

Gestió de Projectes Software:

Acceptació i control de qualitat en
projectes àgils

Curs 2018-19, QP



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

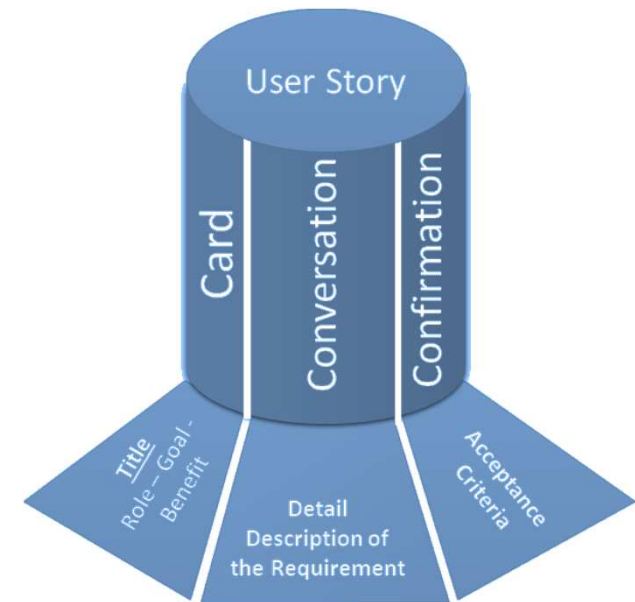
Índex

- Criteris d'acceptació
- Definició de Llest (DoR)
- Definició de Fet (DoD)
- Proves àgils
 - Principis
 - Piràmide d'automatització

Acceptació i control de qualitat

Història d'usuari – Recordem...

- 3 seccions
 - Card: Identificació història
 - (As a – I want – So that o text pla).
Funciona com a recordatori
 - Conversation: Converses per extreure el detall
 - Pantalles, notes, ...
 - Confirmation: **Criteris d'acceptació**
per determinar com
provar/demostrar que s'ha
aconseguit la història

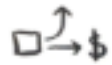


Acceptació i control de qualitat

Criteris d'acceptació – Objectius



Definir les **fronteres** d'una història d'usuari



Ajudar al PO a respondre què necessita en aquesta història per **donar valor**

- Típicament són els requisits funcionals mínims



Ajudar a l'equip a un **enteniment comú** de la història



Ajudar als desenvolupadors i testers a derivar les **proves**



Ajudar als desenvolupadors a saber quan han de **parar** afegint més funcionalitat a la història

Icones creades per Nathanael Coyne (@nathanaelb)

Acceptació i control de qualitat

Criteris d'acceptació – Com han de ser?

- Han d'expressar intenció, **no una solució**
 - “L'usuari pot escollir un compte” enlloc de
“L'usuari pot escollir el compte d'una llista desplegable”
- Independents de la implementació (la frase hauria de ser la mateixa sigui implementada la història en web, mòbil o Sistema d'activació per veu, p.ex.)
- Relativament a alt nivell (no tots els detalls necessiten ser escrits)

Acceptació i control de qualitat

Criteris d'acceptació – Com han de ser?

- Cada història ha de tenir almenys 1 criteri d'acceptació
- Ha de tenir un resultat clar per Passa o Falla
- Inclou criteris funcionals i no funcionals

Acceptació i control de qualitat

Criteris d'acceptació

Com a client de targeta de crèdit

Vull poder veure el meu balanç

Per a què pugui pagar

Criteris d'acceptació

- S'ha de mostrar el balanç a l'autenticar-se
- S'han de mostrar les últimes 3 transaccions realitzades
- S'ha de mostrar un missatge si el servei no està disponible

Acceptació i control de qualitat

Criteris d'acceptació

Com a comprador

Vull poder-me donar d'alta

Per a què disposi d'informació ja predeterminada a l'entrar

Criteris d'acceptació

- La paraula clau ha de ser de mínim 8 caràcters. Ha de contenir com a mínim un caràcter de cadascun dels següents grups: una lletra en minúscula, una en majúscula i un caràcter especial (!,@,#,\$,%)

Acceptació i control de qualitat

Definició de Llest (Definition of Ready, DoR)

- Definida per l'equip Scrum
- Quan un **ítem** està preparat per entrar al sprint amb un risc assumible
- Inclou valor negoci que permet priorització
- Totes especificacions requerides,...
- Criteri INVEST
 - Estimades i dimensionades per ser completades en un Sprint
- Lliure de dependències externes

Definition of Ready	
<input type="checkbox"/>	Business value is clearly articulated.
<input type="checkbox"/>	Details are sufficiently understood by the development team so it can make an informed decision as to whether it can complete the PBI.
<input type="checkbox"/>	Dependencies are identified and no external dependencies would block the PBI from being completed.
<input type="checkbox"/>	Team is staffed appropriately to complete the PBI.
<input type="checkbox"/>	The PBI is estimated and small enough to comfortably be completed in one sprint.
<input type="checkbox"/>	Acceptance criteria are clear and testable.
<input type="checkbox"/>	Performance criteria, if any, are defined and testable.
<input type="checkbox"/>	Scrum team understands how to demonstrate the PBI at the sprint review.

Acceptació i control de qualitat

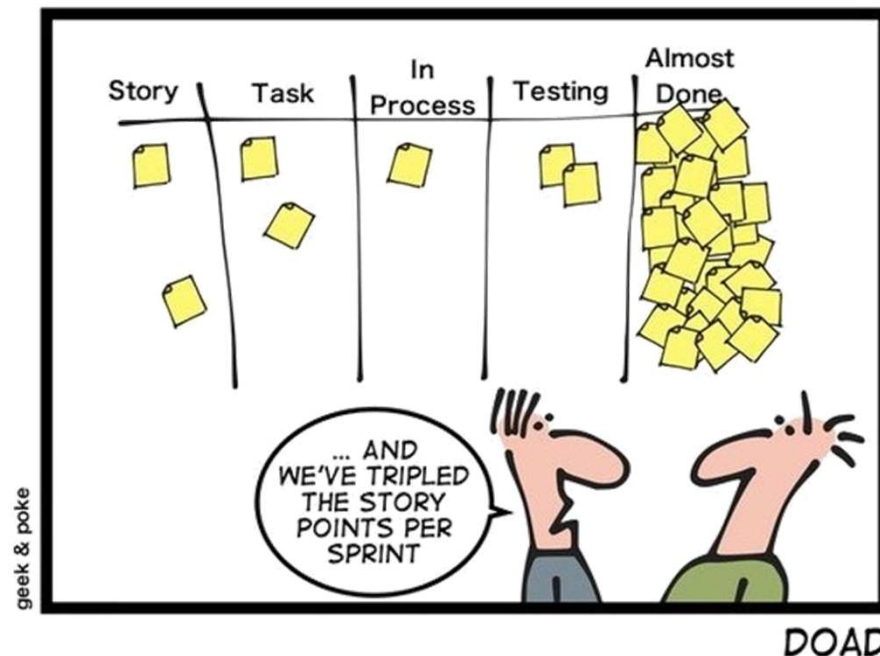
Definició de Fet (Definition of Done, DoD)

- Construir un enteniment comú dins l'equip sobre qualitat i completesa
 - Quins mínims de qualitat i lliurables s'han de complir en tots els objectius/requisits que s'aniran acceptant durant cada iteració?
 - Com sap l'equip que la feina està feta?
- Tenir sempre un producte “potencialment entregable”
 - Com pot saber el Product Owner que un ítem ha estat completat de forma exitosa?
 - Estem preparats per alliberar la versió?

Acceptació i control de qualitat

Definició de Fet (Definition of Done, DoD)

- Acordat entre el Product Owner i l'Equip de Desenvolupament a l'inici del projecte
- Es pot anar millorant en cada iteració
 - Segons s'estableixi en les retrospectives



Acceptació i control de qualitat

Definició de Fet (Definition of Done, DoD)

- Nivells
 - DoD d'una història d'usuari o **item** del product backlog
 - DoD de un sprint
 - DoD de una release
 - DoD d'una release a l'entorn d'integració
 - DoD d'una release a l'entorn de producció
 - ...

Acceptació i control de qualitat

Definició de Fet (DoD) – Història d'usuari

- ☐ Per qualsevol canvi al codi, s'ha escrit una prova que ho verifica
- ☐ Proves unitàries passades
- ☐ Codi revisat
- ☐ El codi implementa funcions de traces (logging)
- ☐ Codi completat i pujat al repositori de codi font
- ☐ Criteris d'acceptació de la història completats
- ☐ Proves funcionals passades
- ☐ Documentada a la wiki (disseny i manuals usuari)
- ☐ La interfície segueix els estàndards
- ☐ No s'ha detectat cap bug de severitat important

Acceptació i control de qualitat

Canviar la Definició de Fet (DoD)

- Pot evolucionar entre sprints/iteracions
- No s'ha de disminuir durant el sprint
 - “No tenim encara el servidor de proves configurat i manquen 2 dies per acabar el sprint”. Fem el review? Mostrem el que tenim en local?
- Es pot evolucionar dins el sprint
 - Exemple: DoD emulador hardware i tenim hardware real

Acceptació i control de qualitat

Definició de Fet i criteris d'acceptació

- La Definició de Fet és **transversal** a totes les històries
- Els criteris d'acceptació afecten a cada història en particular

Acceptació i control de qualitat

Requisits no funcionals

- Requisits no funcionals generals del sistema
 - Poden formar part del DoD (Definition of Done)
 - Ex: Totes les històries han de seguir l'estàndard d'usabilitat
- Requisits no funcionals específics d'una història
 - S'expressen com **criteris d'acceptació**
 - Ex: El temps de resposta ha de ser inferior als 10 segons

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) HP-MC 1.982647000 9.037847025
2.130476415 4.615925059(-2)

(033) PRO 2 2.130476415
conv 2.130676415

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

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

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

1700 First actual case of bug being found.
Antan started.
1700 closed down.

Relay 3375
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

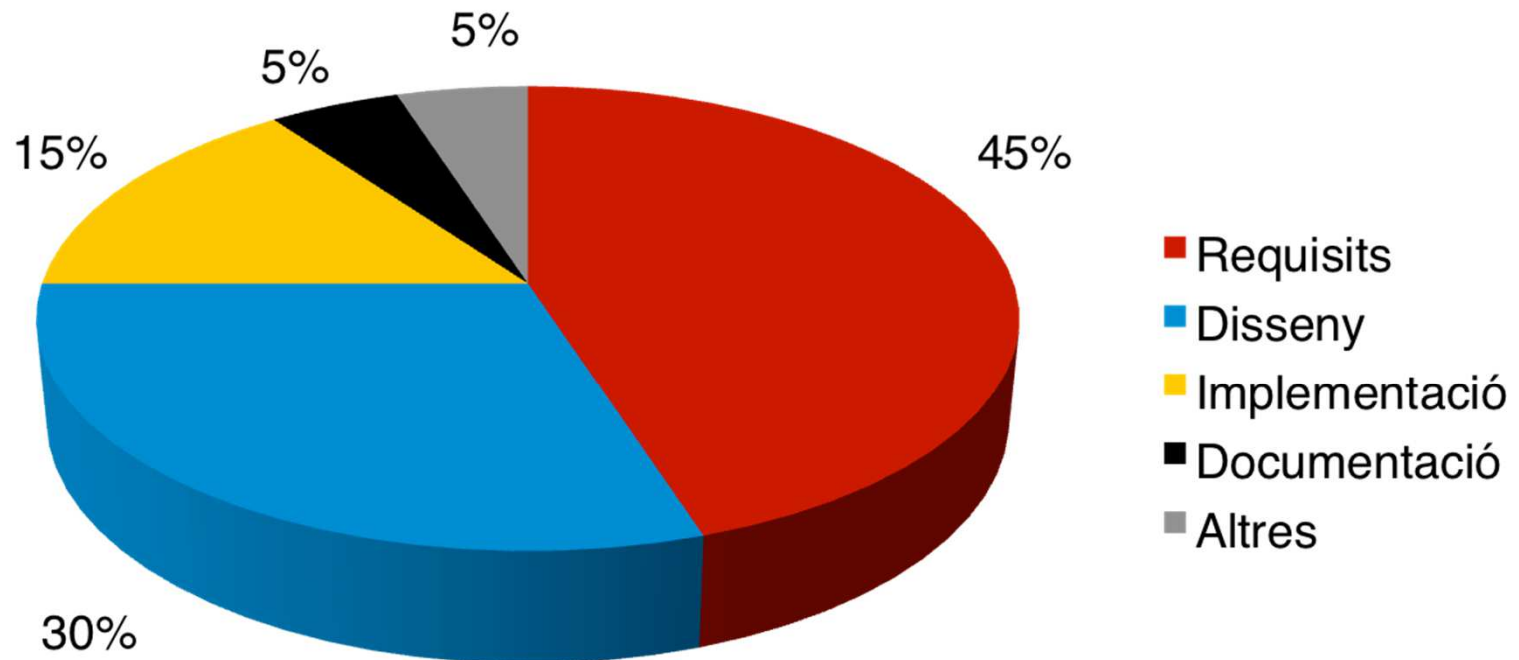
- Les proves de software
 - Milloren la qualitat
 - Defectes trobats i corregits
 - Milloren la confiança
 - Per exemple, a l'hora d'acceptar canvis
 - Permeten millorar el procés

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

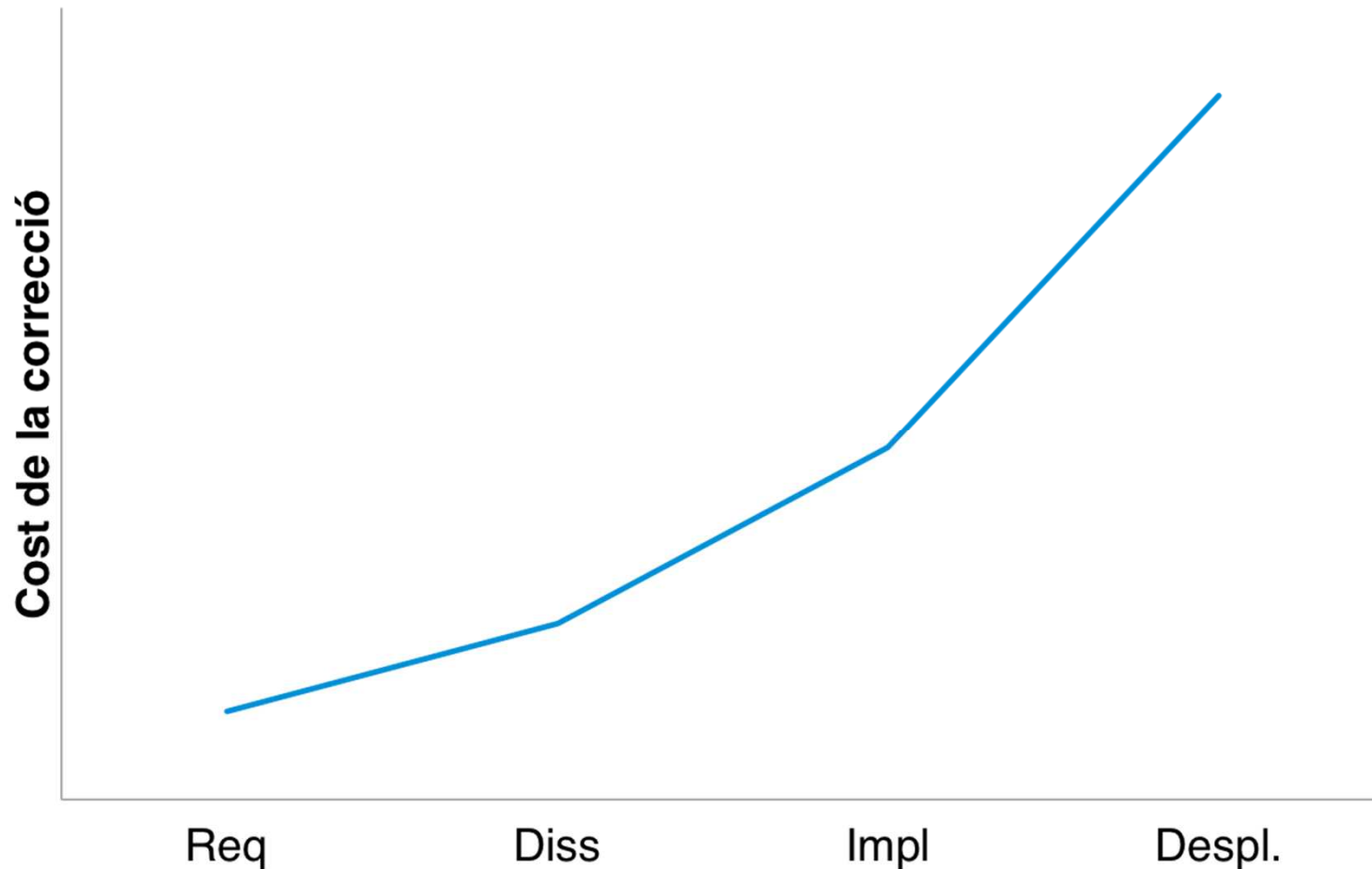
Introducció a les proves

- Origen dels defectes



Introducció a les proves

- Cost de la correcció de bugs

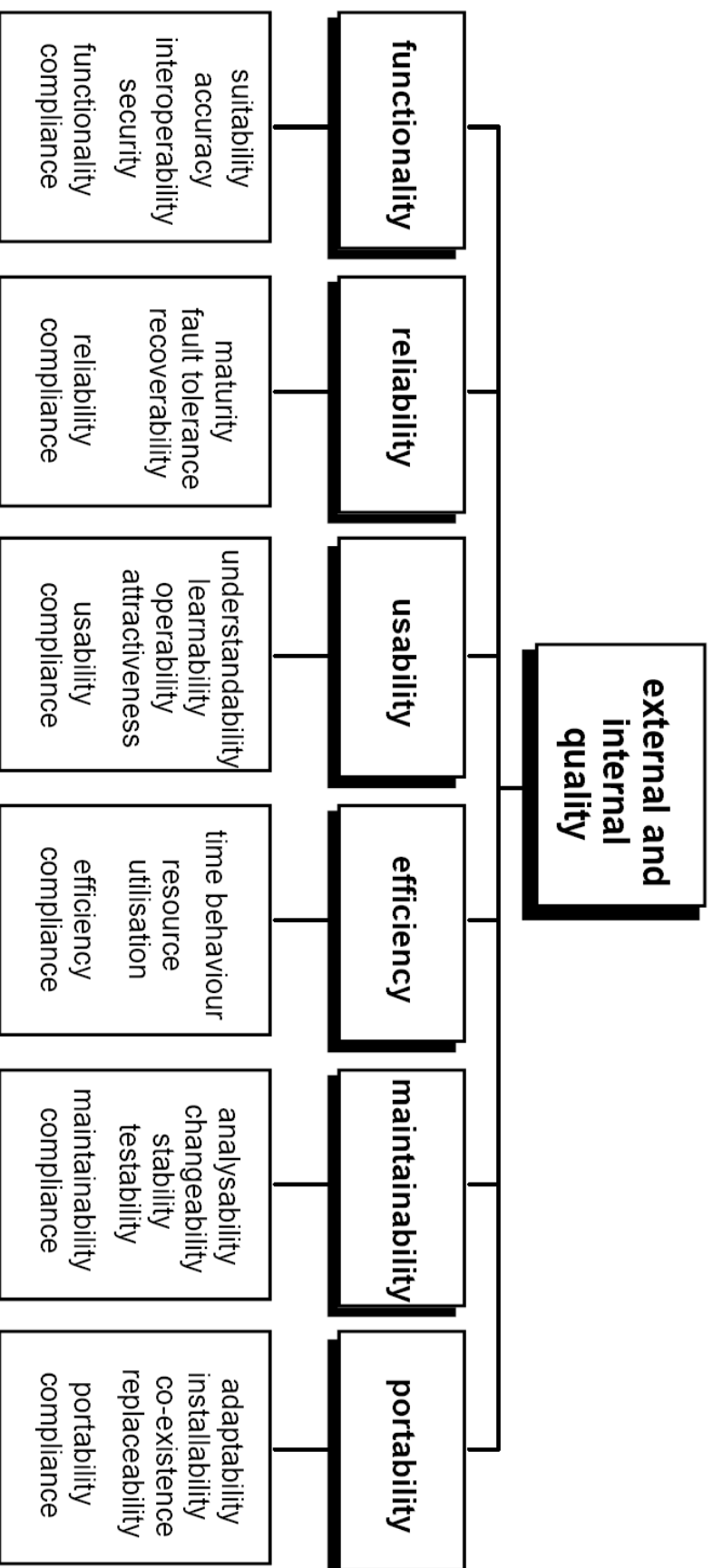


Introducció a les proves

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

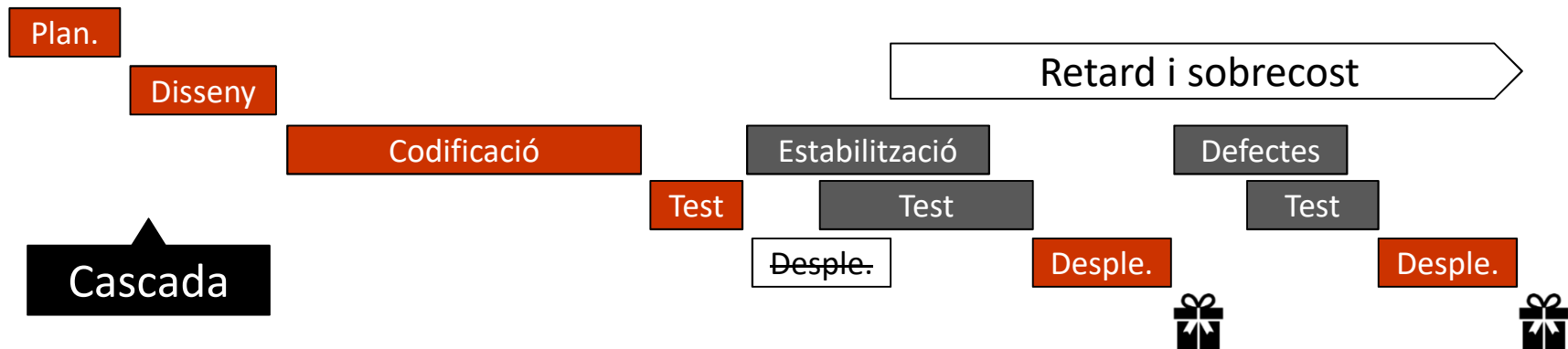
Introducció a les proves

- Què provar?

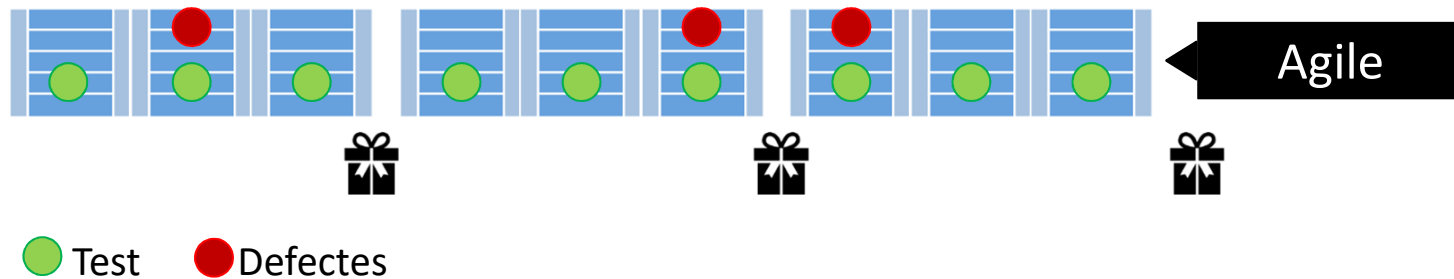


Acceptació i control de qualitat

Proves àgils - Motivació



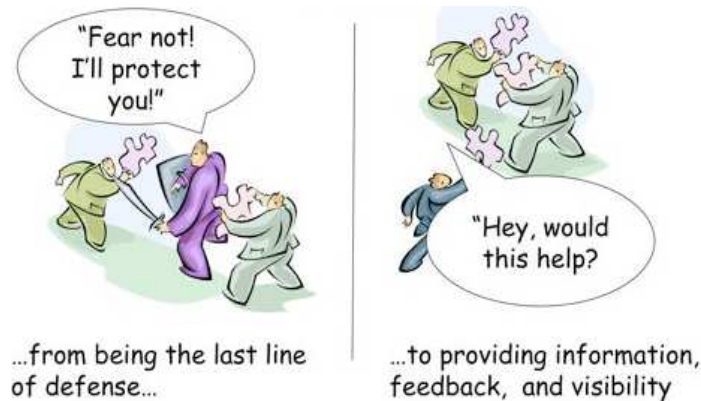
Les proves fetes posteriorment a la programació descobreixen defectes molt més costosos de solucionar. A més, la pressió d'apropar-se a la data pactada de lliurament produeix mala qualitat i descontrol tècnic.



Acceptació i control de qualitat

Proves àgils - Principis (1)

Shifting Focus



Les proves no són una fase

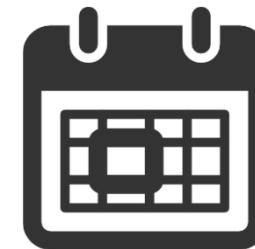


Tots proven



Reducció latència feedback

Provem de forma primerenca i freqüent



Acceptació i control de qualitat

Proves àgils - Principis (2)

Representen expectatives

Si el resultat no és l'esperat,
no estan clares les
expectatives, parlar amb els
stakeholders o el product
owner

Mantenir el codi net



Reduir documentació de proves



Listes de verificació
Essència de la prova
enlloc registre exhaustiu
incidències
Eines lleugeres

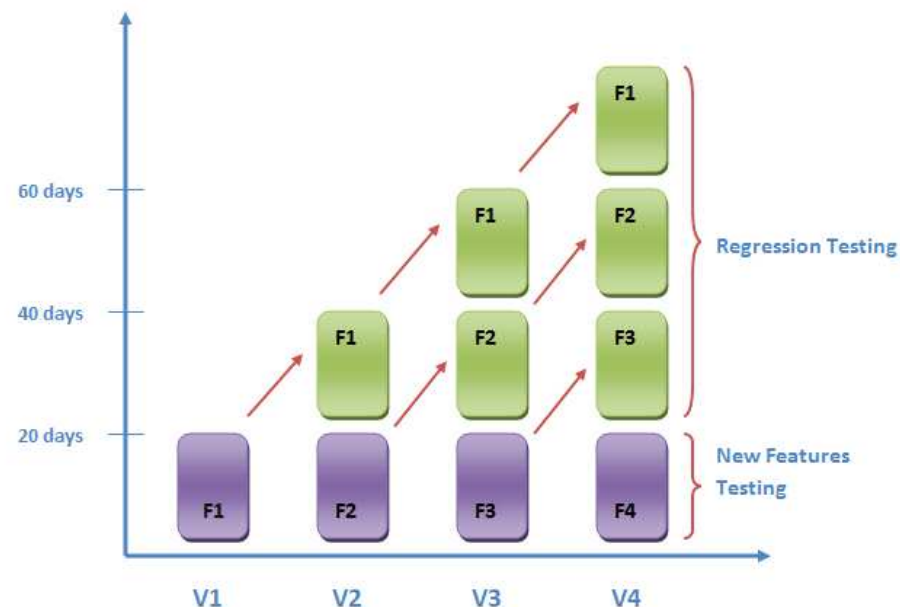
Part del Done



Acceptació i control de qualitat

Proves àgils - Proves de regressió

- Donar feedback de que no hem trencat una funcionalitat important
 - Pack proves de fum
 - Regressió completa (<1h)
- Candidates per automatització



Acceptació i control de qualitat

Proves àgils - Test Driven Development (TDD)

① Abans d'escriure qualsevol nou codi del sistema, **escrivim primer una prova unitària que falli**

VERMELL (Falla)

② Fem el que calgui per fer passar la prova.

Si veiem clara la implementació completa, afegir el nou codi. Si no ho sabem afegir el mínim per tal que la prova passi

REFACTORITZAR

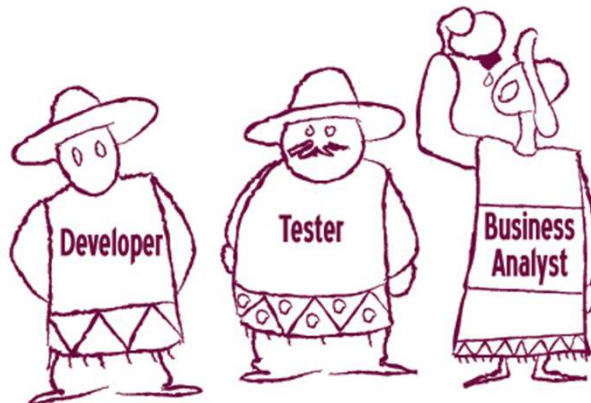
VERD (Passa)

③ Millorem la **qualitat del codi** (codi net, clar). Comprovar que **passen tots els tests**. Repetir cicle per següent prova unitària

Acceptació i control de qualitat

Proves àgils – Especificació per l'exemple

- Especificació per l'exemple
 - Els requisits abstractes i especificacions no són una bona eina de comunicació
 - Discutir **exemples reals** per construir un enteniment comú
 - Usar aquests exemples com **criteris d'acceptació**
 - Automatitzar les proves d'acceptació
 - Usar les proves com una especificació viva per facilitar el canvi



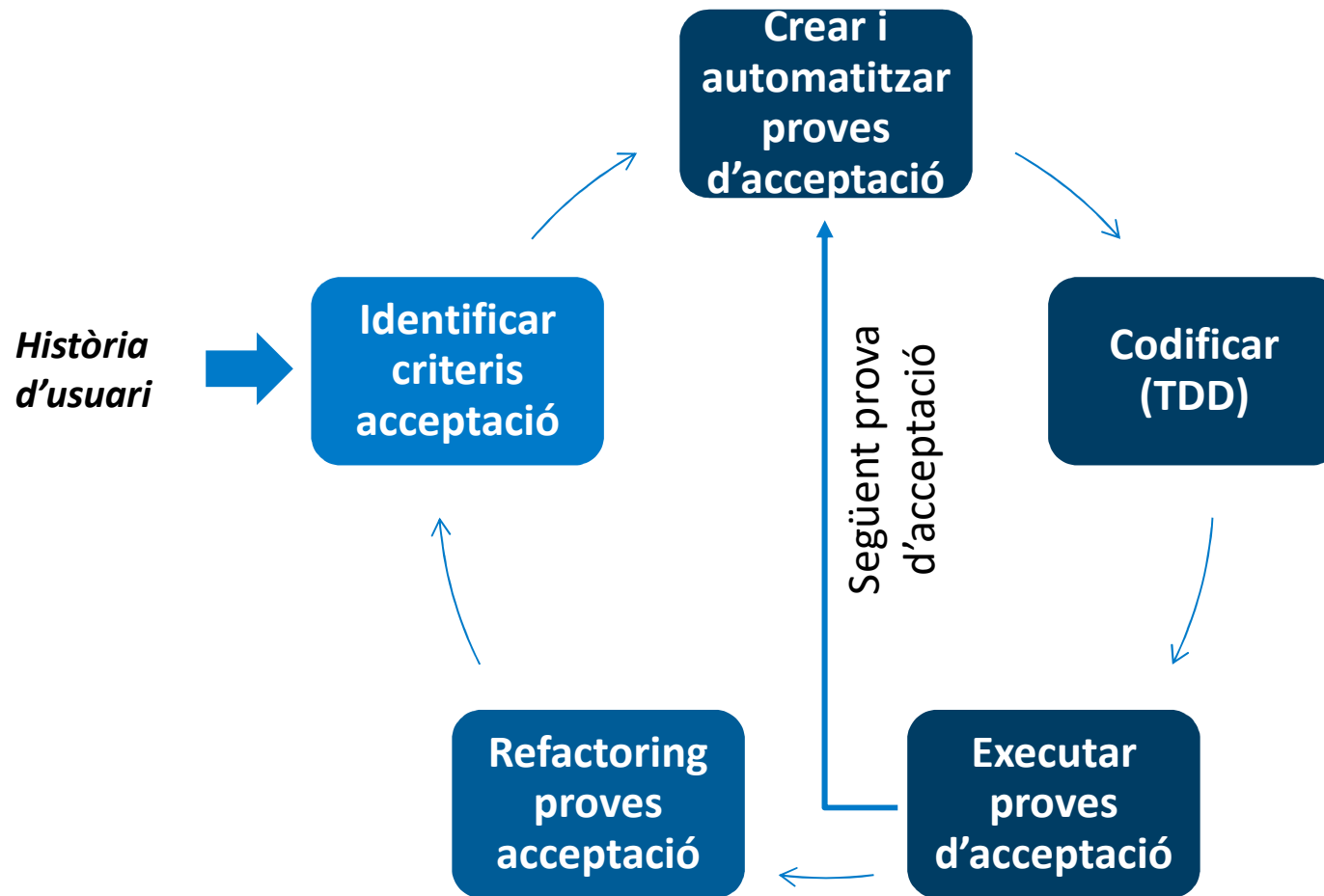
Acceptació i control de qualitat

Proves àgils – Especificació per l'exemple

- Exemple tradicional
 - El sistema ha de calcular la taxa (tradicional)
- Exemple d'especificació per l'exemple
 - Per una subscripció de 20€ amb una proporció del 10% de taxa, el sistema carrega un total de 22€ del compte de l'usuari (especificació per l'exemple)

Acceptació i control de qualitat

Proves àgils - Acceptance Test Driven Development



Acceptació i control de qualitat

Proves àgils - Behaviour Driven Development

- Similar al ATDD però focalitzat en el **comportament** del sistema
- S'usa el format **Gherkin**:

Given algun context inicial (estat de l'aplicació a l'inici del cas de prova)

When succeeix algun esdeveniment

Then hi ha alguns resultats

- Diem que es converteixen en més que proves, són **especificacions executables**

Acceptació i control de qualitat

Proves àgils - Behaviour Driven Development

Scenario: Email duplicat

Quan algú intenta crear un compte des d'una adreça de correu que ja existeix

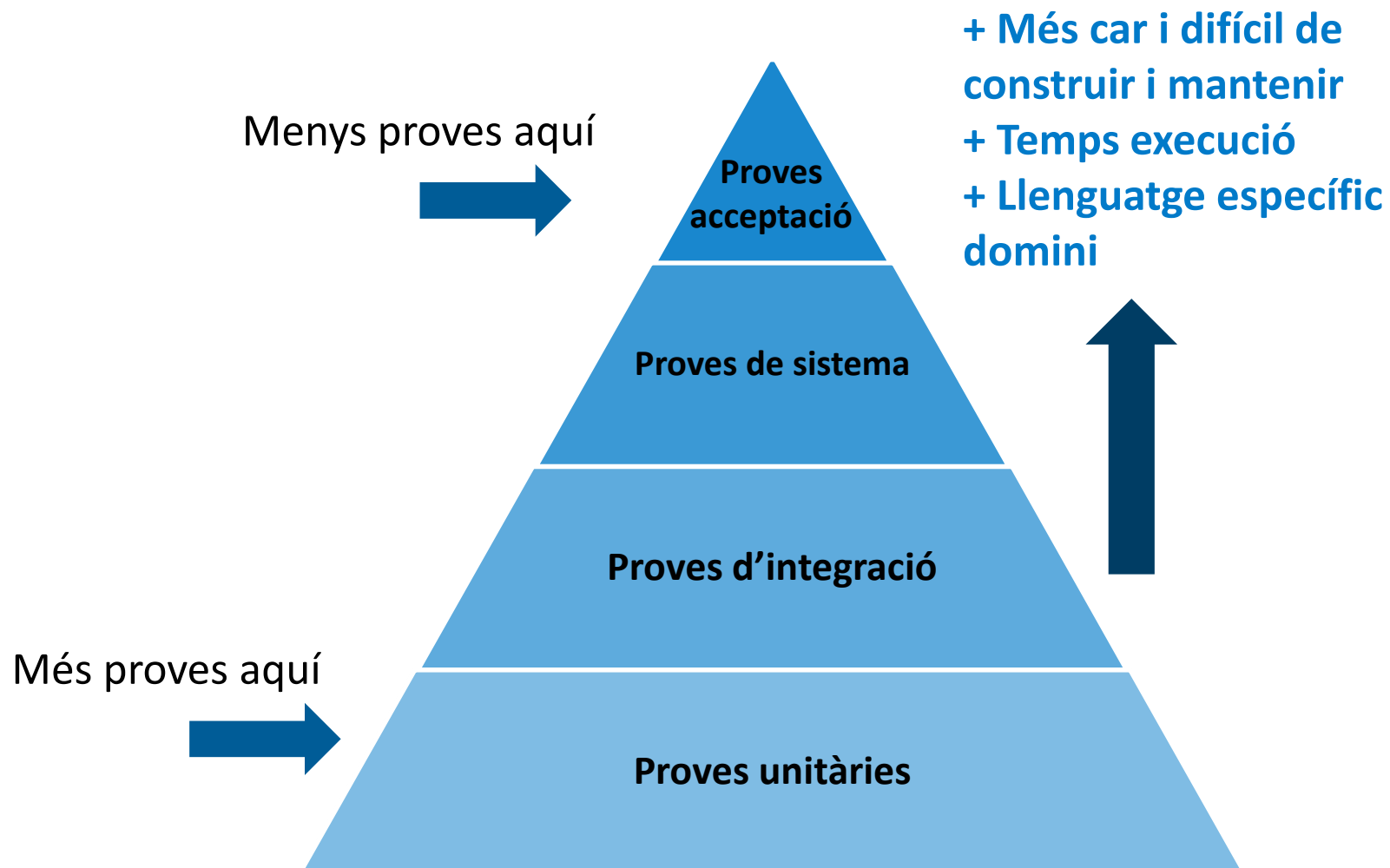
Given He escollit fer signup

When Em dono d'alta amb una adreça de correu ja registrada

Then hauria de dir-me que l'adreça de correu ja està registrada

Acceptació i control de qualitat

Proves àgils - Piràmide de proves



Acceptació i control de qualitat

Integració contínua

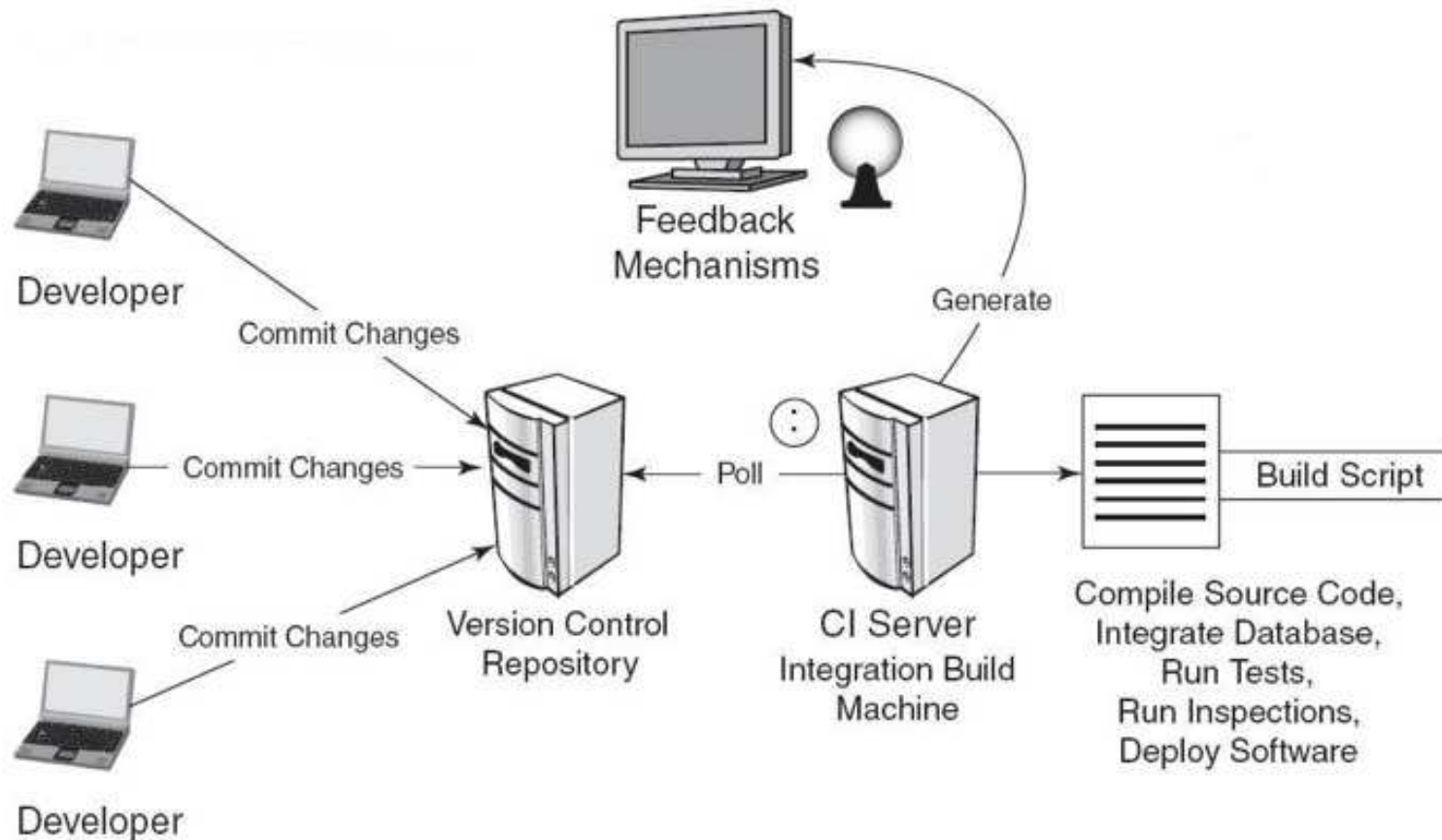
Continuous Integration is a software development practice where members of a team **integrate their work frequently**, usually each person integrates at least daily – leading to multiple integrations per day.

Each integration is verified by an automated build (including test) **to detect integration errors as quickly as possible.**

Martin Fowler

Acceptació i control de qualitat

El cicle de build i feedback continu de la qualitat



Gestió de Projectes Software:

Acceptació i control de qualitat en
projectes àgils

Curs 2018-19, QP



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona