

Tema 1: Introducció al procés de desenvolupament del software

Anna Puig

Enginyeria Informàtica

Facultat de Matemàtiques i Informàtica,
Universitat de Barcelona

Curs 2019/2020

Temari

1 Introducció al procés de desenvolupament del software

2 Anàlisi de requisits i especificació

3 Disseny

4 Del disseny a la implementació

5 Ús de frameworks de testing

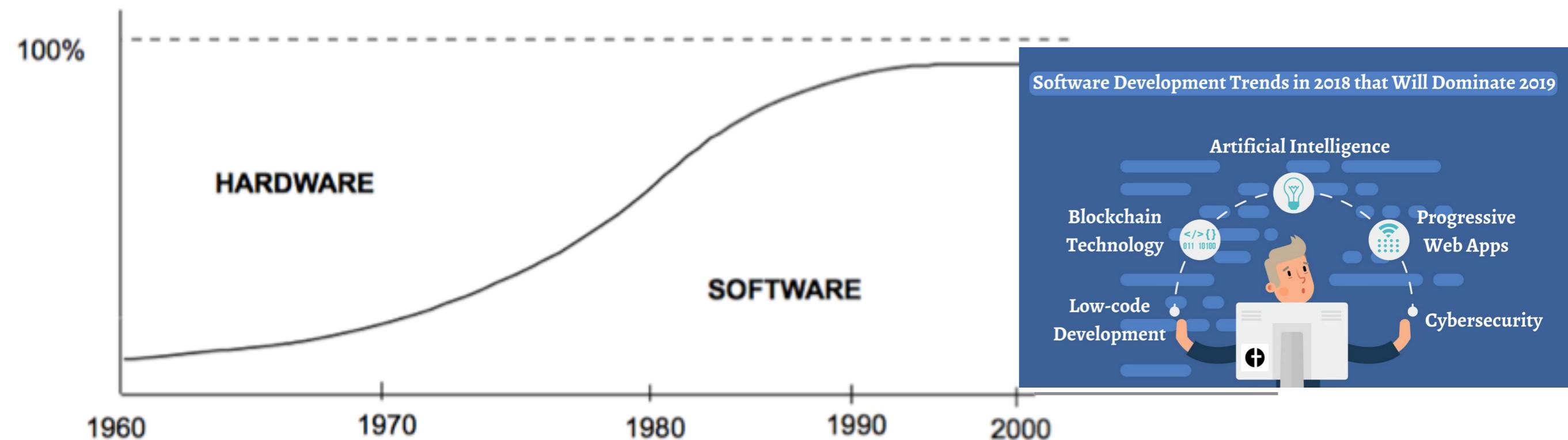
Temari

1	Introducció al procés de desenvolupament del software	
2	Anàlisi de requisits i especificació	1.1 Introducció
3	Disseny	1.2 Procés de desenvolupament de software
4	Del disseny a la implementació	1.3 Desenvolupament basat en tests
5	Ús de frameworks de testing	1.4 Exemple senzill

1.1. Introducció

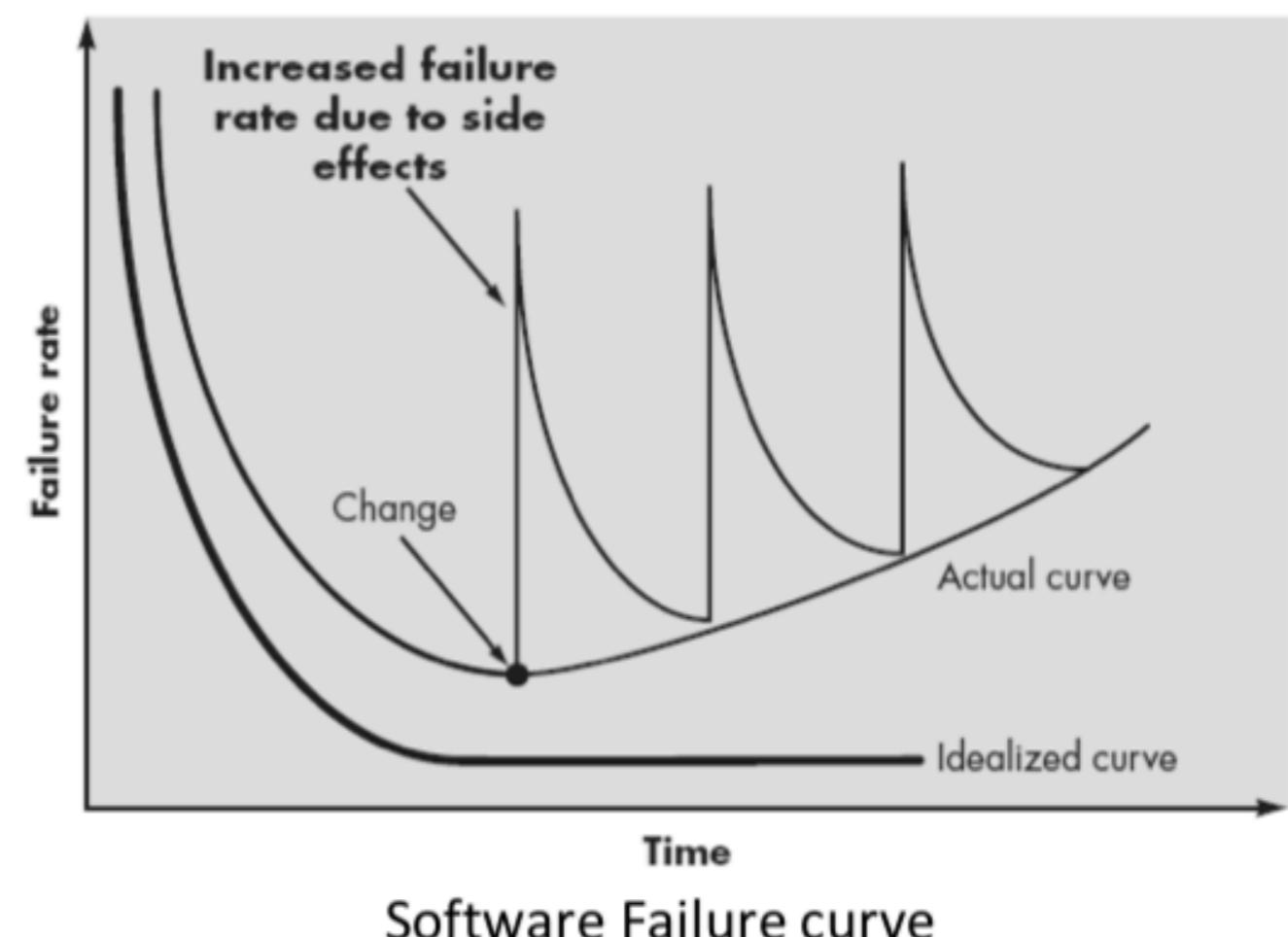
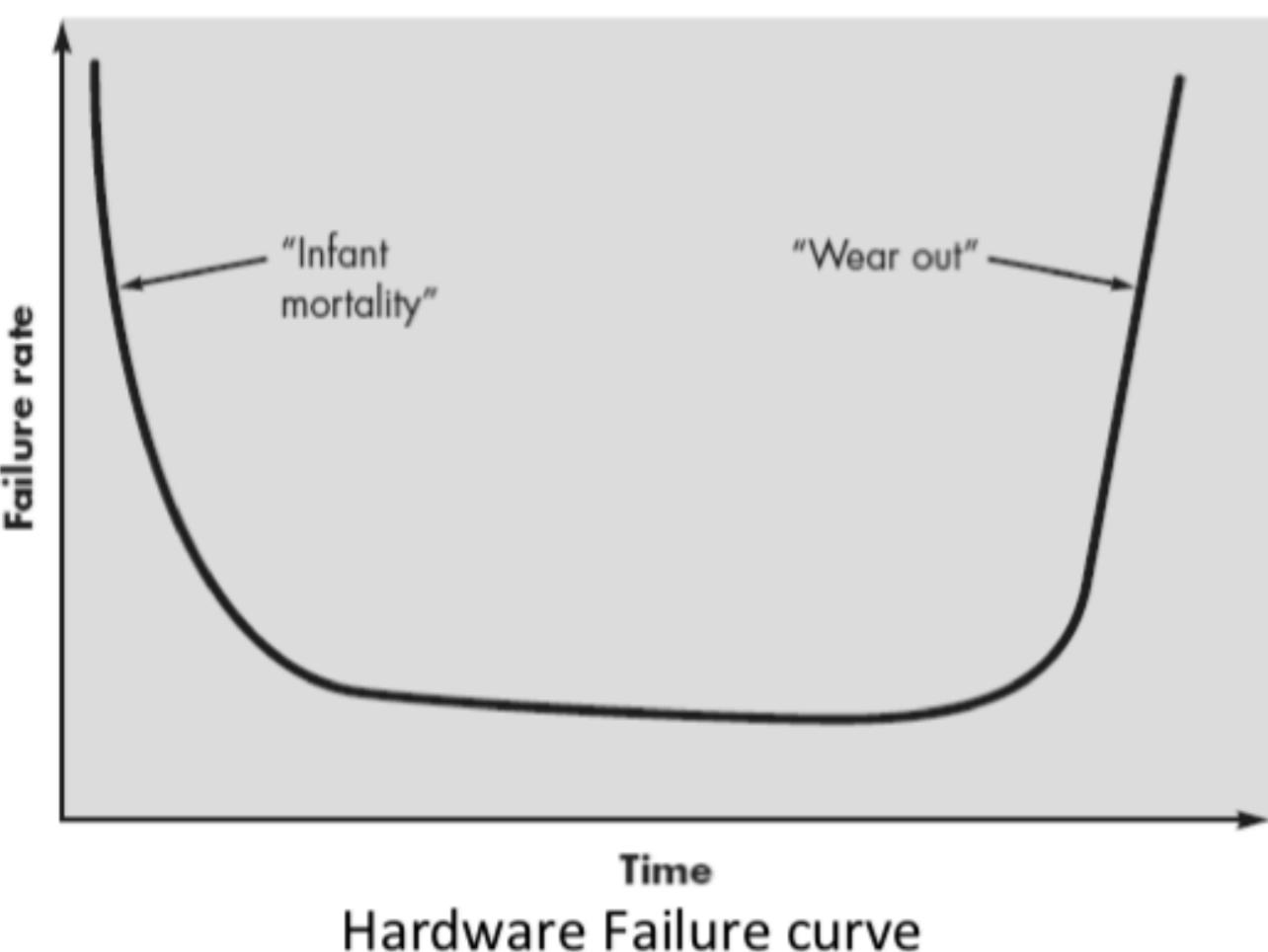
Evolució dels costos del hardware i del software

<http://robertvienneau.blogspot.com.es/2013/07/trends-in-hardware-and-software-costs.html>



Projectes més grans,
arquitectures més complexes

1.1. Introducció



1.1. Introducció

Software: “term used to describe a collection of computer programs, procedures and documentation that perform some task on a computer system” (wikipedia)

Característiques:

- Es desenvolupa, no es fabrica
- No s'espalla
- Sovint cal desenvolupar a mida
- Més difícil manteniment que el hardware



1.1. Introducció

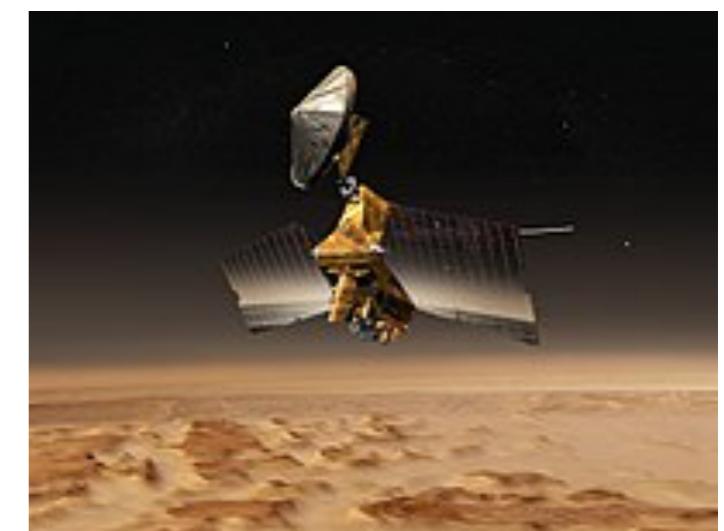
Pifies o fallides de software:

Therac 25 (85-87): 6 persones van morir sobre-exposades a radiació en tractament de càncer (dosis 100 vegades més gran de l'esperat). ***Contador d'un byte desbordat***



Taurus (93): projecte abandonat després de 10 anys invertits en digitalitzar la borsa de Londres. ***No es va preveure la complexitat del problema amb diferents tipus de clients i adopció de solucions no escalables.***

Mars Climate Orbiter (99): la sonda va impactar a la superfície de Mart en confondre milles i kilòmetres.
Problema de requeriments o de comunicació amb l'usuari



1.1. Introducció

Pifies o fallides de software:

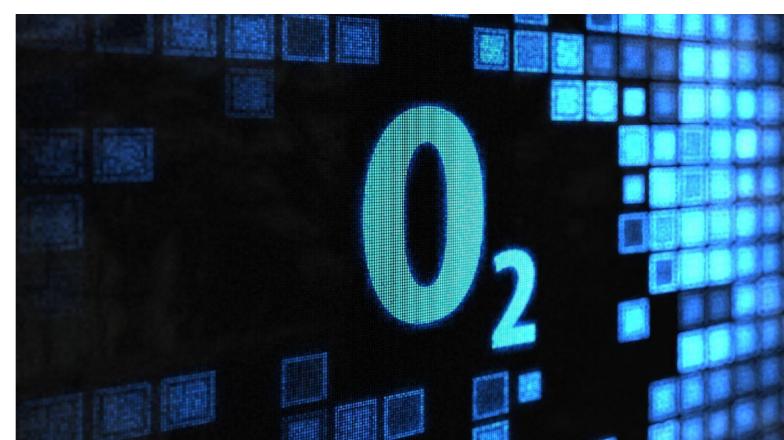
Instagram, whatsapp, facebook (2019):

Impossible carregar fotos o enviar missatge.

Error en una part del manteniment del software.



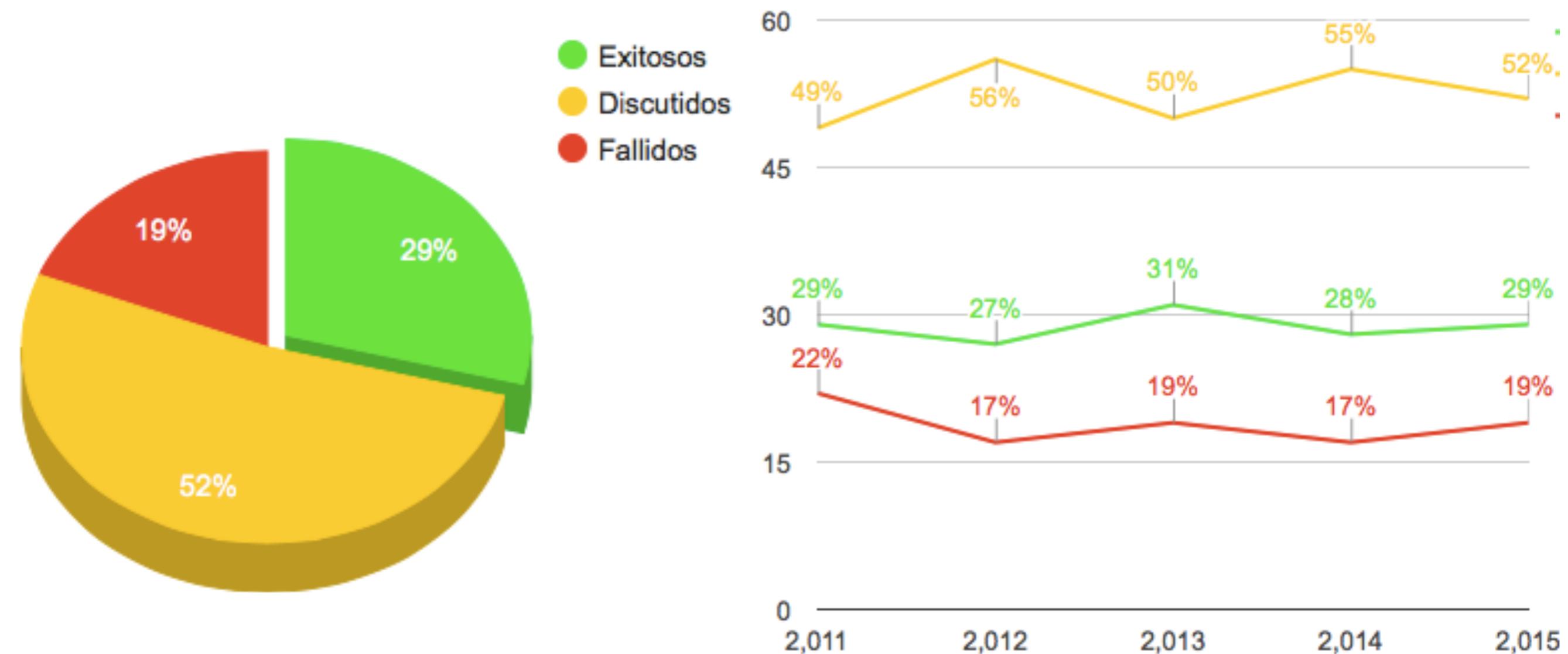
O2 (2018): millers d'usuaris no van poder accedir a les seves dades i es van produir errors en els horaris dels trens i autobusos de Londres. ***Un glitch de software va ser el desencadenant.***



<https://www.computerworld.com/article/3412197/top-software-failures-in-recent-history.html>

El software en xifres

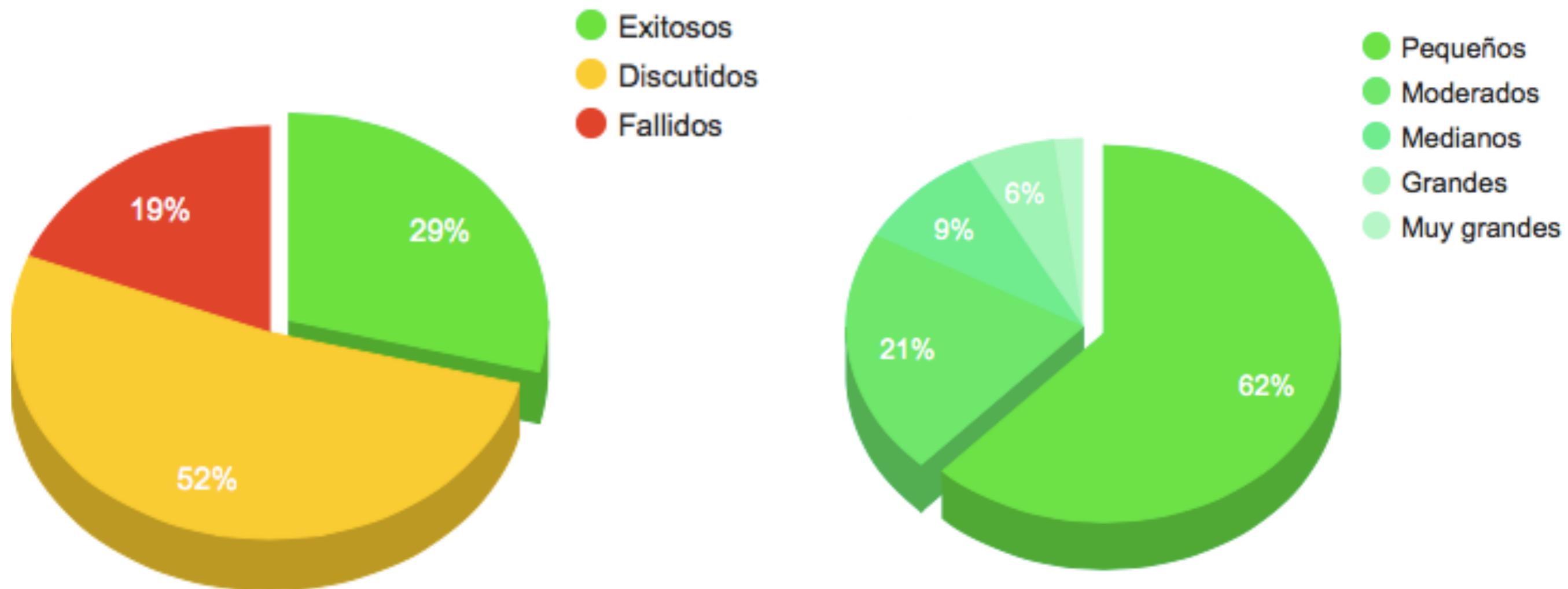
- Projecte Chaos (2015):



[Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch](#)

El software en xifres

- Projecte Chaos (2015):



[Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch](#)

El software com art

“Software is a great combination between artistry and engineering.”

Bill Gates

“A code is like love, it has created with clear intentions at the beginning, but it can get complicated.”

Gerry Geek

“When I’m working on a problem, I never think about beauty. I think only how to solve the problem. But when I have finished, if the solution is not beautiful, I know it is wrong.”

Freeman Dyson

El software com enginyeria

"I'm not a great programmer; I'm just a good programmer with great habits."

Kent Beck

"No one in the brief history of computing has ever written a piece of perfect software. It's unlikely that you'll be the first."

Andy Hunt

"Adding manpower to a late software project makes it later."

Fred Brooks

"9 women CANNOT make a baby in ONE MONTH."

Fred Brooks

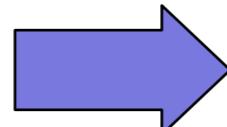
Enginyeria del software

Enginyeria del software: Establiment i ús de principis de l'enginyeria orientats a obtenir software (IEEE Standard 610.12).

Software:

- Econòmic
- Fiable
- Eficient
- Que satisfaci les necessitats dels usuaris

Com es fa?



- Llei del mínim esforç
 - Escollir la més senzilla de les opcions possibles
 - Reutilització (codi, artefactes software, coneixement, etc.)
- Usar estàndards (IEEE, OMG, etc.)
- Aprendre de l'experiència (bones pràctiques, patrons, etc.)

OMG
patrons
documentació
ISO
IEEE
bones pràctiques
models
escalà
complexitat
simplicitat
reutilització

1.2. Procés de desenvolupament de software

Metodologia o **procés** que permet:

- **Modelar** els problemes de forma fàcil i senzilla
- Ajudi a pensar el codi i a desenvolupar-lo
- Ajudi a documentar



“Software Architecture and UML” de Grady Booch
(Rational Software).

1.2. Procés de desenvolupament de software

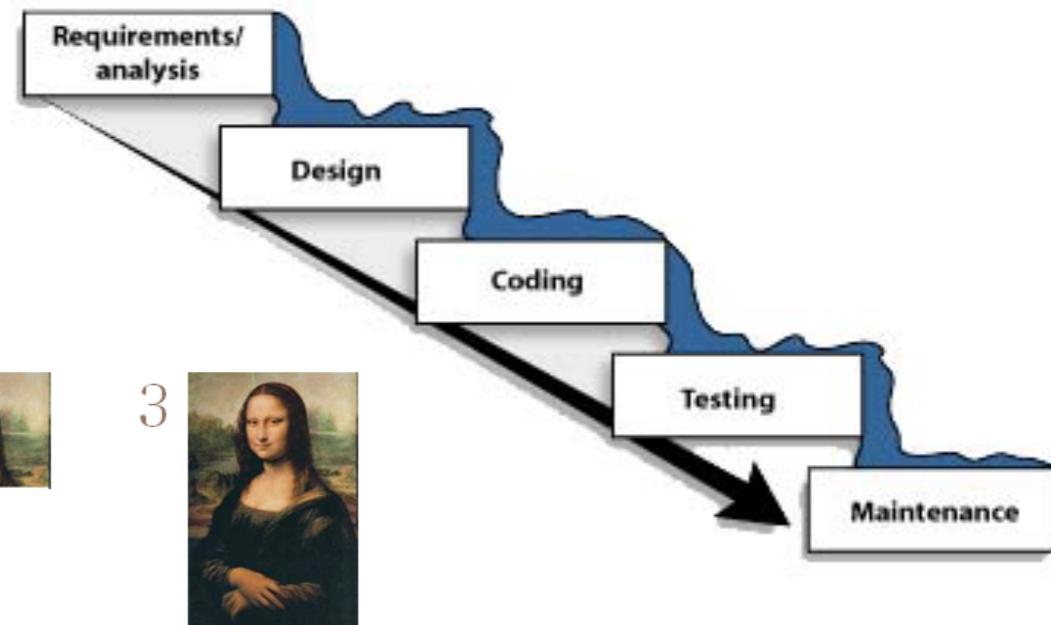


- Anàlisi del sistema
- Planificació del projecte
- Anàlisi de requisits del software
- Disseny del software
- Codificació
- Prova
- Correcció
- Adaptació
- Millora

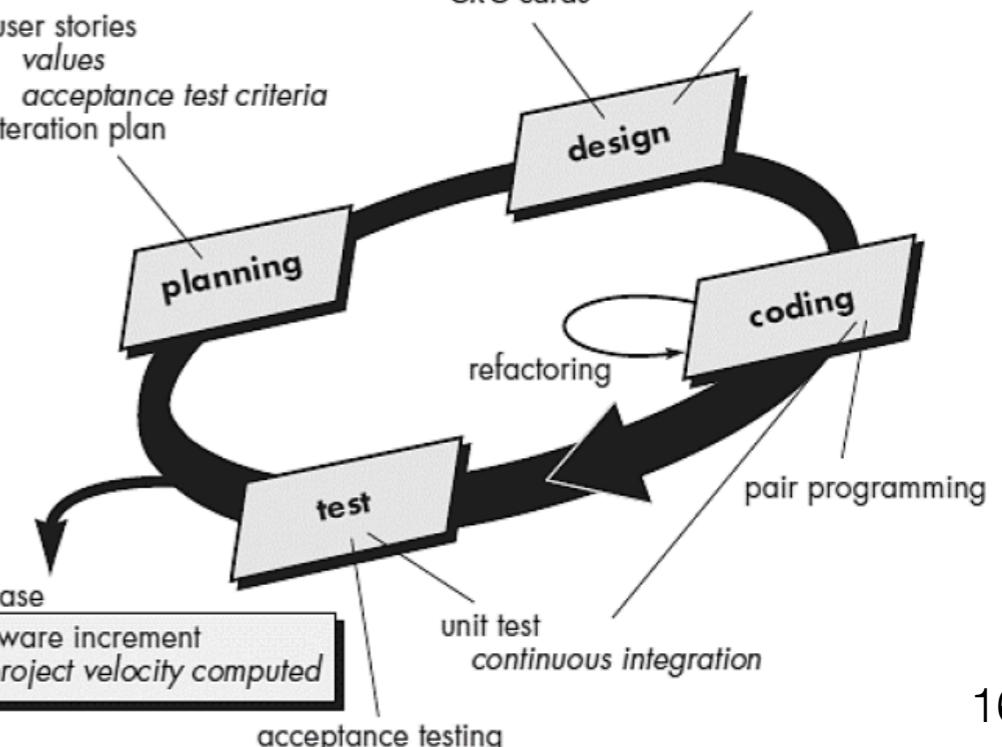
1.2. Procés de desenvolupament de software

- **Metodologies:**

- **Waterfall:** procés seqüencial (Winston Royce, 1970)



- Metodologia **incremental** i **iterativa** (90's)

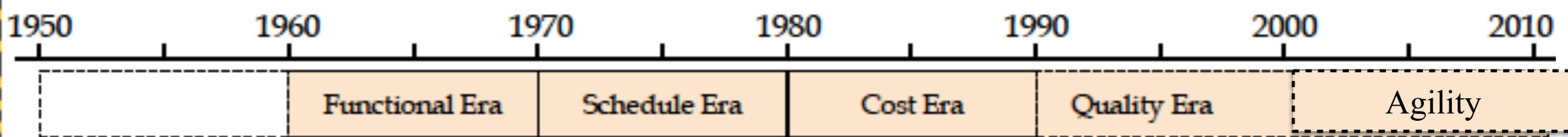


- Metodologia basada en **prototipus**

- Metodologies **àgils** (Agile Manifesto, 2001)

Evolució històrica

http://a-lifelong-tester.cocolog-nifty.com/Chronology/History_of_Software_TestingChronology_20120508_English.pdf



1960s: La era funcional

1970s: La era de fer els projectes a temps

- waterfall

1980s: La era dels costos

- model espiral

1990s: La era de la qualitat

- RUP (*Rational Unified Process*),
- Notacions per modelar (UML) i proposta de **patrons de disseny** (Gang of Four, 94))
- models **iteratius** i **incrementals**

2000s: Agilitat i valor

- Agil Manifesto (<http://agilemanifesto.org>, 2001) : **SCRUM**, Extreme-Programming. **Test-Driven Development** (Kent Beck), etc.

1.2. Procés de desenvolupament de software

Metodologies predictives:

- Des del començament es fixen els objectius i el pressupost i es proposen solucions estables, control de tot el procés, ben documentades

Metodologies Àgils (Agile Manifesto, 2001, Kent Beck at al.):

- La prioridad és satisfer al client mitjançant **Illiuraments de software ràpids** i contínus que li aportin valor.
- Donar la benvinguda als canvis.

Exemples: **SCRUM**, **Test Driven Development (TDD)**, Crystal Methodologies, Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD), Feature-Driven Development (FDD), Lean Development (LD), eXtrem Programming (XP).

1.2. Procés de desenvolupament de software

Cascada vs. Àgils (resultats Informe Chaos 2015)

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

1.2. Procés de desenvolupament de software

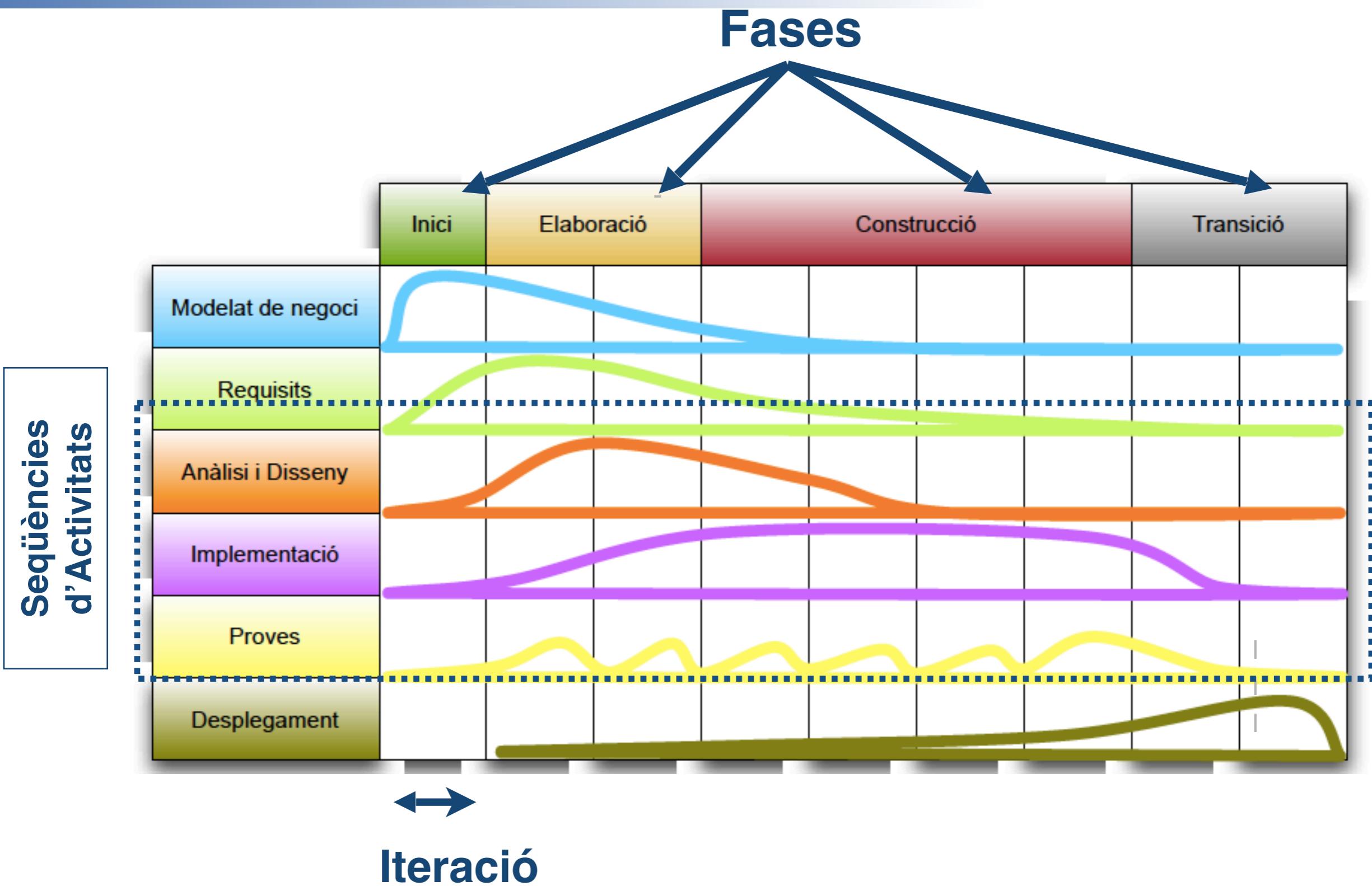
- El **procés unificat** (UP) de desenvolupament del software és un procés **iteratiu i adaptatiu** (Ivar Jacobson, Grady Booch y James Rumbaugh, 1999, Craig Larman. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process.*)



Es **composa de:**

- **Fases:** Inici, Elaboració, Construcció i Transició
- **Iteracions**
- **Disciplines** o seqüències d'activitats: requeriments, disseny, implementació i proves són les que es tractaran a l'assignatura
- **Artefactes**
- El procés unificat pot ser un procés **àgil (Agile RUP)**

Procés Unificat



1.2. Procés de desenvolupament de software

“El **procés** defineix la **seqüència d’activitats** que ha de fer l’equip de desenvolupament, els productes que han de crear-se, l’assignació de tasques, heurístiques i els criteris per controlar el procés”

Per exemple: Procés Unificat (UP) en Disseny Orientat a Objectes

“Un **model software** és una descripció d’un aspecte del sistema, expressada en un llenguatge ben definit”.

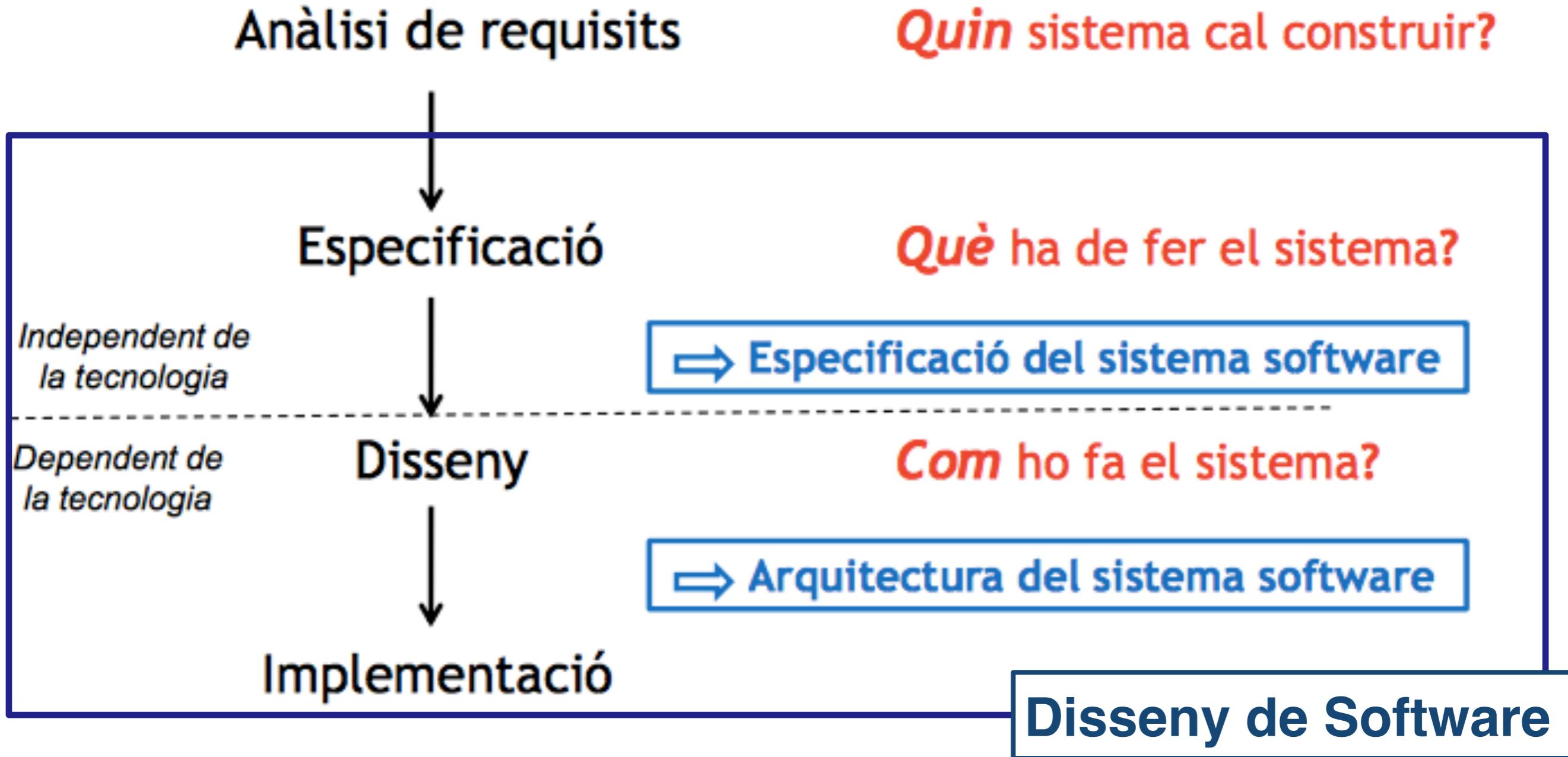
Per exemple: Llenguatge **UML**: Model de casos d’ús, Model Estructural, Model de Comportament, Model d’Activitats, Model d’Implementació.

Els models es defineixen amb l’ajut d’**artefactes** (Diagrames, codi, documentació) mitjançant uns convenis o **Llenguatges**.

Per exemple: Llenguatge **UML**: Diagrama de Casos d’Ús, Diagrama de Model de Domini, Diagrama de Classes, Diagrama de Seqüència de Sistema, etc.

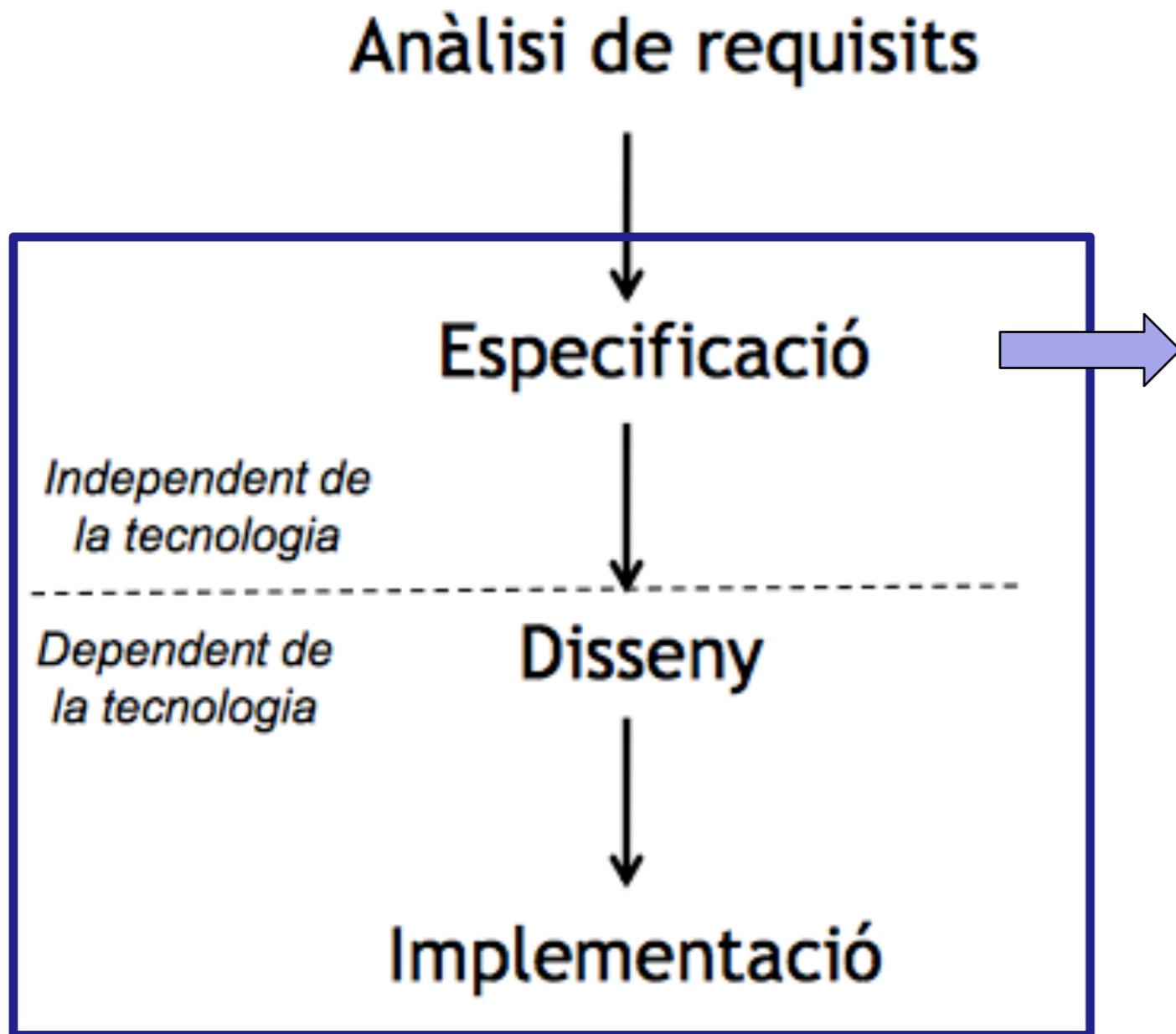
1.2. Procés de desenvolupament de software

Procés sistemàtic:



1.2. Procés de desenvolupament de software

Procés sistemàtic:



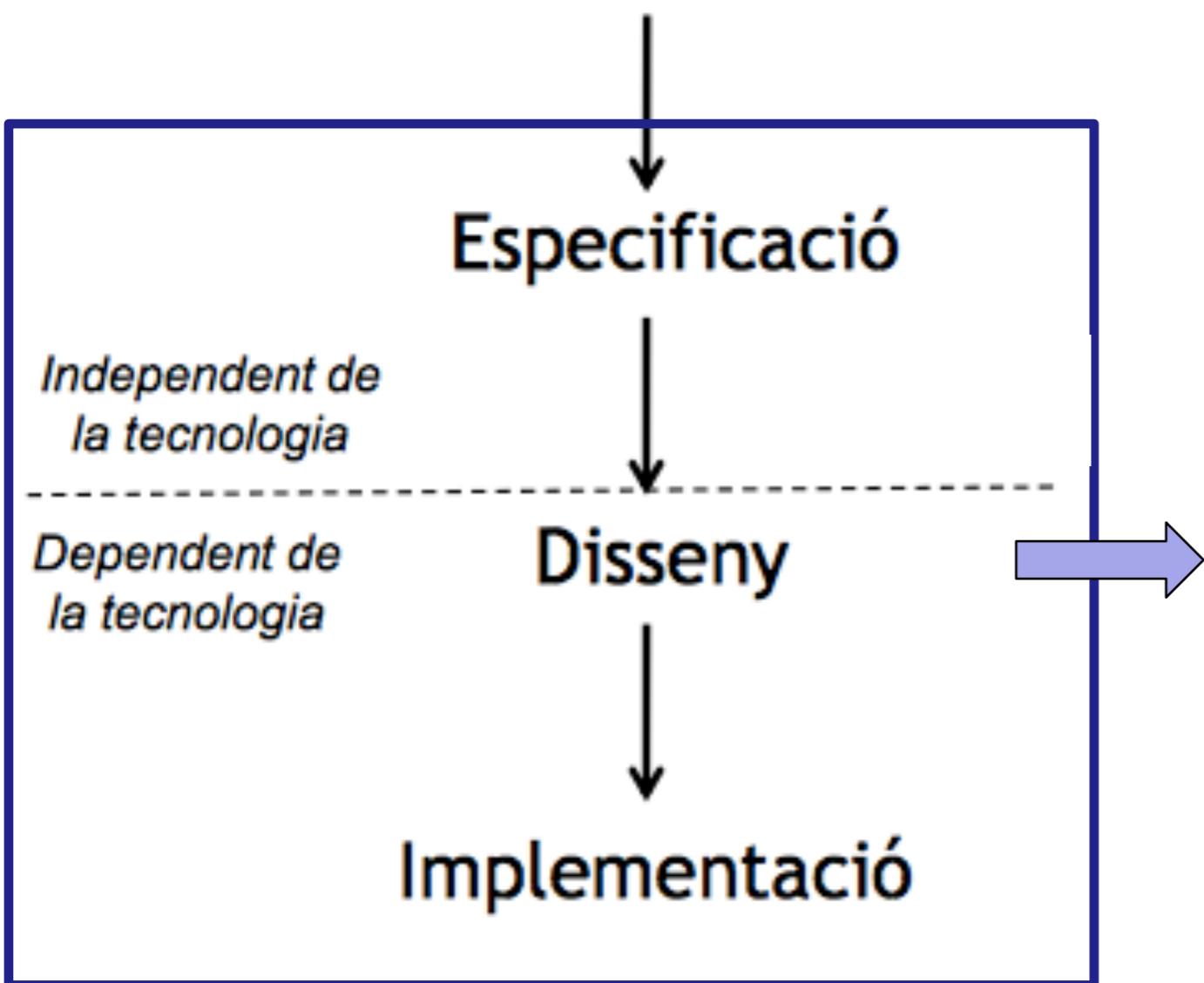
Què ha de fer el sistema?

- Especificació de la interacció amb l'usuari (**Casos d'ús, user stories**)
- Especificació de dades (**Esquema conceptual de les dades: Model de Domini**)
- Especificació de processos (**Esquema del comportament**)

1.2. Procés de desenvolupament de software

Procés sistemàtic:

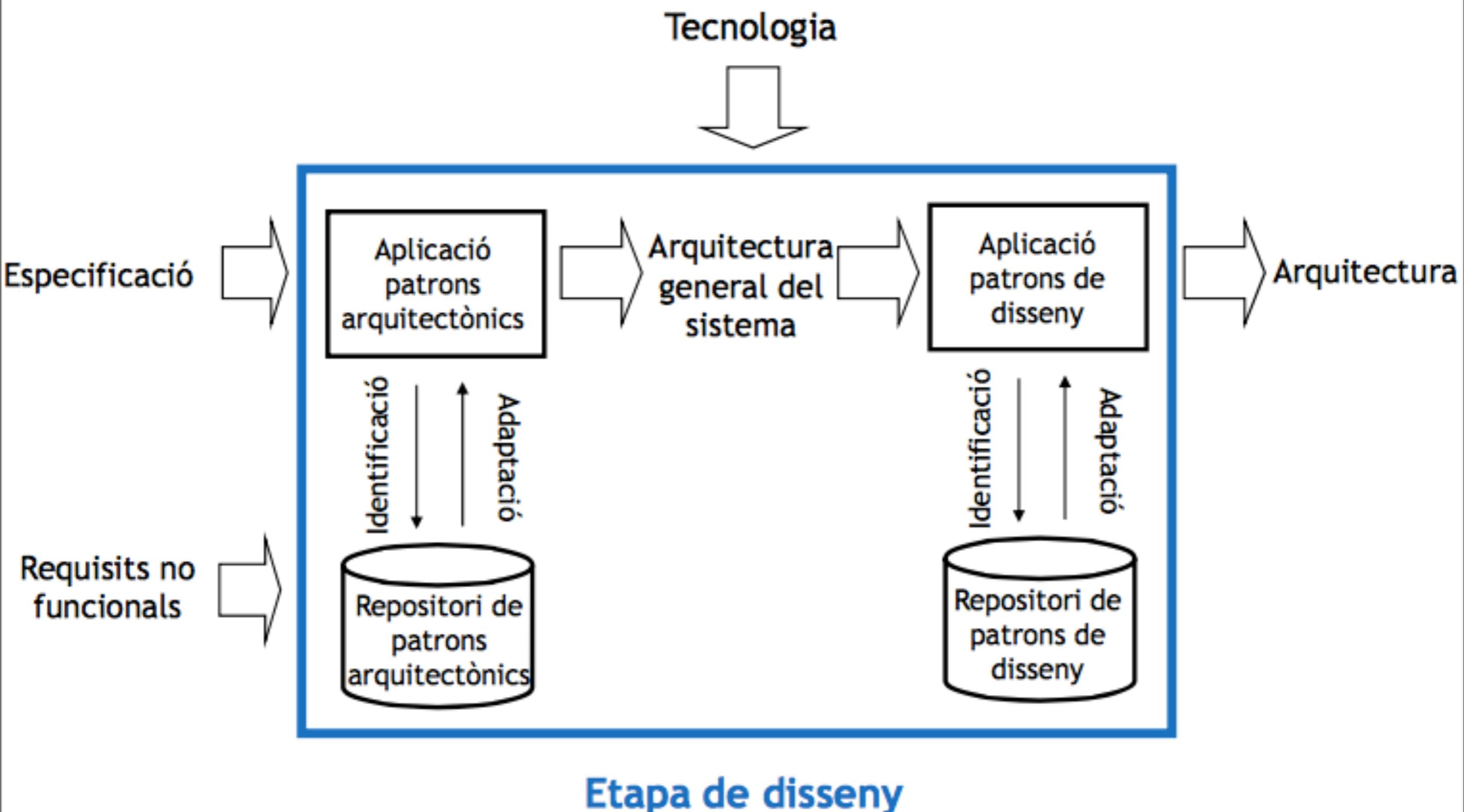
Anàlisi de requisits



Com ho fa el sistema?

- Descripció dels subsistemes i components del sistema software
 - Relació entre components
 - **Diagrama de classes**
 - Diagrames d'activitats

Disseny de Software



1.3. Desenvolupament guiat per tests

TDD (Test Driven Development): Integra el testing en el procés de desenvolupament de software:

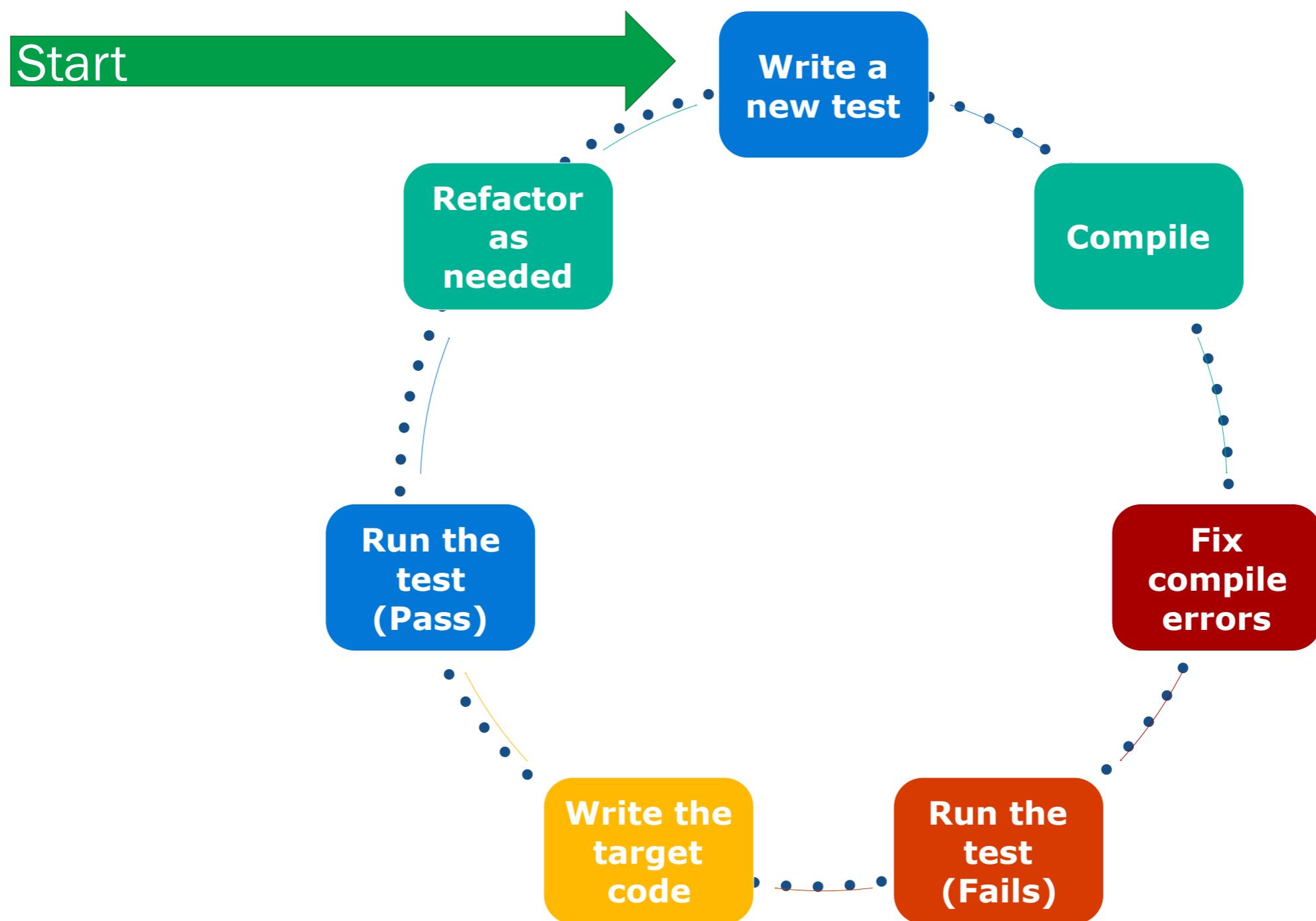
“Test Driven-Development:
Clean code that works.”
– Ron Jeffries

Els tests defineixen **el què** el codi ha de fer:

- Els tests provenen de les especificacions o dels casos d'ús refinats en històries d'usuari (**user stories**)
- Cal escriure el codi que passi els tests
- No s'escriu codi que no serveixi per a passar un test.

1.3. Desenvolupament guiat per tests

Red/Green/Refactor

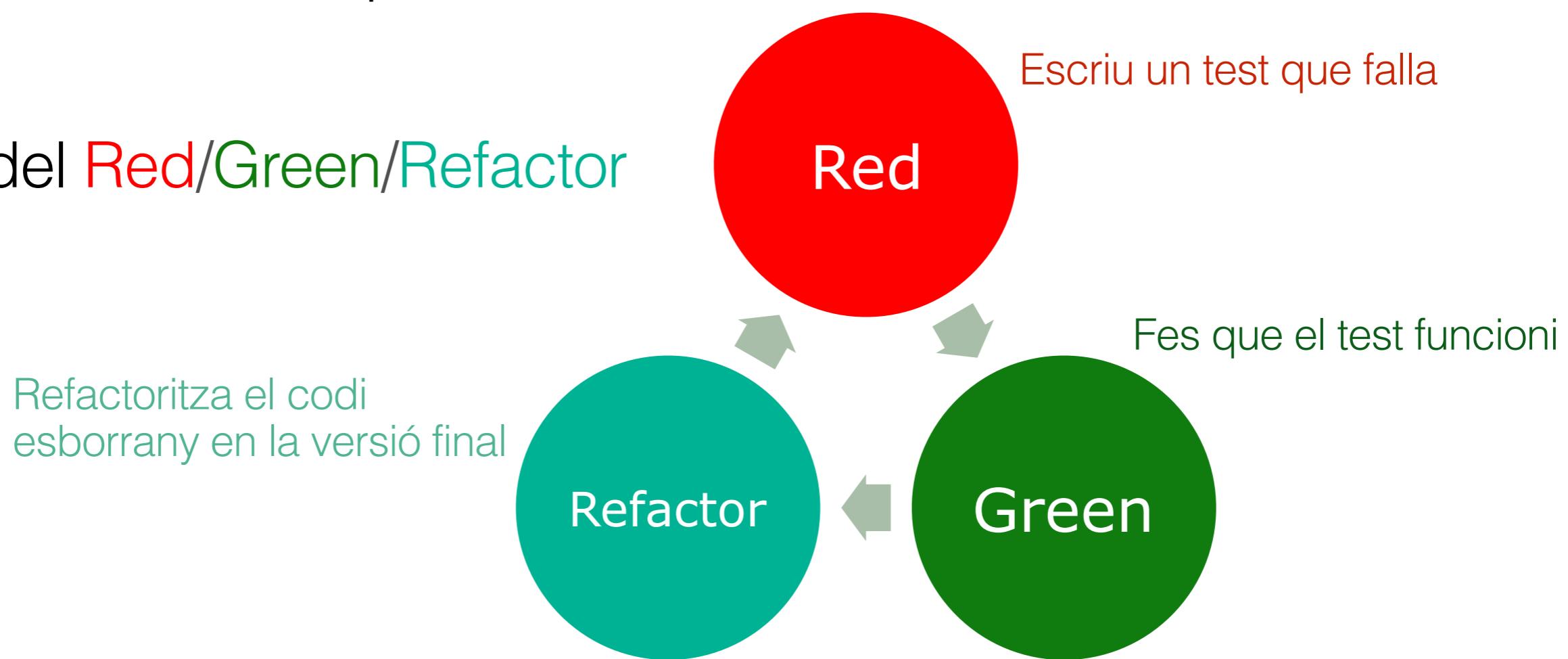


1.3. Desenvolupament guiat per tests

Basat en **dues** regles senzilles:

1. Escriu el nou codi només si el test automàtic ha fallat
2. Elimina la duplicació de codi.

Model Red/Green/Refactor



1.3. Desenvolupament guiat per tests

Cal que el codi sigui flexible i entendible!

“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”

Martin Fowler

“Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.”

John F. Woods

Sites interessants

<https://acodez.in/12-best-software-development-methodologies-pros-cons/>

Metodologies d'Enginyeria de Software

<http://dn.codegear.com/article/31863>

Article de UML en acció

https://www.tutorialspoint.com/design_pattern/index.htm

Tutorials de patrons de disseny amb explicacions senzilles

<https://cleancoders.com>

Vídeos de pràctiques de desenvolupadors. Refactorings i Disseny

<https://refactoring.guru/refactoring/course>

Curs de refactoring i patrons de disseny