

# Métodos y Modelos en Ingeniería de Software I (95.20)

Conclusiones y cierre del curso

Segundo cuatrimestre 2023  
(16a.edición)





Este curso es una **introducción** a la **ingeniería de software moderna**, con una fuerte orientación al **trabajo en equipo** y con un adecuado **balance** entre contenidos **teóricos** y actividades **prácticas**.

# Ingeniería de Software

Una ingeniería de cosas abstractas.

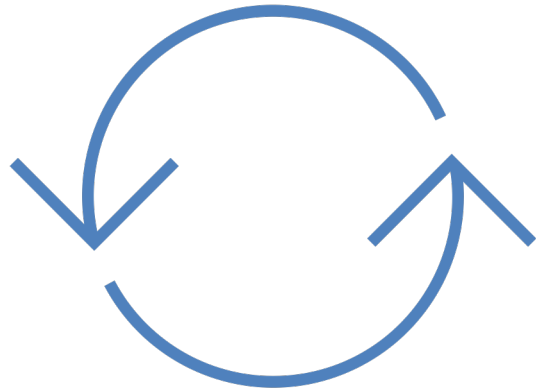


“Software design is an exercise  
in human relationships”

Kent Beck, creador de *Extreme Programming*

# Desarrollo de software

Un sistema sociotécnico complejo



- Muchos componentes
- Muchas personas, muchas interacciones
- Es un esfuerzo **de equipo**
- Todos debemos comprometernos a dar lo más que podamos para aportar al buen funcionamiento del sistema

# Desarrollo de Software

¡No hay recetas!

“Transforming design into code is not a defined, automatable process; it is an empirical one requiring ‘thought.’”

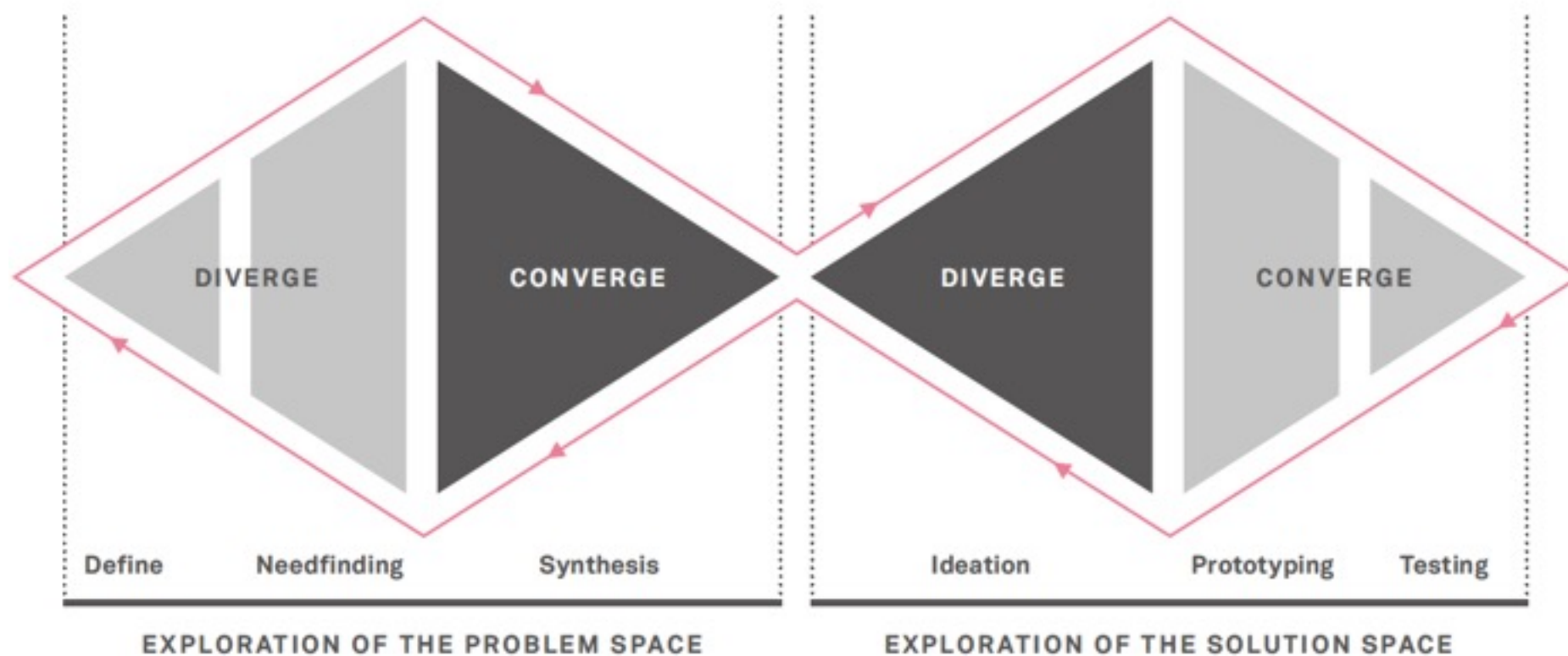
Jim Highsmith

*Wild West to Agile: Adventures in Software  
Development Evolution and Revolution*

- Si bien hay un cuerpo de conocimiento, los abordajes al desarrollo de software son **heurísticos**
- No hay teoremas, no hay  $A \Rightarrow B$ , no hay algoritmos, **no hay fórmulas mágicas**
- Hay métodos, técnicas, herramientas, etc. que se pueden aplicar con **criterio dependiendo del contexto**

# Desarrollo de Software

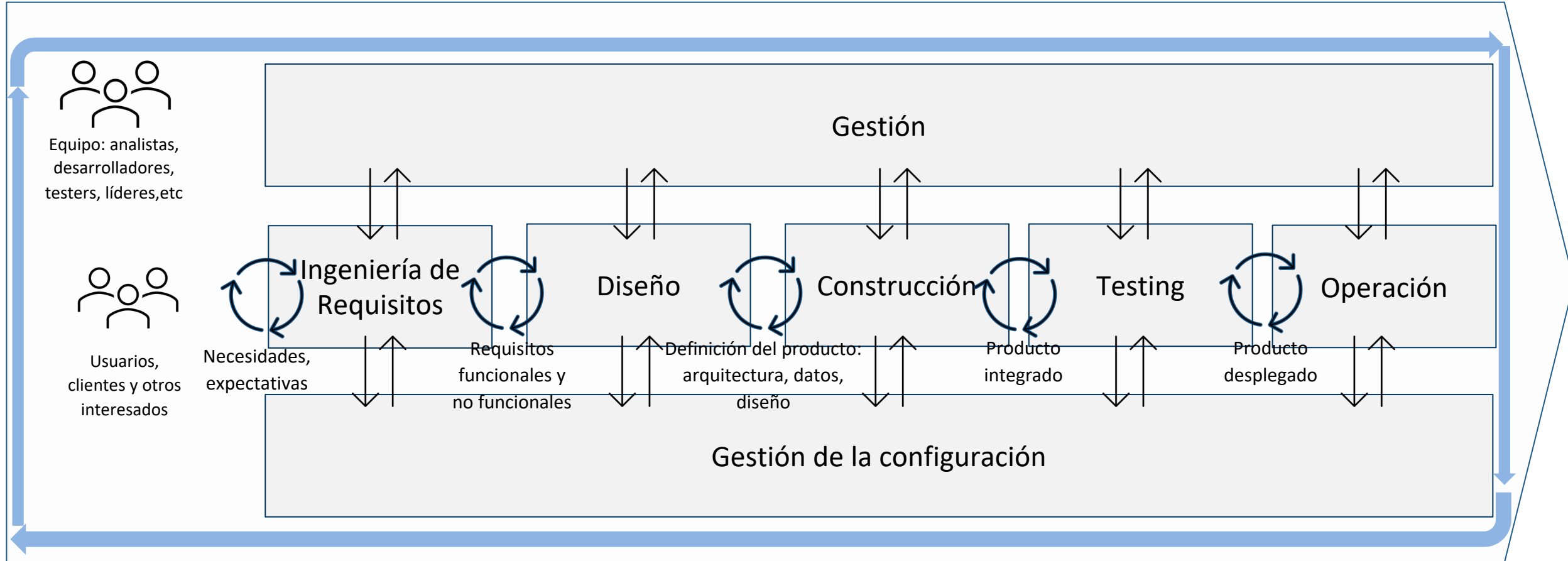
¡Siempre tendremos que lidiar con la **ambigüedad**!



Adaptado del tutorial *DT4RE: Design Thinking for RE*, por Jennifer Hehn, Falk Uebernickel, Daniel Méndez



# Un proceso iterativo e incremental, integrado por múltiples disciplinas





# Contenidos

## Métodos y Modelos en Ingeniería de Software I

Concepción

Creación

Liberación

Operación

### Requisitos

Alcance y objetivos: Visión, Impact Mapping

Descubrimiento: entrevistas, talleres de requisitos, focus groups, historias de dominio, etc.

Especificación & análisis: historias de usuario, casos de uso, modelado de dominio, reglas de negocio, prototipado

### Diseño

Conceptos clásicos, vistas 4+1, arquitectura y RNF

Arquitectura, estilos arquitectónicos

Experiencia de usuario

Bases de datos relacionales

### Testing, revisiones

Revisiones de pares, tipos de testing, testing automatizado

### Procesos, Gestión & Peopleware

Story mapping, story points, product backlog, sprint backlog, release plan, reporte de avance

Complejidad del desarrollo de software, modelos de ciclos de vida, desarrollo iterativo e incremental, agilidad, proyectos, aspectos humanos

Configuraciones, cambios, despliegue automático, camino a producción, DevOps

# Conclusiones

Qué aprendimos: distintas técnicas para distintas situaciones



Herramientas sencillas.

Proceso sencillo.

Una persona para construirla.

Herramientas potentes.  
Proceso bien definido.  
Un equipo de personas para  
construirla.  
**Modelado imprescindible.**



# Conclusiones

Entender el problema y definir la solución



- Las **organizaciones de desarrollo de software exitosas** son aquellas que consistentemente:
  - **Entregan software de calidad** que satisface las necesidades de sus usuarios;
  - **Cumplen con los compromisos** acordados con sus clientes;
  - Hacen **uso eficaz y eficiente** de sus **recursos**.

# Métodos y Modelos en Ingeniería de Software I / Análisis de la Información

## Cómo seguimos

- Cierre de notas: semana del 18/12
- Retrospectiva (próximamente)
- Anotarse en la fecha de final del 18/12.
- ¡No desinstalar Slack!
- ¡No desinstalar Classroom!

# Equipo docente

## Responsable:

- Sergio Villagra

## Auxiliares:

- Guido De Benedetti
- Tomás Bruno
- Leonardo Fernández

## Colaboradores:

- Francisco Aller Atucha
- Franco Mazzoni
- Nicolás Outeda
- Jonathan Moguilevsky

