

# Scrum



UNIVERSIDAD DE  
**MURCIA**

# Una metodología para la gestión de proyectos



# Disciplina vs. Agilidad

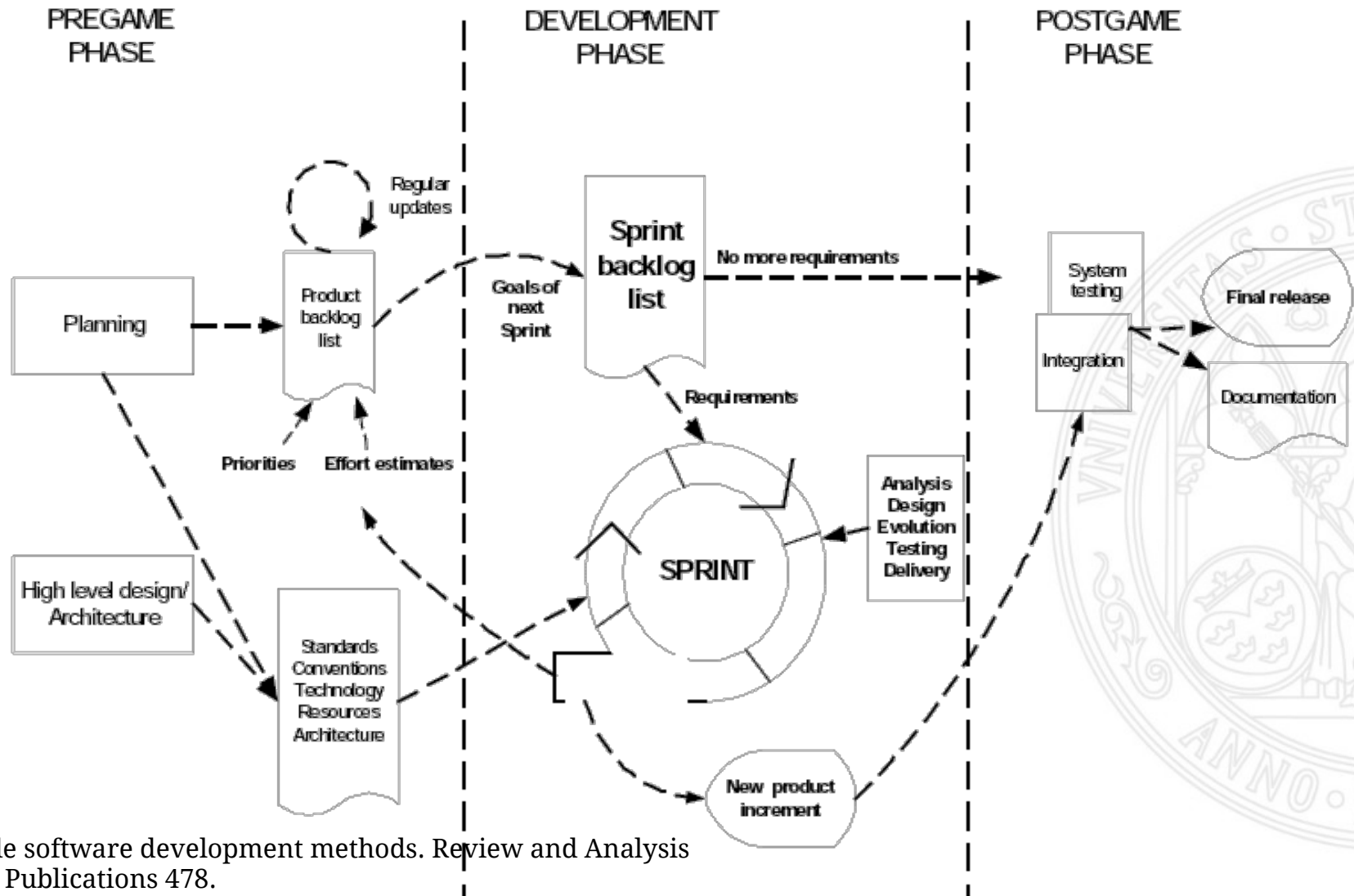
- Individuos e interacciones frente a procesos y herramientas
- Software trabajando frente a documentación
- Colaboración con los clientes frente a negociación
- Respuesta a los cambios frente a seguir los planes

# Disciplina vs. Agilidad

- Utilizan reglas ligeras pero suficientes para el comportamiento del proyecto.
- Utilizan reglas orientadas a la comunicación.
- Utilizan el paradigma del desarrollo iterativo.
- Procuran la satisfacción del cliente en el momento de la entrega más que a la iniciación del proyecto.

- Aproximación empírica que aplica ideas de la teoría de control de procesos industriales.
- No especifica técnicas de desarrollo del software para la fase de implementación.
- Se concentra en como funcionan los miembros del equipo para conseguir un sistema flexible en un entorno cambiante.
- Conlleva actividades de gestión frecuentes para identificar deficiencias e impedimentos en el desarrollo.

# Fases



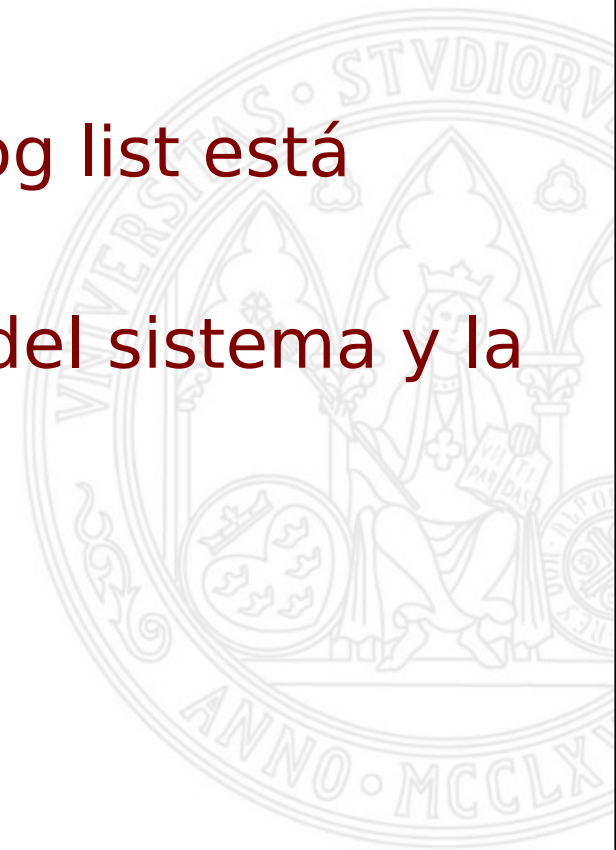
Fuente: Agile software development methods. Review and Analysis  
VTT Publications 478.  
Pekka Abrahamson, Outi Salo, Jussi Ronkainen y Juani Warsta

- Fase Pre-game.
  - **Planificación.**
    - Incluye la definición del sistema a construir.
    - Product backlog list contiene todos los requisitos conocidos.
    - Priorización de requisitos.
    - Equipo de proyecto, herramientas, riesgos, etc.
  - **Arquitectura.**
    - Diseño del sistema a alto nivel.
    - Planificaciones para los contenidos de las releases.

- Fase de desarrollo.
  - Se observan y controlan todas las variables técnicas y de entorno que afectan al desarrollo.
    - Marcos de tiempo, calidad, requisitos, recursos, tecnologías y herramientas.
  - Sprints son ciclos iterativos que producen incrementos en el producto.
  - Cada Sprint incluye requisitos, análisis, diseño, evolución, test y entrega.
  - Arquitectura y diseño evolucionan durante los sprint.



- Fase Post-game.
  - Se cierra la versión.
  - Comienza cuando la product backlog list está vacía.
  - Se produce la integración, los test del sistema y la documentación.



- Scrum Master
  - Vela por la “pureza” de Scrum. Elimina cualquier impedimento o dificultad. Forma parte del equipo de desarrollo.
- Product Owner
  - Oficialmente responsable para el proyecto, de la gestión, el control y hacer visible la Product Backlog list.
  - Transformación de la Product backlog list en características a desarrollar.

- Scrum Team
  - Autoorganizado para conseguir los objetivos de cada sprint.
  - Estimación de esfuerzos, crear el sprint backlog, sugerir impedimentos a eliminar, etc.
- Customer
  - Tareas relacionadas con la Product Backlog list.
- Management
  - Toma la decisión final sobre, métodos, procedimientos, convenciones, etc. a utilizar.

No son prácticas de desarrollo de software.

Son prácticas de gestión y herramientas.

- Product Backlog.
  - Lista de necesidades para el producto final.
  - Constantemente actualizada y priorizada.
- Estimación de esfuerzo.
  - Estimación para los ítems de la Product Backlog.
- Sprint.
  - Procedimiento de adaptación a los cambios ambientales.

- Reunión de Planificación de sprint (Sprint Planning meeting)
  - Fase 1: Se decide los objetivos y la funcionalidad del siguiente Sprint.
  - Fase 2: Cómo se incrementará y se implementará el producto.
- Pila de Sprint (Sprint Backlog)
  - Lista de items a implementar en un Sprint.
  - Estable durante el Sprint.

- Scrum diario (Daily Scrum meeting)
  - Qué se ha hecho desde la última reunión.
  - Qué va a hacerse hasta la siguiente.
- Demo de sprint (Sprint Review meeting)
  - El último día, se presentan los resultados del Sprint.
  - Se toman decisiones acerca del siguiente Sprint o la finalización de la release.

- Proyectos ya iniciados.
  - Comenzar por prácticas concretas.
  - Intentar ofrecer un producto válido lo más rápidamente posible.
- Proyectos nuevos.
  - Trabajar durante días para definir una Product Backlog list.
  - Construir y diseñar un sistema inicial.
  - Añadir funcionalidad.
- Útil para equipos pequeños y autoorganizados.