
- La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Reunión de Planificación de Sprint
- Se trata de una reunión restringida a un bloque de tiempo de tres horas para Sprints de un mes. Para Sprints más cortos se reserva un tiempo proporcionalmente menor.

# Scrum

## Eventos principales

### **Retrospectiva de Sprint (Sprint Retrospective). Su propósito es:**

- Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas
- Identificar y ordenar los elementos más importantes que fueron bien, y posibles mejoras
- Crear un plan para implementar las mejoras para la forma en la que el Equipo Scrum desempeña su trabajo.

**El Scrum Master alienta al equipo para que mejore, dentro del marco de proceso Scrum, su proceso de desarrollo y sus prácticas para hacerlos más efectivos y amenos para el siguiente Sprint**



# Scrum

## Artefactos

### **Pila de Producto (Product Backlog):**

- La Pila de Producto es una lista ordenada de todo lo que podría ser necesario en el producto
- Es la única fuente de requerimientos para cualquier cambio a realizarse en el producto.
- El Dueño de Producto (Product Owner) es el responsable de la Pila de Producto, incluyendo su contenido, disponibilidad y ordenamiento.
- Una Pila de Producto nunca está completa.
- El desarrollo más temprano de la misma sólo prepara los requerimientos conocidos inicialmente y mejor entendidos.
- La Pila de Producto es dinámica, cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil.
- La Pila de Producto es ordenada normalmente por valor, riesgo, prioridad y necesidad.

# Scrum

## Artefactos

### **Pila de Sprint (Sprint Backlog):**

- La Pila de Sprint es el conjunto de elementos de la Pila de Producto seleccionados para el Sprint mas un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint
- La Pila de Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad
- La Pila de Sprint define el trabajo que el Equipo de Desarrollo llevará a cabo para convertir elementos de la Pila de Producto en un Incremento “Hecho”
- La Pila de Sprint hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para satisfacer el Objetivo del Sprint
- La Pila de Sprint es un plan con un nivel de detalle suficiente como para que los cambios en el progreso puedan ser entendidos en el Scrum Diario