



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita



Ingeniería de software III

Unidad 3 – Diseño en las metodologías ágiles

Definiciones



- ¿Qué es el Software?
- ¿Qué es la ingeniería del Software?
- ¿Qué es un Proceso de Desarrollo de Software?

- Definición

"Programas de cómputo y documentación asociada."

"Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación (extraído del estándar 729 del IEEE)."

"Término genérico que se refiere a una colección de datos e instrucciones de computadora qué le dicen a la computadora cómo hacer su trabajo. En contraste, el hardware es el que realiza el trabajo"

- Definición

“Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software”

(IEEE 1993)

Sistemático -> normas, procedimientos

Disciplinado -> Orden y subordinación entre los miembros

Cuantificable -> Medible

Proceso de software

- Definiciones

“Conjunto de actividades que conducen a la elaboración de un producto de software”

- Requiere la definición de roles
- Requiere la especificación de las actividades

Ingeniería de software



- En términos de proceso
 - ¿Cómo piensan ustedes que se debe desarrollar software de manera correcta?
 - ¿Cuáles son los pasos a seguir?
 - ¿Qué actividades lo componen?
 - ¿Cómo harían el planning?
 - ¿Cuáles son los roles que identifican?

Ingeniería de software



- ¿Cómo desarrollan el software que tienen que presentar en un trabajo práctico para la facultad?

Modelo de Proceso de Software



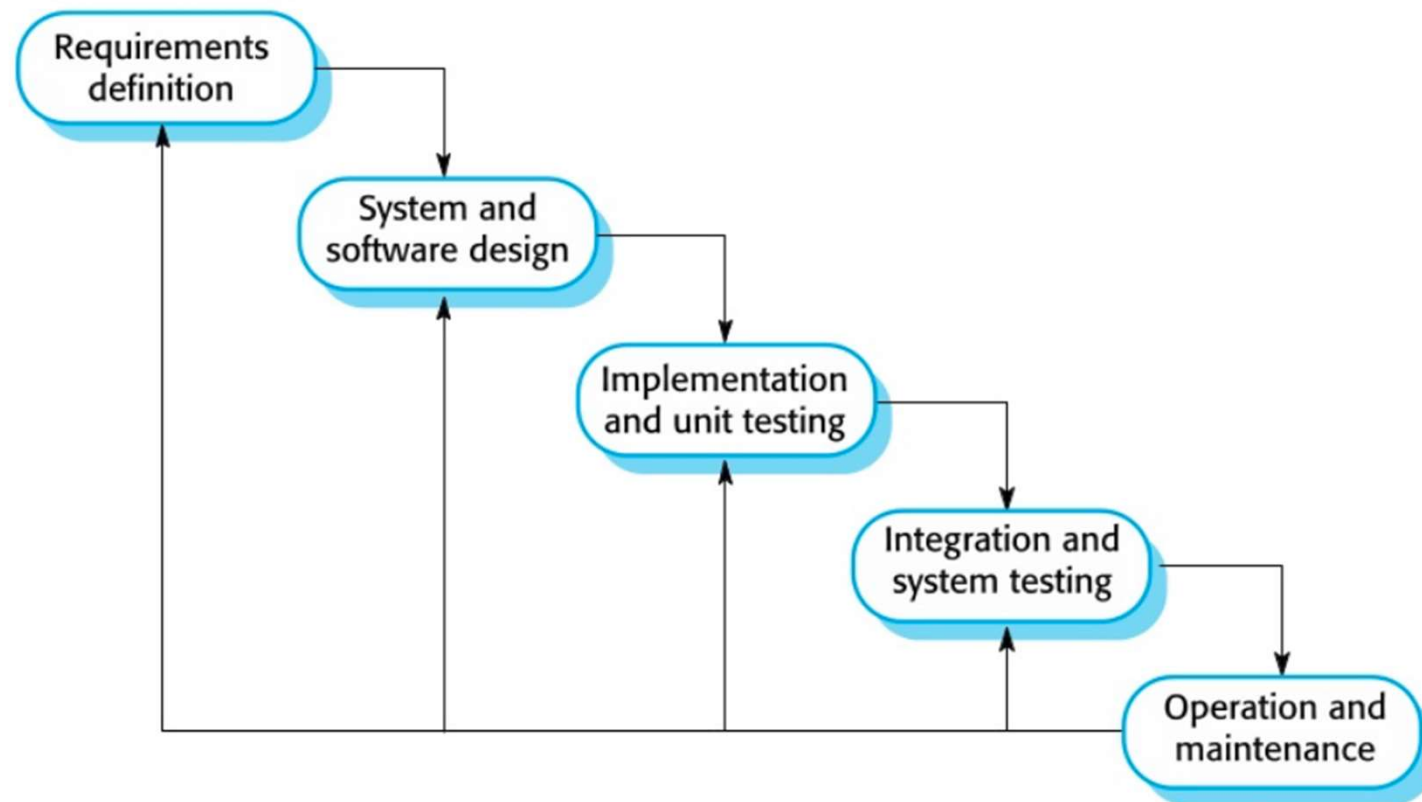
- Definiciones

“Representación simplificada del proceso de software que pueden utilizarse como base para crear procesos de software específicos ”

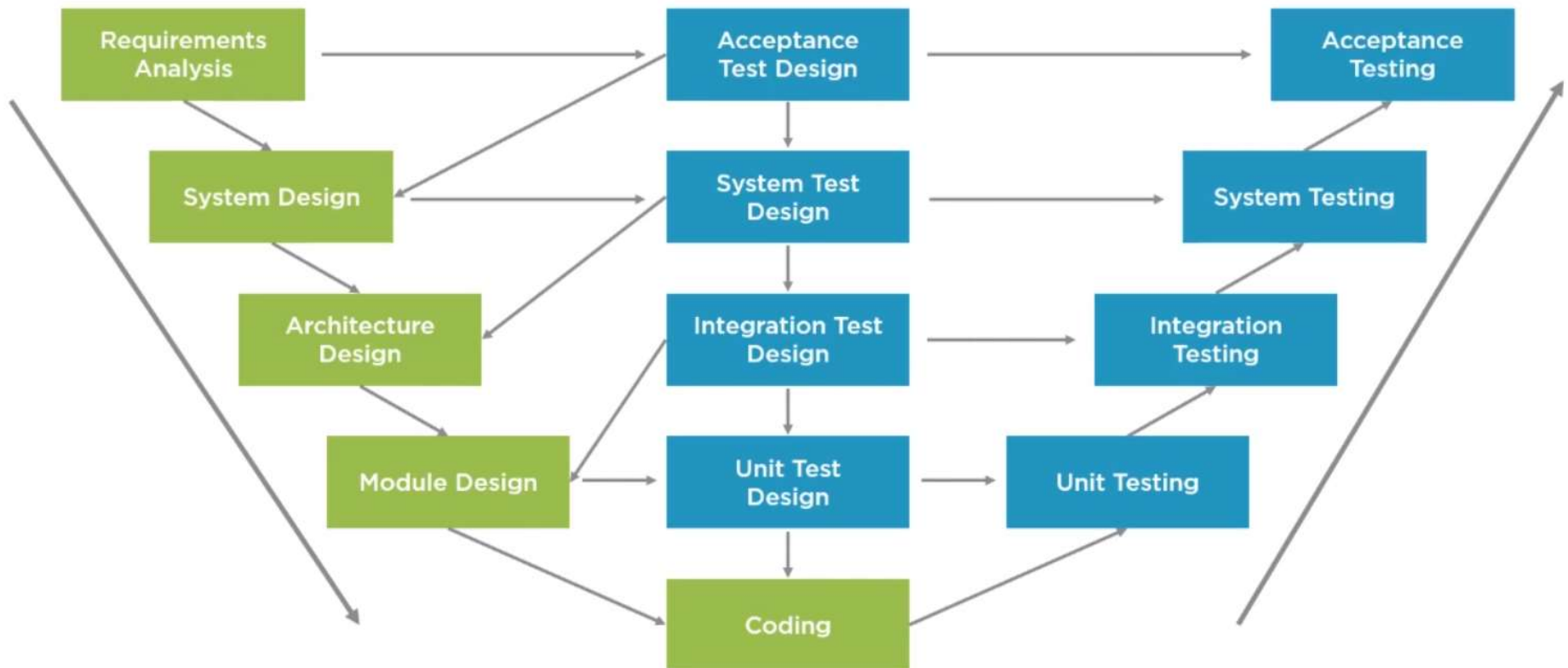
Modelo en cascada



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita



Paper – Winston Royce



Costo de los cambios



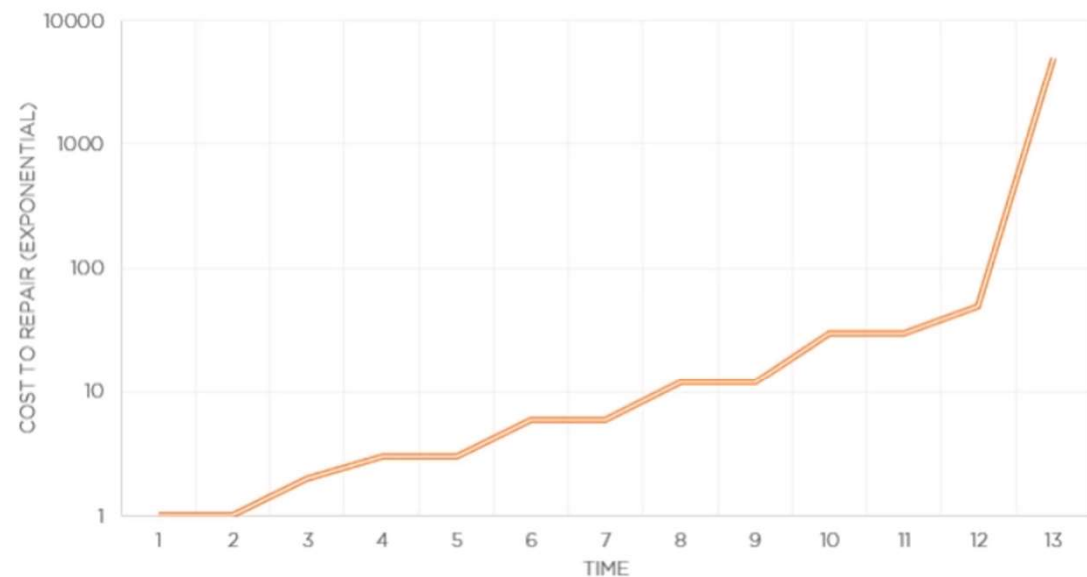
UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

The price of fixing a
bug increases over its
lifetime

From requirements

To development

To Production



Modelos secuenciales

- Problemas
 - Inflexible
 - Difícil responder al cambio de los requerimientos
 - Poco probable que se construya lo que el cliente realmente necesita
- Usado cuando
 - los requerimientos se entienden por completo
 - el software no cambiará (sistema embebido)

Modelos secuenciales

- Problemas
 - Inflexible
 - Difícil responder al cambio de los requerimientos
 - Poco probable que se construya lo que el cliente realmente necesita
- Usado cuando
 - los requerimientos se entienden por completo



Modelos secuenciales



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

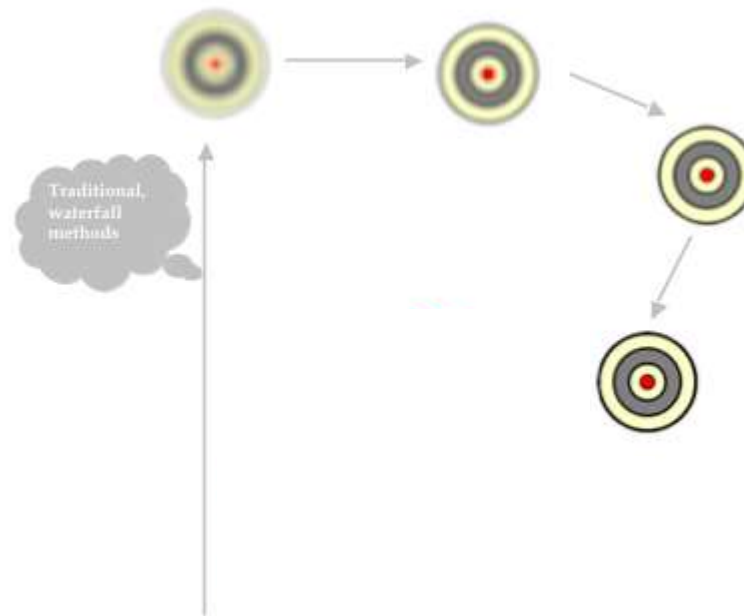
- Problemas
 - Inflexible





Modelos secuenciales

- Problema
 - Influye
 - Difícil
 - Poco conocido
- Usado (o no)
 - los clientes
 - el sistema



imientos
el cliente realmente

pleto
(ido)



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

Is There a Better Way?





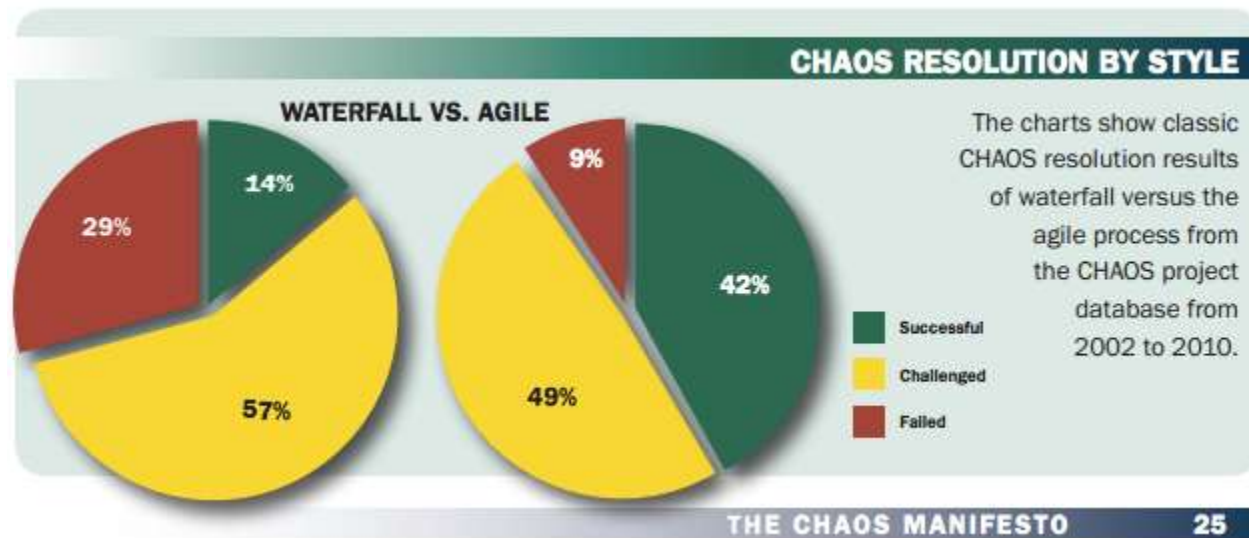
Desarrollo de software

- Historia del proceso de desarrollo del software
 - Early days
 - Mercury -> programa special
 - 68 -> Ingeniería de Software
 - 70's
 - Winston Royce -> waterfall ([paper](#))
 - “Appealing” para los managers
 - 85 -> estandar 2167 DoD
 - Mid/late 90's -> Agile
 - 1995 -> Jeff Sutherland, Mike Beedle, Martine Devos, Ken Schwaver -> Scrum
 - 1999 -> Kent beck -> XP
 - 1994 -> DoD reemplaza el estándar 2167 4984
 - Otros
 - 2001 -> [Agile manifesto](#)
 - 2009 -> [softwarecraftmanship manifesto](#)
 - 2009 -> [Congreso EEUU nueva ley 111-846 impone condiciones al DoD](#)

Agile



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

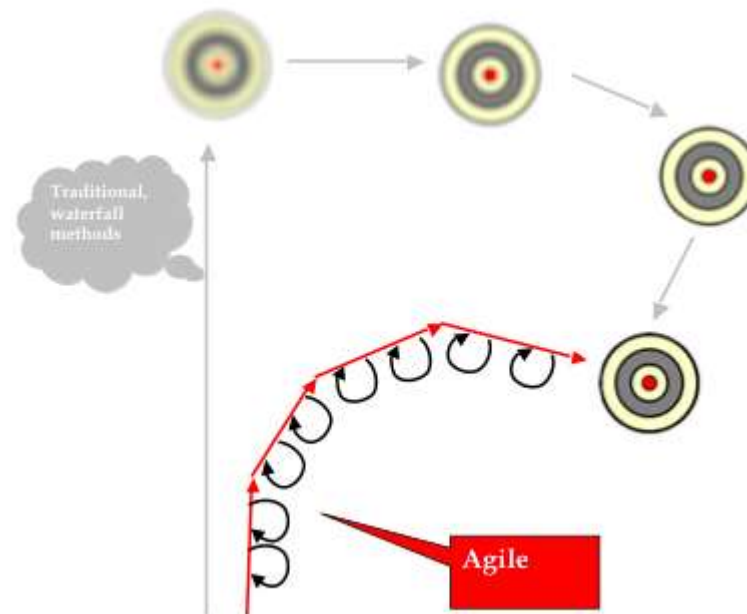


“El proceso ágil es el remedio universal para el fracaso en los proyectos de desarrollo de software. Las aplicaciones de software desarrolladas a través del proceso ágil tienen tres veces la tasa de éxito del método en cascada tradicional y un porcentaje mucho menor de demoras y sobrecostos. [...] El software debería ser construido en pequeños pasos iterativos, con equipos pequeños y enfocados.” [Standish Group] 2012

Agile



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita



“El proceso ágil es el remedio universal para el fracaso en los proyectos de desarrollo de software. Las aplicaciones de software desarrolladas a través del proceso ágil tienen tres veces la tasa de éxito del método en cascada tradicional y un porcentaje mucho menor de demoras y sobrecostos. [...] El software debería ser construido en pequeños pasos iterativos, con equipos pequeños y enfocados.” [Standish Group] 2012

Plan guiado y el desarrollo ágil

✧ Desarrollo guiado por plan

- Etapas bien definidas
- Entregables por etapas bien definidos
- Entradas de una etapa es la salida de la etapa anterior
- Las iteraciones se producen dentro de las actividades
- Inflexible

✧ Desarrollo Ágil

- Especificación, diseño, implementación y pruebas son intercalados
- Salidas del proceso decididas mediante negociación durante el proceso de desarrollo de software (feedback)
- Iterativo e incremental
- Responde al cambio

Waterfall vs Agile



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

Traditional Waterfall



Agile Development

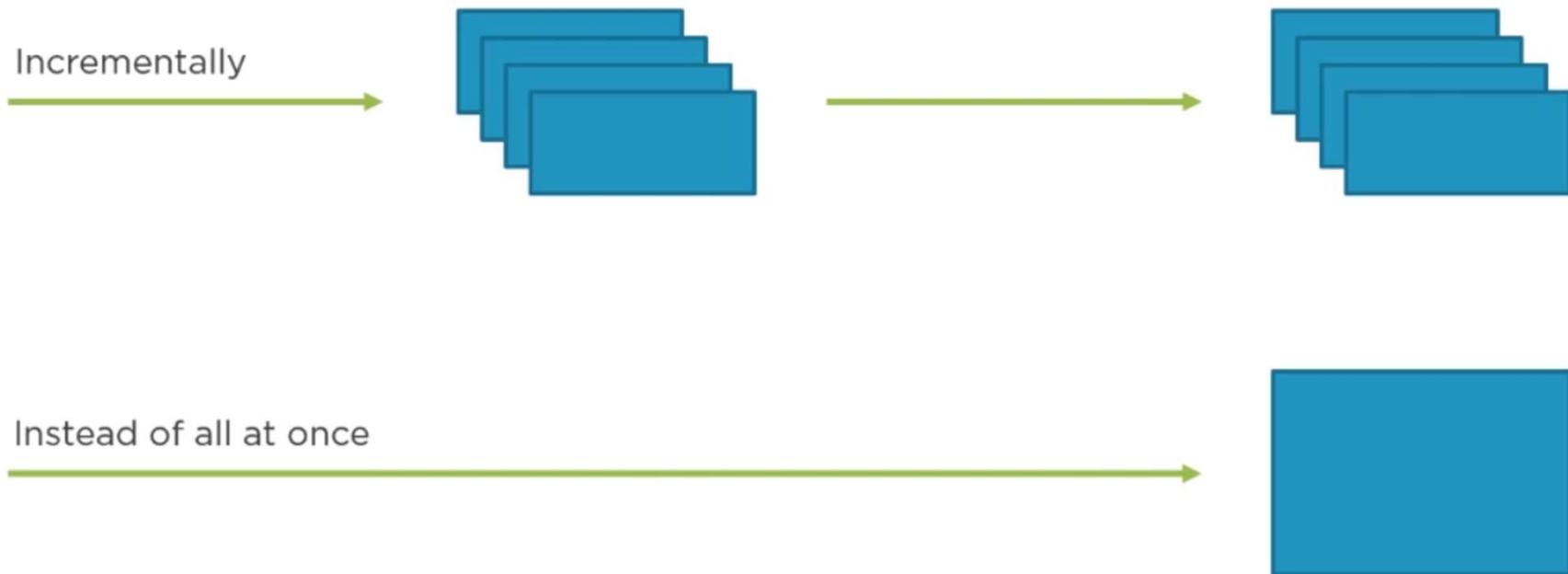


Project Timeline

Waterfall vs Agile



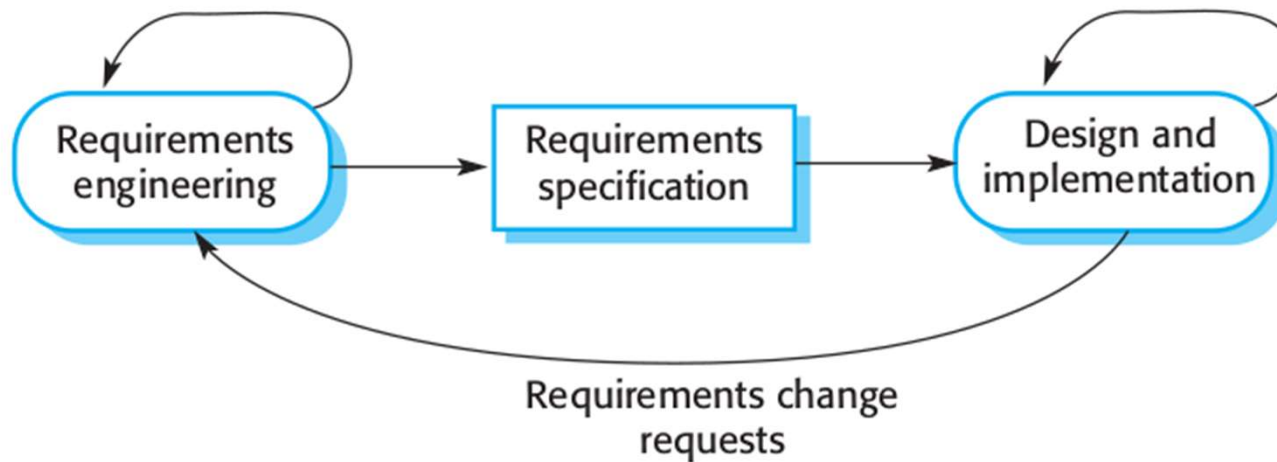
UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita



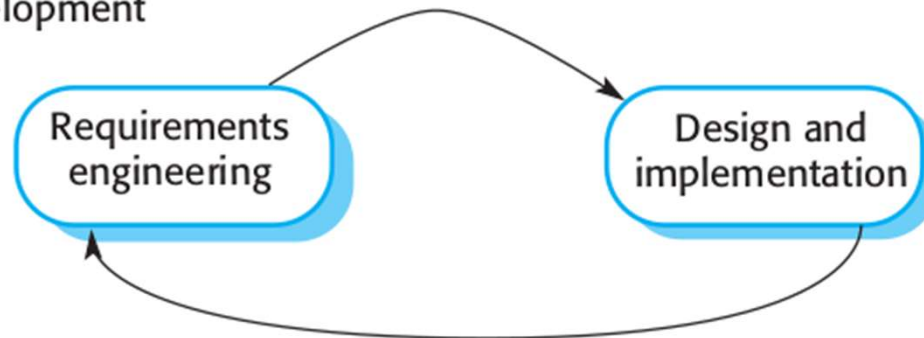
Guiado por plan vs ágil



Plan-based development



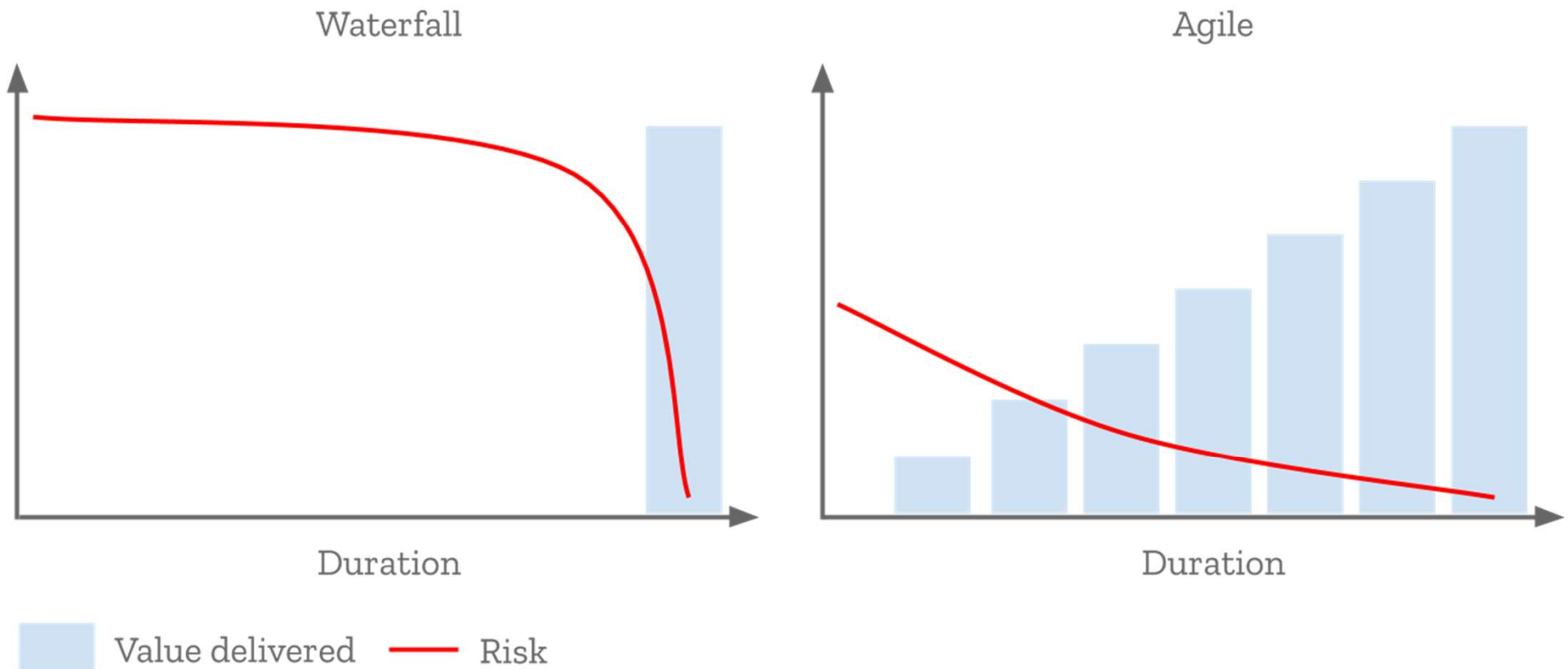
Agile development



Cascada vs Agile



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita



Manifiesto Ágil

- Estamos descubriendo mejores maneras de desarrollar software y al hacerlo ayudando a otros a hacerlo. A través de este trabajo hemos valorado:
 - Individuos e interacciones sobre procesos y herramientas
 - Software funcionando sobre la documentación comprensible
 - Colaboración con el cliente sobre negociación de contratos
 - Responder al cambio sobre seguir el plan
- Es decir, mientras hay valor en los elementos de la derecha, valoramos mas los elementos de la izquierda.

Manifiesto Ágil



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

Individuals and interactions over
processes and tools

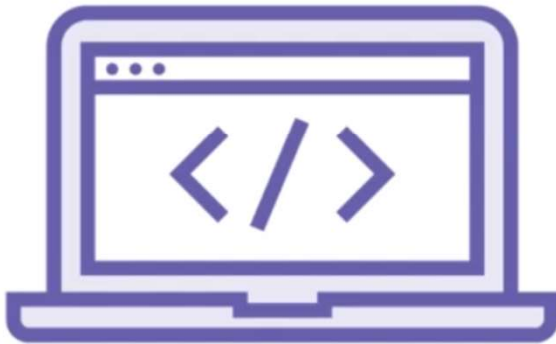
Working software over
comprehensive documentation

Customer collaboration over
contract negotiation

Responding to change over
following a plan

Principios manifiesto Ágil

- 12 principios separados en 3 grupos
 - Entrega del software en intervalos regulares y frecuentes
 - Comunicación en el equipo
 - Excelencia en el diseño



The highest priority is to satisfy the customer through early and continuous delivery of valuable software

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale

Working software is the primary measure of progress

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely

Comunicación en el equipo



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

Business people and developers must work together daily throughout the project

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation

The best architectures, requirements, and designs emerge from self-organizing teams

Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done

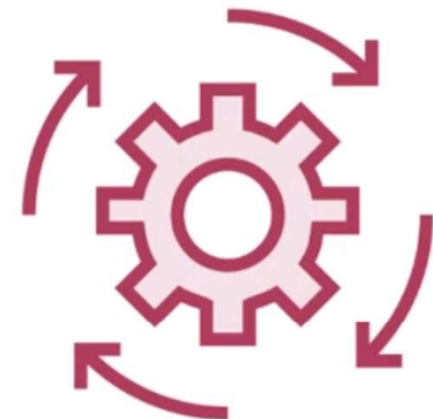
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly



Continuous attention
to technical excellence
and good design
enhances agility



Simplicity, the art of
maximizing the amount
of work not done,
is essential



Agile processes
harness change for
the customer's
competitive advantage

Roles en los teams ágiles



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

Agile teams are software development teams first and members of a department second

A team should have a product or domain expert

A team has members with cross functional skills

A team should have some form of leadership role

A team can benefit from an agile coach or mentor



Agile means 'No Commitment'

Agile development is not predictable

Agile is a silver bullet

There's only one way to do Agile

Agile doesn't need up-front design

Being agile is pain-free

We're doing scrum, so we don't need to pair program, refactor or do TDD

Errores que se comenten en los teams agile nuevos



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

Poor testing

Ignoring customer
feedback

Lack of team
empowerment

Lack of
retrospective and
demo meetings

No plan to
address employee
resistance



Early return on investment

Feedback from real customers

Build the right products

Continually deliver better
quality



Difficult to assess the effort required at the beginning of the software development life cycle

Can be very demanding on the users time

Potential for scope creep

Harder for new starters to integrate in the team

Costs can increase as testers are required all the time instead of at the end

Agile can be intense for the team

Problemas con métodos ágiles

- ✧ Puede ser difícil mantener el interés de los clientes que están involucrados en el proceso.
- ✧ Los miembros del equipo pueden ser inadecuados para la intensa participación que caracteriza a los métodos ágiles.
- ✧ Priorizar cambios puede ser difícil donde hay múltiples partes interesadas.
- ✧ Mantener simplicidad requiere un trabajo extra
- ✧ Los contratos pueden ser un problema, como con otros enfoques para desarrollo iterativo.

Problemas con métodos ágiles

- ✧ Puede ser difícil mantener el interés de los clientes que están involucrados en el proceso.
- ✧ Los miembros del equipo pueden ser inadecuados para la intensa participación que caracteriza a los métodos ágiles.
- ✧ Priorizar cambios puede ser difícil donde hay múltiples partes interesadas.
- ✧ Mantener simplicidad requiere un trabajo extra
- ✧ Los contratos pueden ser un problema, como con otros enfoques para desarrollo iterativo.

Problemas con métodos ágiles



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

- ✧ Puede ser difícil mantener el interés de los clientes que están involucrados en el proceso.
- ✧ Los miembros del equipo pueden ser inadecuados para la intensa participación que caracteriza a los métodos ágiles.
- ✧ Priorizar cambios puede ser difícil donde hay múltiples partes interesadas.
- ✧ Mantener simplicidad requiere un trabajo extra



Principios de los Métodos Ágiles

Resumidos

Principio	Descripción	
Participación de los Clientes	Los clientes deberían participar activamente en todo el proceso de desarrollo. Su función es proporcionar y dar prioridad a los nuevos requisitos del sistema y evaluar las iteraciones de la sistema	
Entrega Incremental	El software se desarrolla en incrementos con el cliente especificando los requisitos que deben incluirse en cada incremento	
Las personas no procesos	Tienen que reconocerse y aprovecharse las habilidades del equipo de desarrollo. Debe permitirse a los miembros del equipo desarrollar sus propias formas de trabajar sin procesos establecidos.	
Aceptar el cambio	Esperara cambios en los requerimientos del sistema y diseñar el sistema para adaptarse a estos cambios	
Mantenga la simplicidad	Enfoquese en la simplicidad tanto del software siendo desarrollado como el proceso de software. Siempre que sea posible, trabajar en actividades para eliminar la complejidad del sistema	

Aplicabilidad del método ágil



- ✧ Desarrollo de productos, donde una compañía de software está desarrollando un producto de pequeño o mediano tamaño para la venta
- ✧ El desarrollo del sistema a medida dentro de una organización, donde hay un claro compromiso por parte del cliente para participar en el proceso de desarrollo y donde no hay muchas reglas y regulaciones externas que afectarán el software.
- ✧ Debido a su enfoque en pequeños equipos, bien integrados, hay problemas en la ampliación de los métodos ágiles a grande sistemas

Cuestiones técnicas, humanas y organizacionales



✧ La mayoría de los proyectos incluyen elementos del plan guiado y procesos ágil. La decisión sobre el equilibrio depende de:

- ¿Es importante contar con una especificación y diseño muy detallado antes de pasar a la implementación?
- Es una estrategia de entrega incremental, donde se entrega el software a los clientes y se obtiene una rápida retroalimentación de ellos, ¿realista?
- ¿Qué tan grande es el sistema que se está desarrollando?

Cuestiones técnicas, humanas y organizacionales



- ¿Qué tipo de sistema se está desarrollando?
 - Los enfoques del plan guiado pueden ser necesarios para los sistemas que requieren una gran cantidad de análisis antes de la aplicación
- ¿Cuál es la expectativa de vida del sistema?
 - Los sistemas de larga vida tal vez requieran mas documentación de diseño para comunicar las intenciones originales de los desarrolladores del sistema al equipo de soporte.
- ¿Qué tecnologías están disponibles para apoyar el desarrollo del sistema?
 - Los métodos ágiles se basan en buenas herramientas para realizar un seguimiento de la evolución de un diseño
- ¿Cómo se organiza el equipo de desarrollo?
 - Si el equipo de desarrollo se distribuye o si parte del desarrollo se subcontrata, entonces puede que tenga que desarrollar los documentos de diseño **para** comunicarse a través de los equipos de desarrollo.

Cuestiones técnicas, humanas y organizacionales



- ¿Hay cuestiones culturales o de organización que puedan afectar al desarrollo del sistema?
 - Las organizaciones tradicionales de ingeniería tienen una cultura basada en el plan de desarrollo, ya que esta es la norma en la ingeniería.
- ¿Qué tan buenos son los diseñadores y programadores del equipo de desarrollo?
 - A veces se argumenta que los métodos ágiles requieren niveles más altos de capacitación que los enfoques basados en planes, en el que los programadores simplemente traducen un diseño detallado en código.
- ¿El sistema está sujeto a regulación externa?
 - Si un sistema tiene que ser aprobado por un regulador externo (por ejemplo, que el FAA aprueba el software, es crítico para el funcionamiento de una aeronave) entonces usted probablemente tendrá que producir detallada documentación como parte de la justificación de la seguridad del sistema.

Evitar



UNIVERSIDAD
CATÓLICA DE CÓRDOBA
Universidad Jesuita

WAgile



