

# Gestió de Projectes Software: Gestió àgil de Projectes

Curs 2014-15, QT



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

---

Facultat d'Informàtica de Barcelona

# Temari

- El manifest àgil
- Incepció de projectes àgils
- Scrum
- Històries d'usuari
- Estimació i planificació àgils
- **Proves, proves d'acceptació i demos**
- Pràctiques de l'equip àgil
- Kanban

# Sprint Review Meeting



- En finalitzar cada Sprint per:
- Inspeccionar l'Increment
- Adaptar el PB si és necessari
- Time boxed: 1 h / setm de duració del Sprint
- (+ 1h max. Per preparar)
- Col·laboració entre l'Equip Scrum i els stakeholders

# Sprint Review Meeting



- Inclou
  - El PO identifica què està FET i què no
  - L'entrenador discuteix què ha anat bé, quins problemes van sorgir i com es van solucionar
  - L'entrenador demostra el treball FET i respon preguntes
    - Es presenten les funcionalitats acabades (es podrien utilitzar en producció)
    - Es presenta la funcionalitat implementada a través d'una demo en directe
  - El PO discuteix el Product backlog i fa previsions
  - Tot el grup col·labora en què fer després, proporcionant feedback per a la següent reunió
- Resultat: Product backlog revisat

# Sprint retrospective meeting

- L'Equip Scrum fa introspecció i planifica la millora del procés
  - Inspeccionar com va ser el Sprint pel que fa a les persones, relacions, processos i eines
  - Identificar i potenciar les pràctiques que van ser ben
  - Crear un pla de millora de la pràctica de Scrum en l'equip
- Time boxed: 45 min / sem de durada del Sprint (o menys)
  - Després de l'Sprint Review i abans del siguiente SPM
- Assisteixen el SM, el DT i (opcionalment) el PO

# Sprint retrospective meeting

- Començar, Aturar, Continuar
  - Què cal començar a fer que no es fa?
  - Què ha anat malament i cal deixar de fer?
  - Què ha anat bé durant l'últim Sprint (per reforçar-)?
- L'ScrumMaster ho anota i tot seguit es parla de les millores que es poden implantar per al següent Sprint

# Criteris d'acceptació

- Permeten afegir detalls a la història
- Exemple: Com a usuari vull pagar la meva compra electrònica amb targeta
  - S'ha de poder pagar amb Bisa i NasterCar
  - Cal tractar els números de targeta incorrectes
  - Cal tractar les targetes expirades
  - Cal provar diferents quantitats totals, incloent quan s'excedeix el límit de la targeta

# Proves d'acceptació

- Verifiquen que una història d'usuari (o requisit) s'ha completat



# Introducció a les proves

- Prova: Activitat dirigida a l'avaluació d'un atribut o capacitat d'un sistema software amb l'objectiu de determinar si satisfà els requisits [Hetzel]
- Motivació


9/9

0800 Antan started  
1000 " stopped - antan ✓  
1300 (033) MP-MC 1.98264000 1.2700 9.037 847 025  
2.130476415 9.037 846 995 correct  
(033) PRO 2 2.130476415 4.615925059(-2)  
correct 2.130676415

Relays 6-2 in 033 failed special speed test  
in relay " 11.000 test.

Relays changed

1100 Started Cosine Tape (Sine check)  
1525 Started Multi Adder Test.

1545  Relay #70 Panel F  
(moth) in relay.

First actual case of bug being found.

1630 Antan started.  
1700 closed down.

Relay 3145  
Relay 3370

# Introducció a les proves

- Defecte (bug)
  - Diferència entre el comportament del software i el comportament especificat
  - Diferència entre el comportament del software i el comportament esperat
- Causes dels defectes de software?

# Introducció a les proves

- Cost dels defectes del software?
  - NASA Mars Climate Orbiter perduda el 1999
    - Per un error de conversió entre unitats SI i angleses
    - 655 milions de \$



# Introducció a les proves

- Cost dels defectes de software?
  - El SW de Knight Capital Group Inc. Inunda el mercat bursàtil amb ordres de compra-venta d'accions desastroses
    - Causat per un bug
    - 440 milions de \$ (4 x beneficis anuals any anterior)
    - - 75% del valor en borsa



# Introducció a les proves

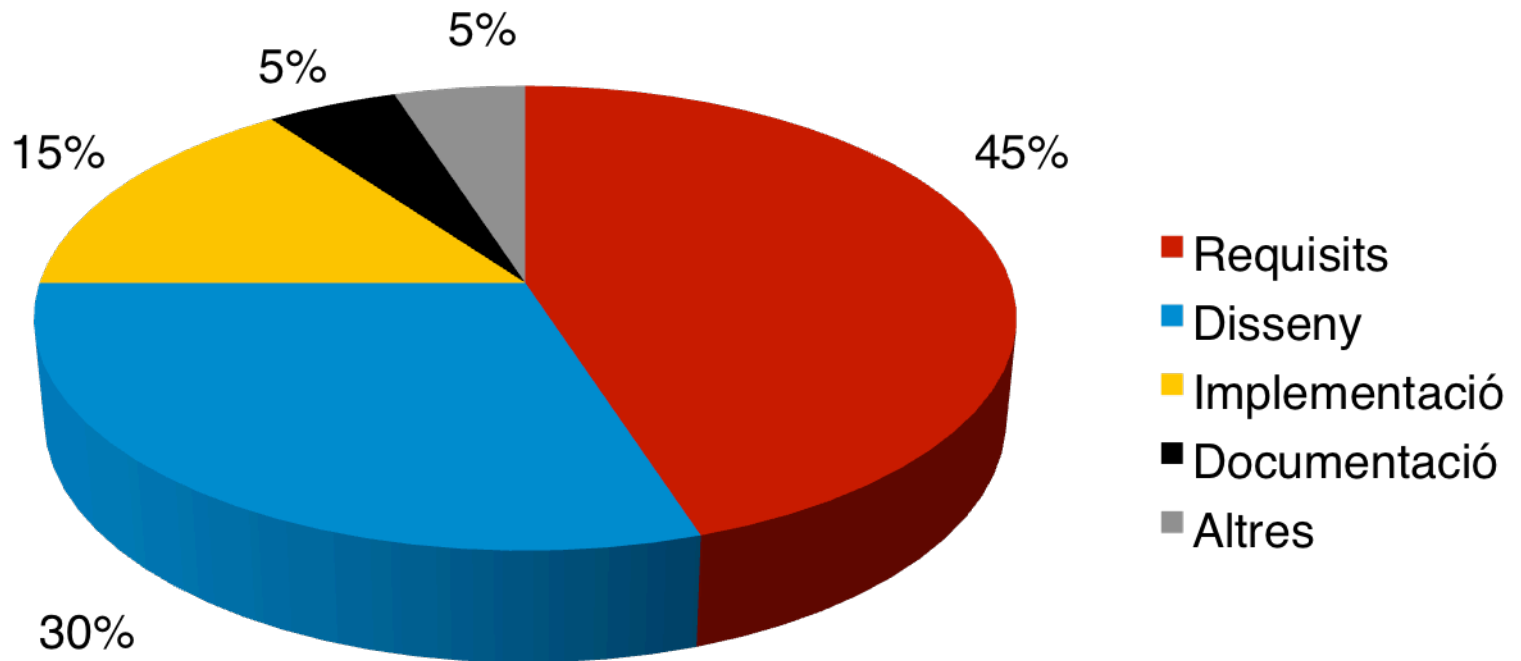
- Les proves de software
  - Milloren la qualitat
    - Defectes trobats i corregits
  - Milloren la confiança
    - Per exemple, a l'hora d'acceptar canvis
  - Permeten millorar el procés

# Introducció a les proves

- Objectius de les proves de software?
  - Trobar defectes
  - Millorar la confiança
  - Prevenir defectes
  - ~~Demostrar l'absència de defectes~~
- Es poden fer proves exhaustives?
  - No
  - Prioritzar segons el risc

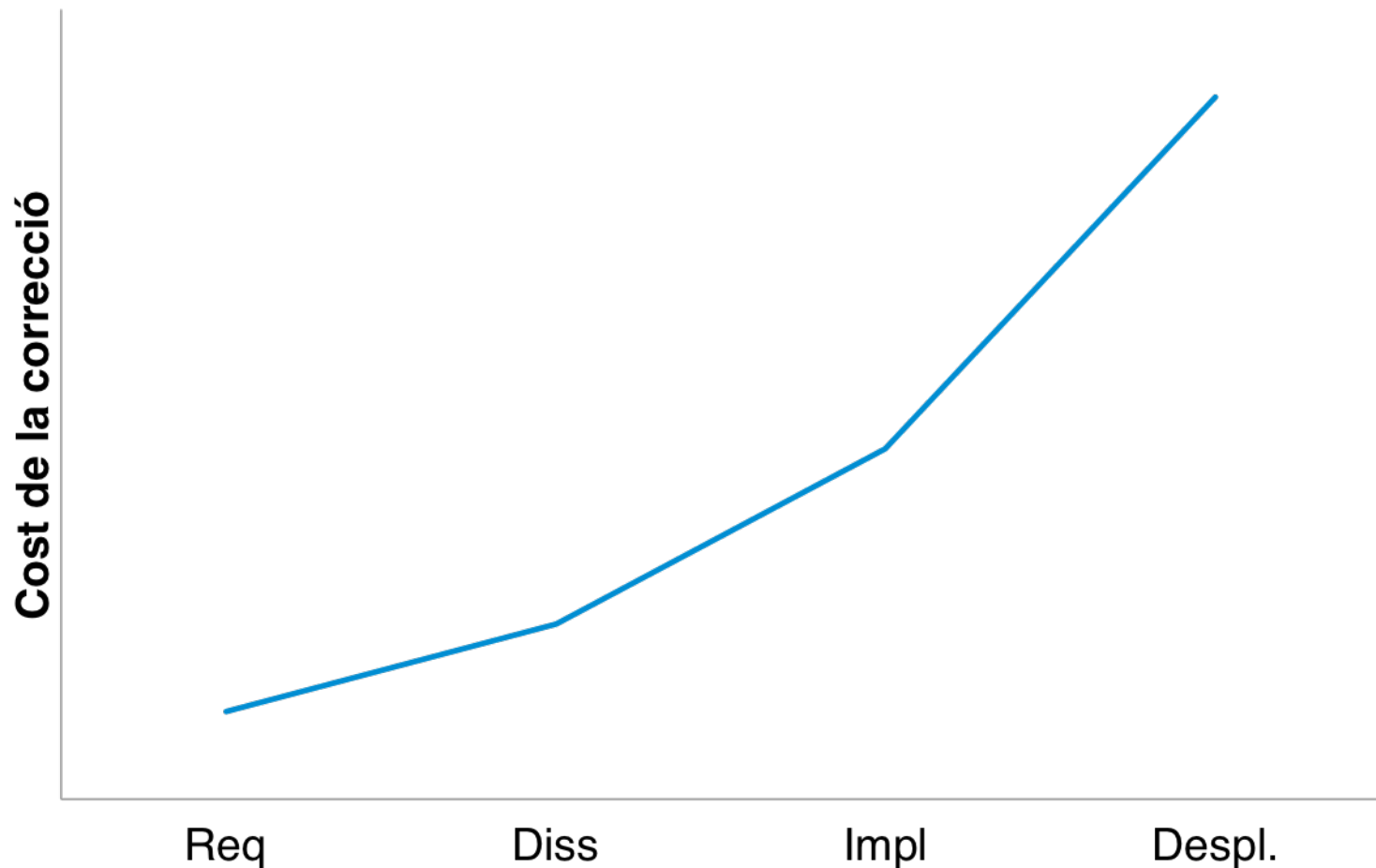
# Introducció a les proves

- Origen dels defectes



# Introducció a les proves

- Cost de la correcció de bugs



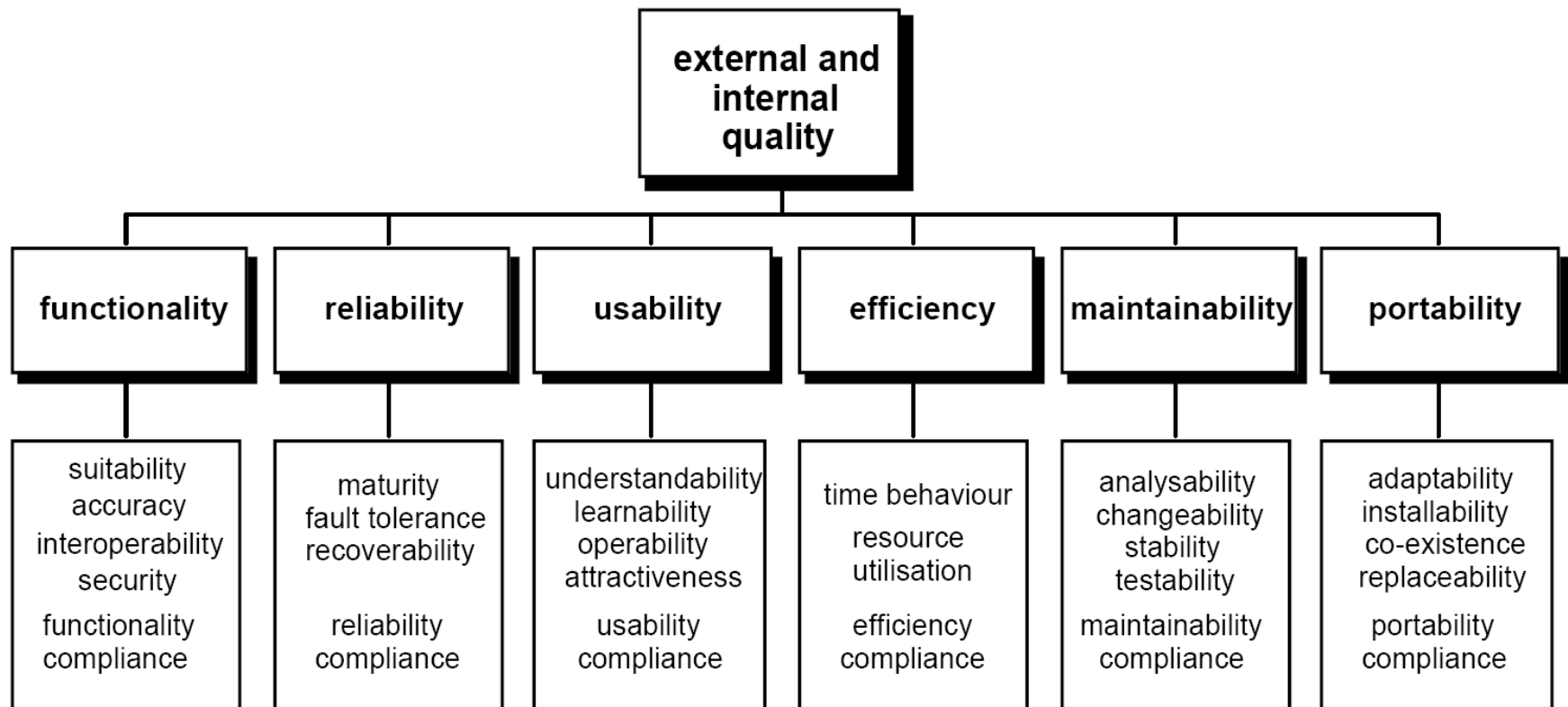


# Introducció a les proves

- Tipus de proves
  - Estàtiques (verificació) / dinàmiques (validació)
  - Unitàries / d'integració / de sistema / ...
  - Alfa / Beta / ...
  - En parlem més tard:
    - Manuals / Automatitzades
    - D'acceptació / de regressió / ...

# Introducció a les proves

- Què provar?



# Proves automatitzades

- Avantatges
  - Es poden executar cada cop amb un cost nul (o gairebé)
  - Permeten fer proves impossibles o molt difícils de fer manualment
    - Ex: Provar el comportament d'un sistema quan hi accedeixen 500.000 usuaris concurrentment
- Inconvenients
  - Són molt més costoses de preparar

# Proves automatitzades

- Recorda, recorda...
  - “Acceptem de bon grat canvis als requisits, inclús si arriben cap el final del desenvolupament. Els processos àgils aprofiten el canvi per a donar un avantatge competitiu al client.”
  - Proves de regressió

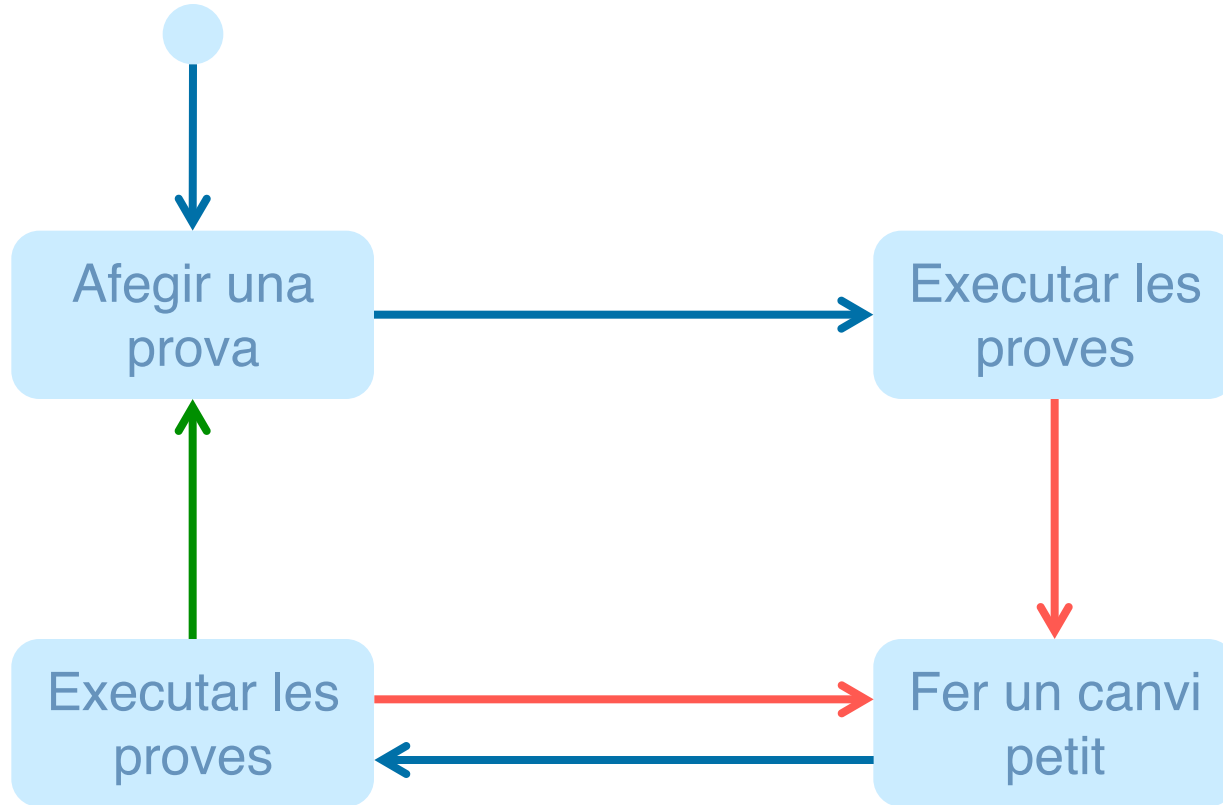
# Proves automatitzades

- Procés genèric de proves
  1. Situar l'objecte de la prova en un estat conegut
  2. Executar la funcionalitat a provar i recollir els resultats
  3. Consultar l'estat final de l'objecte de la prova
  4. Fer assercions sobre els resultats obtinguts i sobre l'estat final

# Proves automatitzades

- Entorn de proves de sistema
  - Com coneixem l'estat del sistema quan té “memòria”?
  - Separant els entorns
    - Entorn de desenvolupament
    - Entorn de proves d'integració
    - Entorn de proves d'acceptació
    - Entorn de demo
    - ...

# Proves d'acceptació automatitzades: ATDD



# BDD

- Un cas particular d'ATDD...
- ... que fa servir una especificació en un llenguatge ubic
  - **ubiqüitat**  
*f.* [RE] [FS] [LC] Fet d'estar present pertot.



# BDD

- Proves **d'acceptació**?
  - Qui accepta o no un producte software?
  - Qui ha d'entendre les proves d'acceptació?
  - Entén Java / Scala / PHP / .Net?

# BDD

- Separar intenció i implementació
  - Intenció: La prova en el llenguatge dels stakeholders (DSL)
    - Pot evolucionar i corregir-se de forma independent de la implementació
    - Inclou dade com “un producte ‘Baldufa zx23’
  - Implementació: Traducció de les frases del DSL a llenguatge tècnic
    - Pot evolucionar i corregir-se de forma independent de la intenció
    - Afegeix detalls com l’identificador intern de la ‘Baldufa zx23’

# Demo de BDD

The screenshot displays an IDE interface with the following components:

- Project Explorer (Left):** Shows the project structure with folders for `src`, `test`, and `features`. The `CreateNewProduct.feature` file is selected.
- Feature File (Left Panel):** Contains the BDD feature file content:

```
# language: ca
#noinspection SpellCheckingInspection
Característica: Donar d'alta un producte al catàleg.

Escenari: Crear un nou producte
  Donat un producte "Prod1"
  Quan llisto els productes
  Aleshores obtinc una llista amb 1 element
  I l'element número 1 és el producte "Prod1"

Escenari: Evitar la creació de productes amb nom:
  Donat un producte "Prod1"
  Quan intento crear un producte "Prod1"
  Aleshores obtinc un error que diu: "Ja existeix"
```
- StepDefinitions.java (Right Panel):** Contains the Java code for the steps:

```
public class StepDefinitions {

    private ProductsRepository productsRepository = new Produc
    private ProductService productService = new ProductsServ
    private List<Product> resultProducts;
    private Exception exception;

    @Given("^un producte \"([^\"]*)\"$")
    public void I_create_a_product(String productName) throws
        productService.newProduct(productName);
    }

    @And("^llisto els productes$")
    public void
        I_list_the_products() throws Throwable {
        this.resultProducts = productService.listProducts();
    }

    @Then("^obtinc una llista amb (\\d+) elements?$")
    public void I_get_a_list_of_elements(int numberOfElements)
```
- Run Panel (Bottom):** Shows the execution results for the feature `Feature: Donar d'alta un producte al catàleg`. The results are as follows:

Test Results
Feature: Donar d'alta un producte al catàleg
Scenario: Crear un nou producte
Scenario: Evitar la creació de productes amb noms duplicats
Donat un producte "Prod1"
- SBT Console (Bottom):** Shows the command `4: Run` and the output `1 file committed: Canvis menors a les proves (moments ago)`.

# Referències

- The Big Cost of Software Bugs
  - <http://www.bloomberg.com/slideshow/2012-08-03/the-big-cost-of-software-bugs.html>
- Acceptance test automation
  - <http://www.thoughtworks.com/insights/blog/acceptance-test-automation>

# Gestió de Projectes Software: Gestió àgil de Projectes

Curs 2014-15, QT



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

---

Facultat d'Informàtica de Barcelona