

FIB - Gestió de Projectes del Software

Gestió àgil (AGILE)

Temari

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

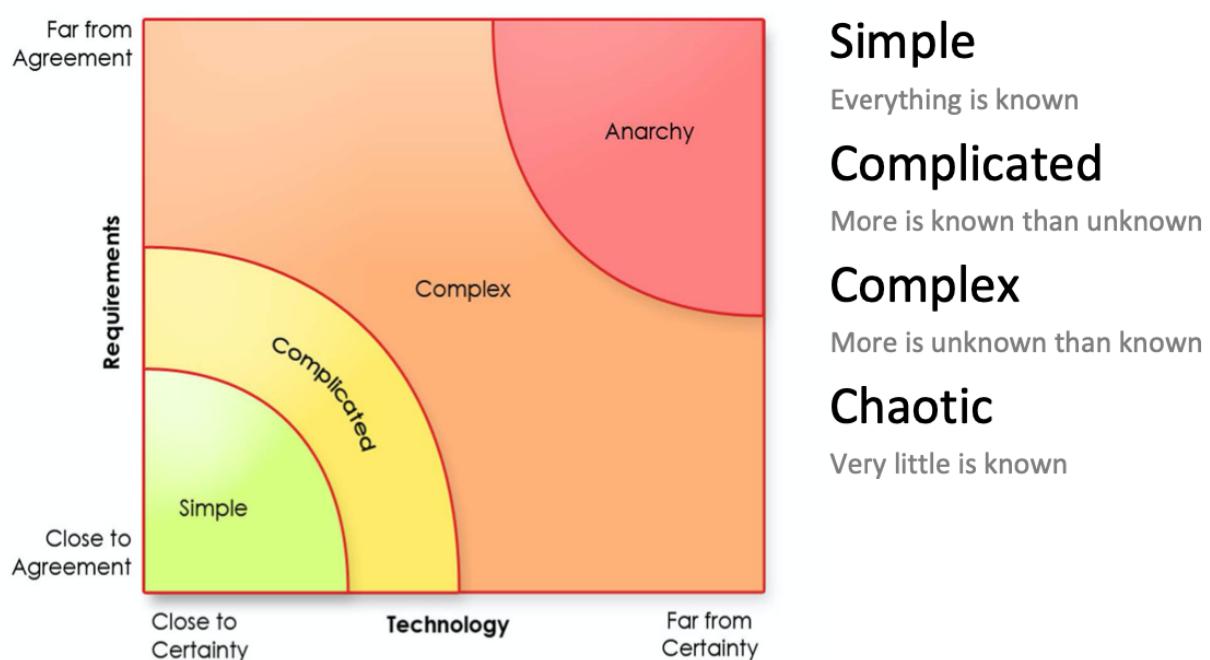
El manifest àgil

Introducció i història

Per què enfocaments predictius?

- Desenvolupament sostenible de software de qualitat
- Entrega a temps i dins de pressupost
- No dependre de persones individuals
- Treball en equip
- Assegura un procés predecible i repetible

Introducció als projectes àgils - Aplicació dels mètodes àgils



Una mica d'història

El febrer de 2001, 17 desenvolupadors es reuneixen a Utah per discutir sobre mètodes de desenvolupament lleugers.

- Kent Beck, Alistair Cockburn, Ward Cunningham, Martin Fowler, Jim Highsmith, Andrew Hunt, Ron Jeffries, Robert C. Martin, Ken Schwabber, Jeff Shutterland, Dave Thomas

El manifest

Individus i interaccions per sobre de processos i eines

Programari que funciona per sobre de documentació exhaustiva

Col·laboració amb el client per sobre de negociació de contractes

Resposta al canvi per sobre de cenyir-se a una planificació

És a dir:

encara que els elements de la dreta tenen valor,

nosaltres valorem més els de l'esquerra.

Valors i principis

Valor

1 1 m. o f. [LC] [AR] [AN]

Qualitat o conjunt de qualitats que fan que alguna cosa o alguna persona sigui preuada, valgui. El valor d'una obra literària. El valor nutritiu d'un aliment. Escala de valors.

Principi

3 1 m. [FS] [FIF]

Veritat fonamental, axioma, postulat. Els principis d'una doctrina.

6 m. [LC] [FS]

Regla de conducta. Un principi de moral. Un home sense principis. No té principis.

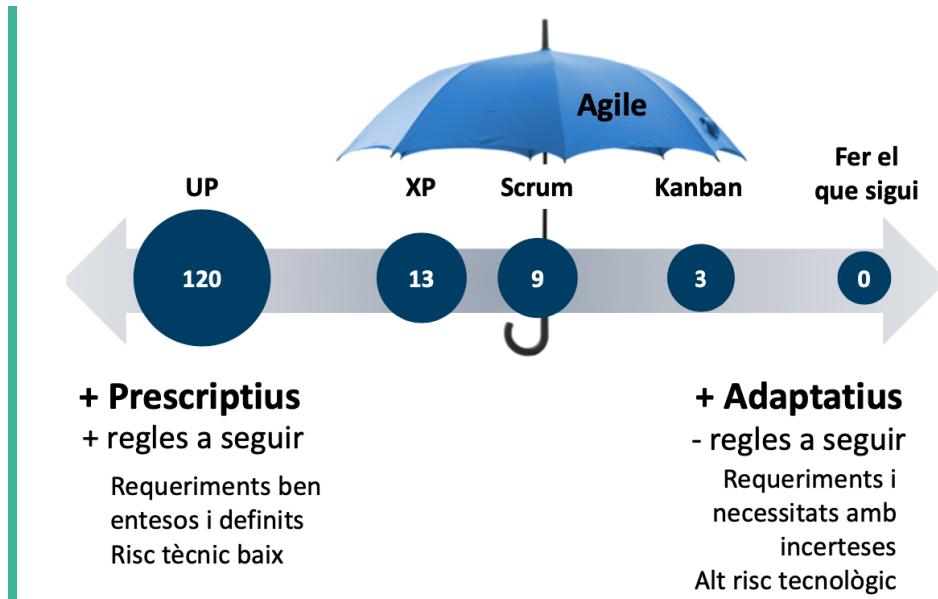
12 principis

1. La nostra principal prioritat és satisfer al client mitjançant el lliurament primerenc i continu de programari que aporti valor.
2. 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.
3. Lliurem amb freqüència programari que funcioni, des d'un parell de setmanes fins a un parell de mesos, amb preferència per l'escala de temps més curta.
4. La gent de negoci i els desenvolupadors han de treballar junts de manera quotidiana durant tot el projecte.
5. Construïm projectes amb l'ajuda d'individus motivats. Els donem l'entorn i el recolzament que necessiten i confiem en ells per fer la feina.
6. El mètode més eficient i efectiu de comunicar informació cap a i dins d'un equip de desenvolupament és la conversa cara a cara.
7. El programari que funciona és la principal mesura de progrés.

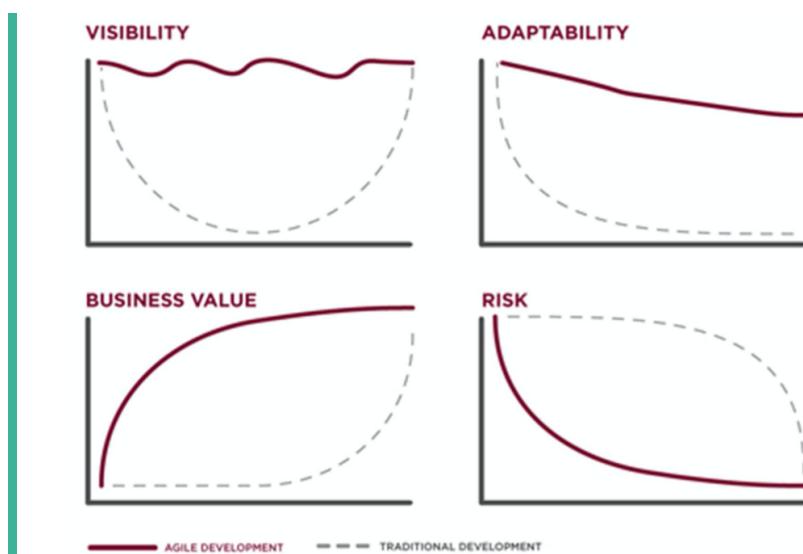
8. Els processos àgils promouen el desenvolupament sostingut. Els promotores, desenvolupadors i usuaris han de ser capaços de mantenir un ritme constant de manera indefinida.
9. L'atenció contínua a l'excel·lència tècnica i al bon disseny millora l'agilitat.
10. La simplicitat, l'art de maximitzar la quantitat de feina que no es fa, és essencial.
11. Les millors arquitectures, requisits i dissenys emergeixen d'equips autoorganitzats.
12. En intervals regulars, l'equip reflexiona sobre com ésser més efectiu, s'afina i ajusta el seu comportament d'acord amb això.

Introducció a les metodologies àgils

I també adaptatiu



Agile Value Proposition

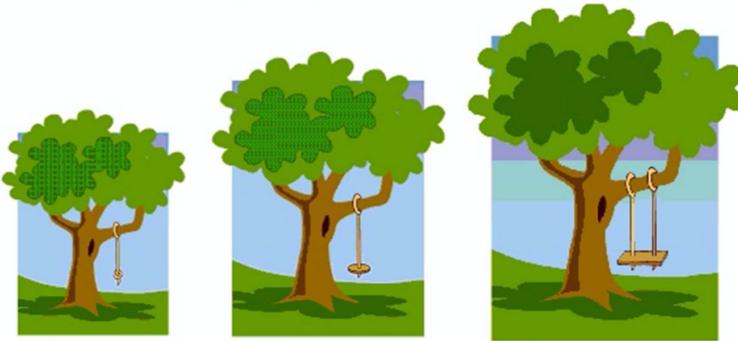


Agile és iteratiu i incremental...

Agile = Iterative + Incremental

Don't try to get it all right from the beginning

Don't build it all at once



Per què agile?

Reasons for Adopting Agile

The reasons stated for adopting agile follow a similar ranking as in the previous year though we did see the biggest change in responses in accelerate software delivery (75% compared to 69% last year), enhancing delivery predictability (46% compared to 30% last year), improving IT/Business alignment (49% compared to 42% last year), and reducing project cost (24% compared to 18% last year).



*Respondents were able to make multiple selections.



Agile Maturity

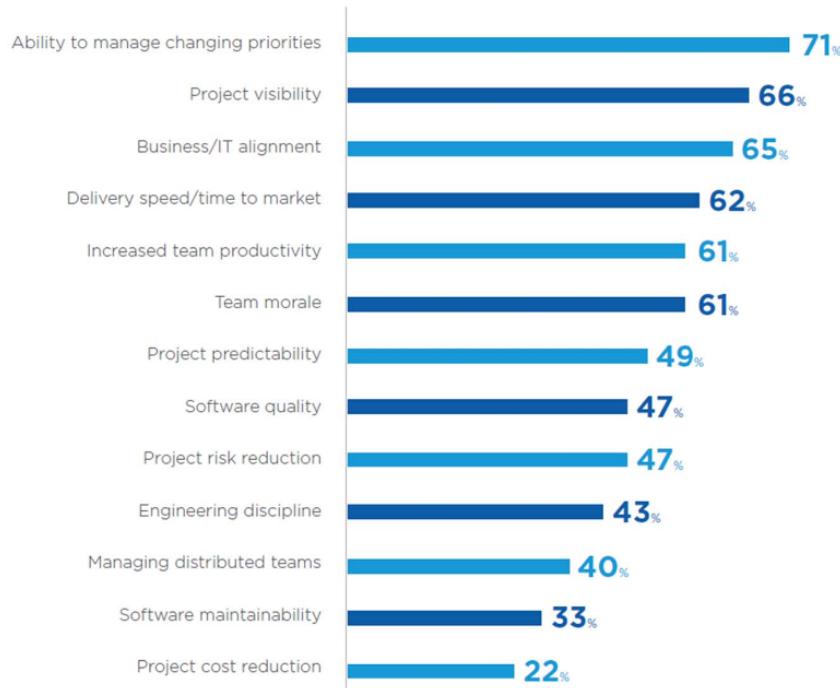
The vast majority of respondents (84%) said their organization was at or below a "still maturing" level.



Beneficis d'adoptar Agile

Benefits of Adopting Agile

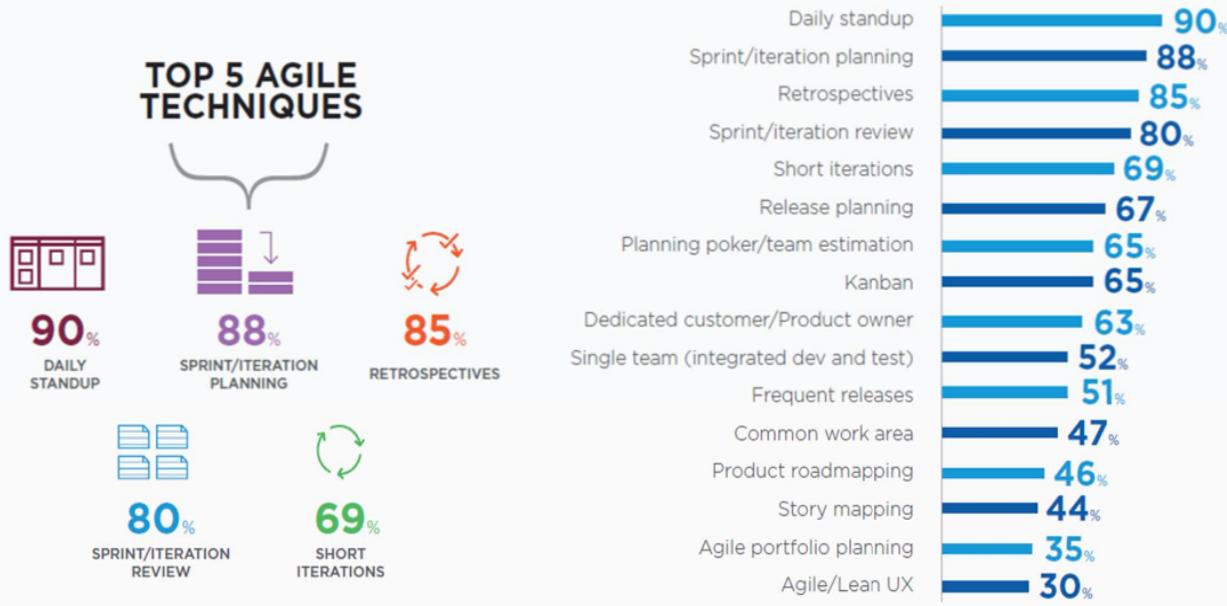
By implementing agile, respondents cited seeing improvements in the following areas:



Tècniques Agile més utilitzades

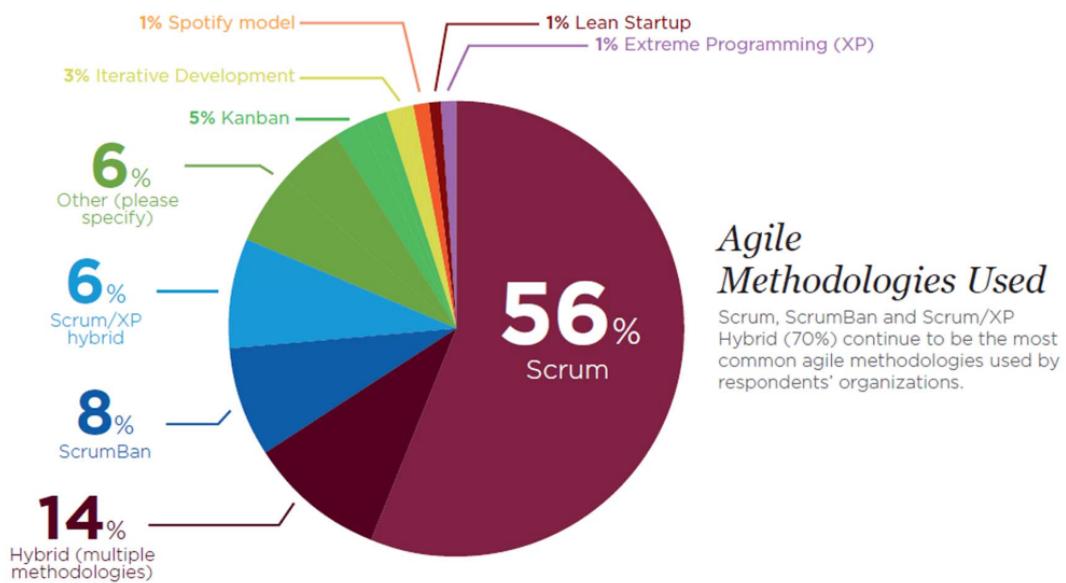
Agile Techniques Employed

From 2016 to 2017, the use of Kanban grew from 50% to 65%; product roadmapping increased from 38% to 46% and portfolio planning went from 25% to 35%.



*Respondents were able to make multiple selections.

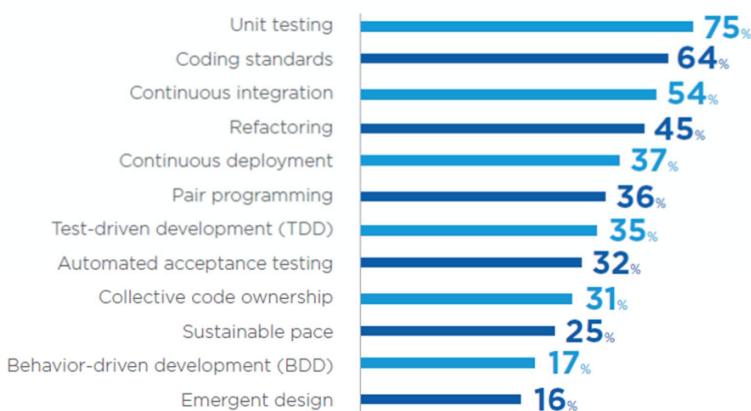
Metodologies Agile



Pràctiques d'enginyeria més usades

Engineering Practices Employed

This year's survey demonstrated an increased use of coding standards (64% compared to 56% last year) and the use of continuous integration and refactoring were cited less as practices used.



*Respondents were able to make multiple selections.

Incepció de projectes àgils

Entendre el problema

- Establir l'àmbit del projecte i condicions de frontera
- Determinar els casos d'ús i els seus escenaris principals
- Visualitzar una arquitectura candidata en base a alguns escenaris primaris
- Estimar el cost i planificació temporal
- Identificar possibles riscos
- Preparar l'entorn de treball del projecte

Concepte

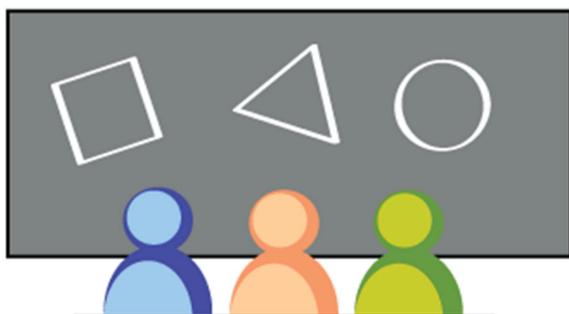
- Agile Inception, Inception Deck o Inception
 - Conjunt de preguntes i exercicis
 - per a alinjar i establir les expectatives
 - de totes les persones involucrades en un projecte

- Objectius addicionals
 - Reduir incerteses
 - Explicar riscos
 - Acordar expectatives
- Reunió de totes les persones involucrades
- Durant hores (fins a 2 dies)
- Activitats
 - Per què estem aquí?
 - Mostra la solució
 - Quant durarà?

Motivació



We are all in agreement then.



Oh.

- Iniciació agile



What if we did this ...



Ah!

Per què estem aquí?

Exemple: Projecte de construcció d'una sistema informàtic per a una constructora

Why are we here?



PER (destinataris del producte)

QUI (necessitats usuaris)

EL (nom de producte)

ÉS UN (categoria de producte)

QUE (major benefici, funcionalitats clau)

A DIFERENCIA (pràctica actual, competència)

EL NOSTRE PRODUCTE (major diferenciació)



The Elevator Pitch

- For [individual construction teams]
- who [need track road access on the construction site],
- the [Road Closure System (RCS)]
- is a [safety communication tool],
- that [informs crews when roads will be closed].
- Unlike [the current paper-based system]
- our product [is web based and can be accessed by all contractors anywhere anytime].

- Dissenyar la caixa del producte

Ràpid! Tens 30 segons per impressionar-me

RCS

"The safest way to control traffic on your construction site."



- ✓ Instant access - anytime, anywhere
- ✓ Headache free setup and administration
- ✓ Fewer accidents
Fewer missed deliveries

Your friendly onsite traffic cop



- Crear una NOT list

IN SCOPE	OUT OF SCOPE
Administer road closure requests Display road closures by date Email distribution lists Basic admin Basic search Reporting	Material tracking Delivery scheduling Automated traffic routing
UNRESOLVED	
Level of map precision Mapping technology to be used Source of satellite imagery	Integration with legacy construction work permit system

- Coneix els teus veïns

Your project community

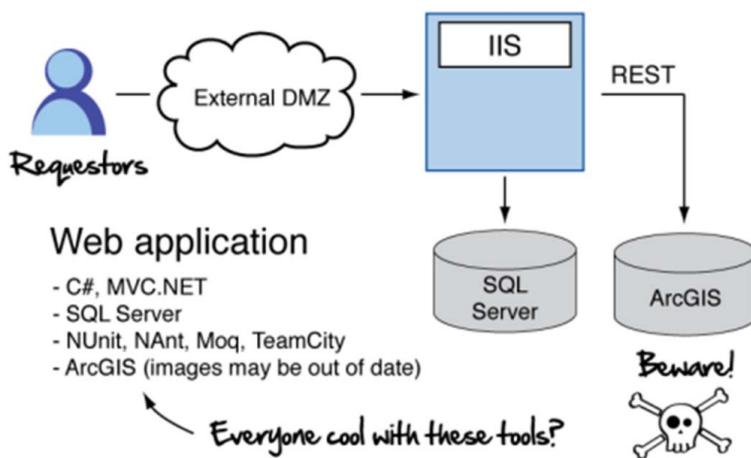


... is always bigger than you think !

Mostra la solució

- Entre els tècnics descriure diagrames d'arquitectura i assegurar que tots estan d'acord

Technical architecture



- Què no ens deixa dormir per la nit?

Stuff we lose sleep over

- Availability of the customer.
- Not having a dedicated dev/test environment.
- Not have a co-located team.
- Not having full time team members.
- Having ten analysts and two developers.
- Betting the farm on a technology that we have no in-house expertise around.



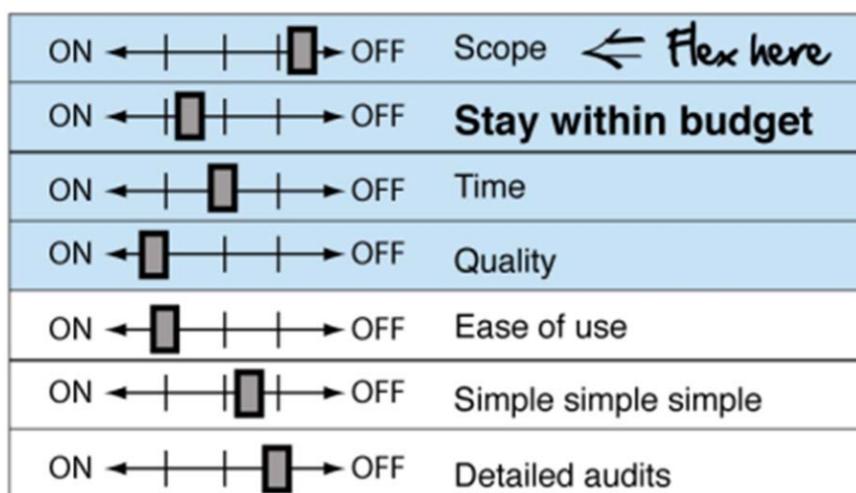
<insert favorite project risk here>

o

- Establiment d'expectatives. Forces en conflicte. Què podem reduir?
- El pressupost i les dates acostumen a ser fixes, la qualitat no és negociable.



Trade-off sliders



o

Quant durarà?



The first release



3 people, 3 1/2 months, \$250K



Històries d'usuari

Com a client habitual,
vull poder consultar les últimes novetats de la botiga

Objectiu

- Requisits del software = Problema de comunicació
 - Desenvolupadors del software
 - Persones per als que el desenvolupen
- Cal gestionar els recursos disponibles
- L'equilibri és crític
- El domini de qualsevol part implica dificultats importants
 - Què passa quan domina la part de negoci?
 - Què passa quan domina la part de desenvolupament?
- Planificació difícil

Solució

- Prendre decisions amb la informació disponible
- Fer-ho sovint durant el projecte

User stories: C + C + C

Card
Una frase descriu la User Story
Es poden afegir notes, estimacions, proves d'acceptació, etc.

Conversa
Els detalls es conversen amb el Product Owner

Confirmació
Les condicions d'acceptació que confirmen la Story s'ha desenvolupat correctament

Exemples

Plantilla	
Com a	<rol>
vull	<objectiu>
per tal de	<raó>

Com a alumne,
vull consultar la disponibilitat de places en els grups d'una assignatura

per tal de poder planificar la meva matrícula.

Com a alumne,

??? he d'indicar la sessió abans d'usar el sistema,

per tal de garantir la privacitat del meu inici de matrícula.

On són els detalls?

Com a alumne,

vull consultar la disponibilitat de places en els grups d'una assignatura

per tal de poder planificar la meva matrícula.

Detalls

Com a criteris d'acceptació?

Com a alumne,

vull consultar la disponibilitat de places en els grups d'una assignatura

per tal de poder planificar la meva matrícula.

- Es mostra la disponibilitat de places en la llista d'assignatures
- Escollint una assignatura es mostra el detall de disponibilitat de cada grup

Com històries més detallades

Com a alumne,

vull consultar la disponibilitat de places en els grups d'una assignatura

per tal de poder planificar la meva matrícula.

EQUIVAL A:

1

Com a alumne,

vull veure les places disponibles de cada assignatura de la qual em podria matricular a la pantalla inicial

per tal de tenir la informació disponible d'un cop d'ull

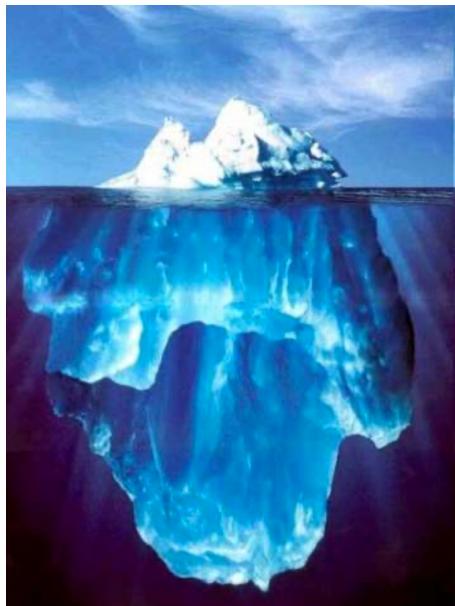
2

Com a alumne,

vull subscriure a una assignatura d

per tal de rebre notificacions freqüents de la disponibilitat de grups en aquesta assignatura

Històries d'usuari i Product backlog



Història d'usuari

Èpica

Tema

Històries massa grans

Dividir-les en històries més petites

- Un exemple?

Story-writing workshops

- Incloure l'equip, clients, usuaris, etc.
- Brainstorm
- Objectiu: Escriure com més històries millor
 - Inicialment, buscar Èpiques
 - Després, refinar
- No prioritzar
- Un possible procés
 - User Role Modeling
 - Brainstorm
 - Organitzar, consolidar, refinar
 - Brainstorm històries d'usuari (per rol d'usuari)
 - Organitzar, consolidar, refinar
- Altres eines
 - Mind mapping
 - Esbossos de la interfície d'usuari (ex. A mà alçada)

Invest

Independents

Negociables

Valuoses

Estimables

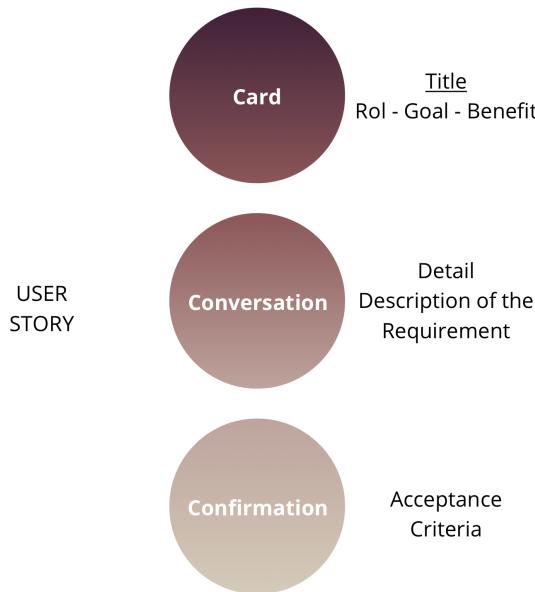
Sized Appropriately

Testables

Criteris d'acceptació

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



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

Com han de ser?

- Han d'expressar intenció, **no una solució**
 - “L'usuari pot escollir un compte” enllot 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)
- 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

Exemple

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

Com a comprador

Vull poder-me donar d'alta

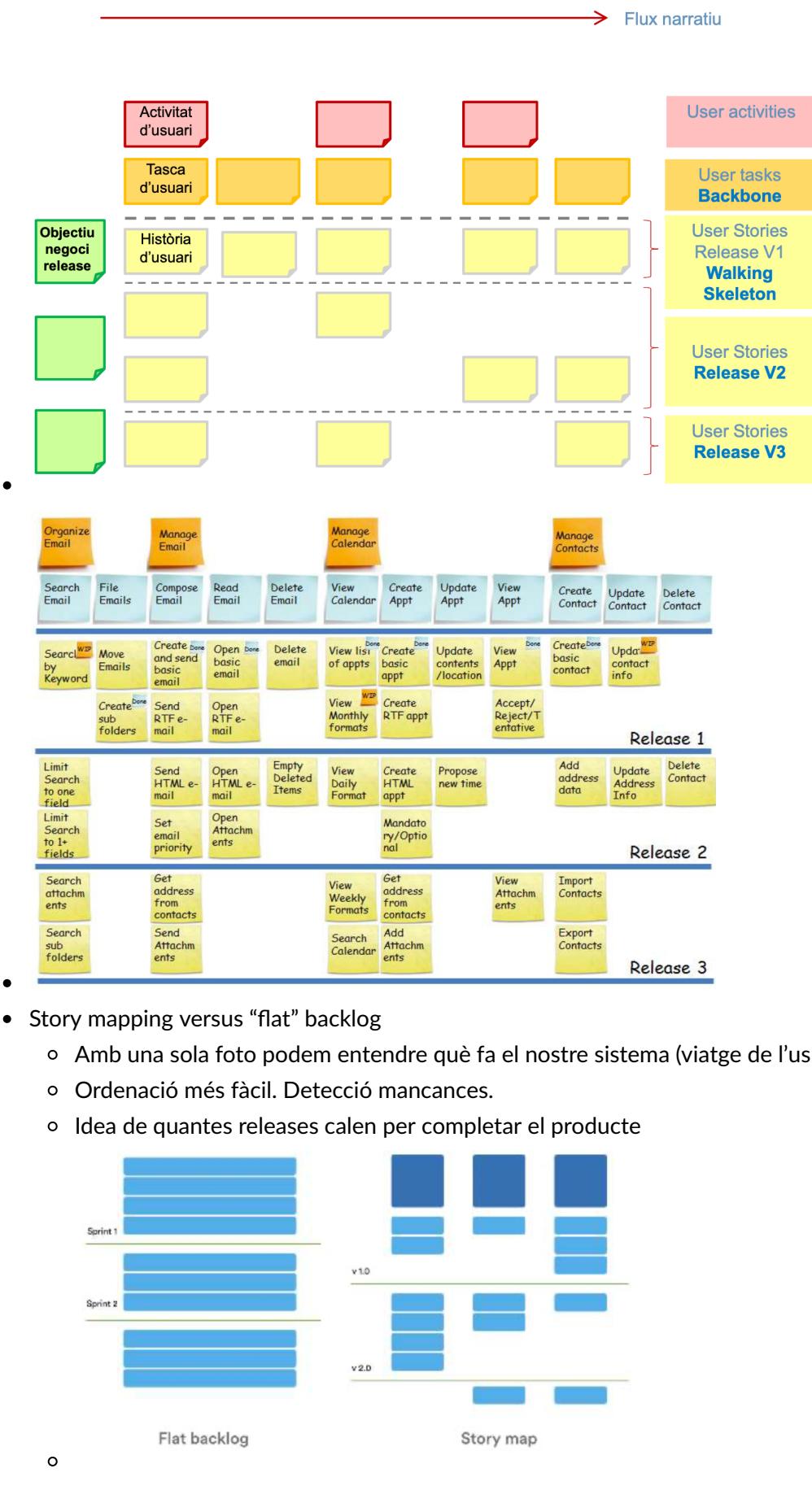
Per a què disposi d'informació ja predeterminada a l'entrar

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 (!,@,#,\$,%)
- ...

Tècnica definició producte - Story Mapping

- Serveix per identificar i organitzar les històries
- Organitza el Product Backlog en 2 dimensions
- Ajudar a organitzar la discussió sobre l'experiència d'usuari amb el nostre producte
 - Per què es construeix el producte
 - Entendre els usuaris i els seus objectius
 - Donar estructura al mapa d'històries d'usuari
- Forma senzilla d'explicar una història i trencar-la en parts
- La Visió s'aconsegueix amb objectius que són completats amb activitats
- Per completar aquestes activitats els usuaris necessiten realitzar tasques
 - Objectius > Activitats > Tasques > Històries d'usuari
- Priorització de cada columna sota activitat per valor (MoSCoW)



> Com crear un story mapping

1. Tasques usuari = backbone

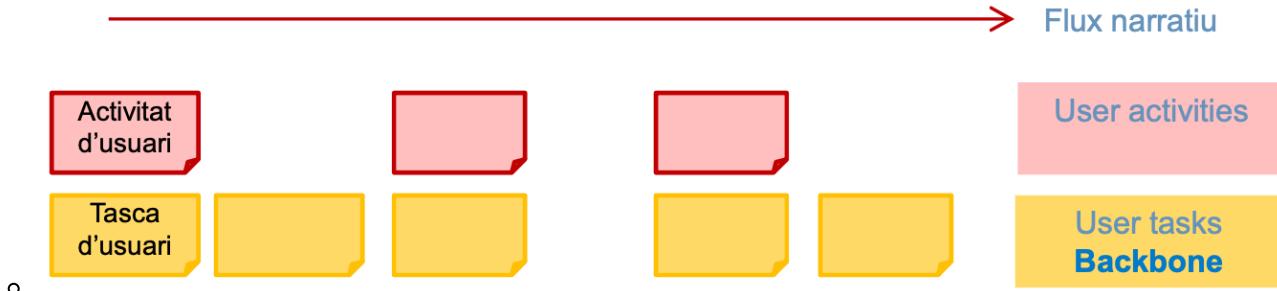
- Recol·lir les tasques principals dels usuaris i descriure de forma individual cadascuna en un post-it

- Cada individu llegeix el seu post-it en alt i el col·loca en el panell en front del grup. Si està duplicat no posar-lo

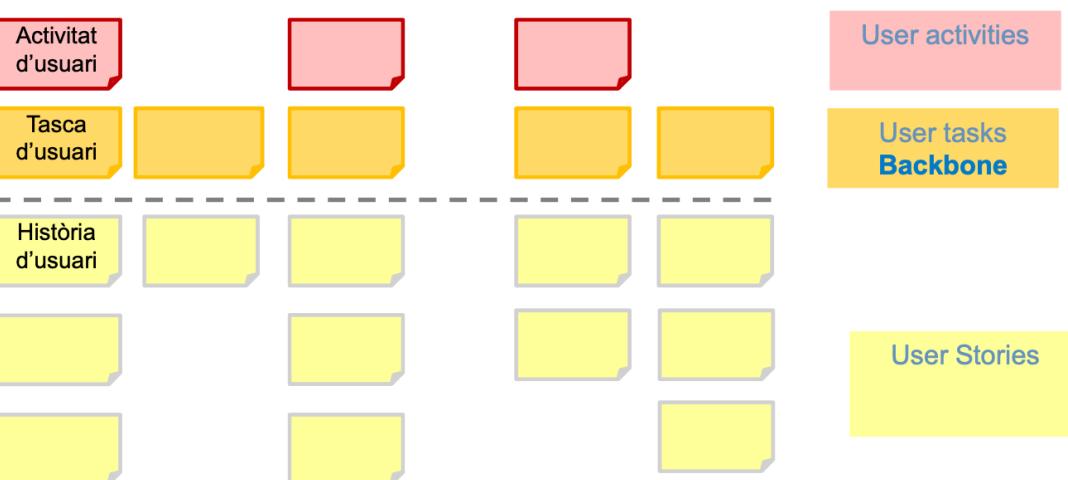


2. Activitats usuari

- Les tasques que semblin similars i succeeixin en el mateix moment agrupar-les sota un post-it amb un altre color i posar-li nom
 - Col·locar els grups d'esquerra a dreta en l'ordre que un usuari completaria habitualment les tasques

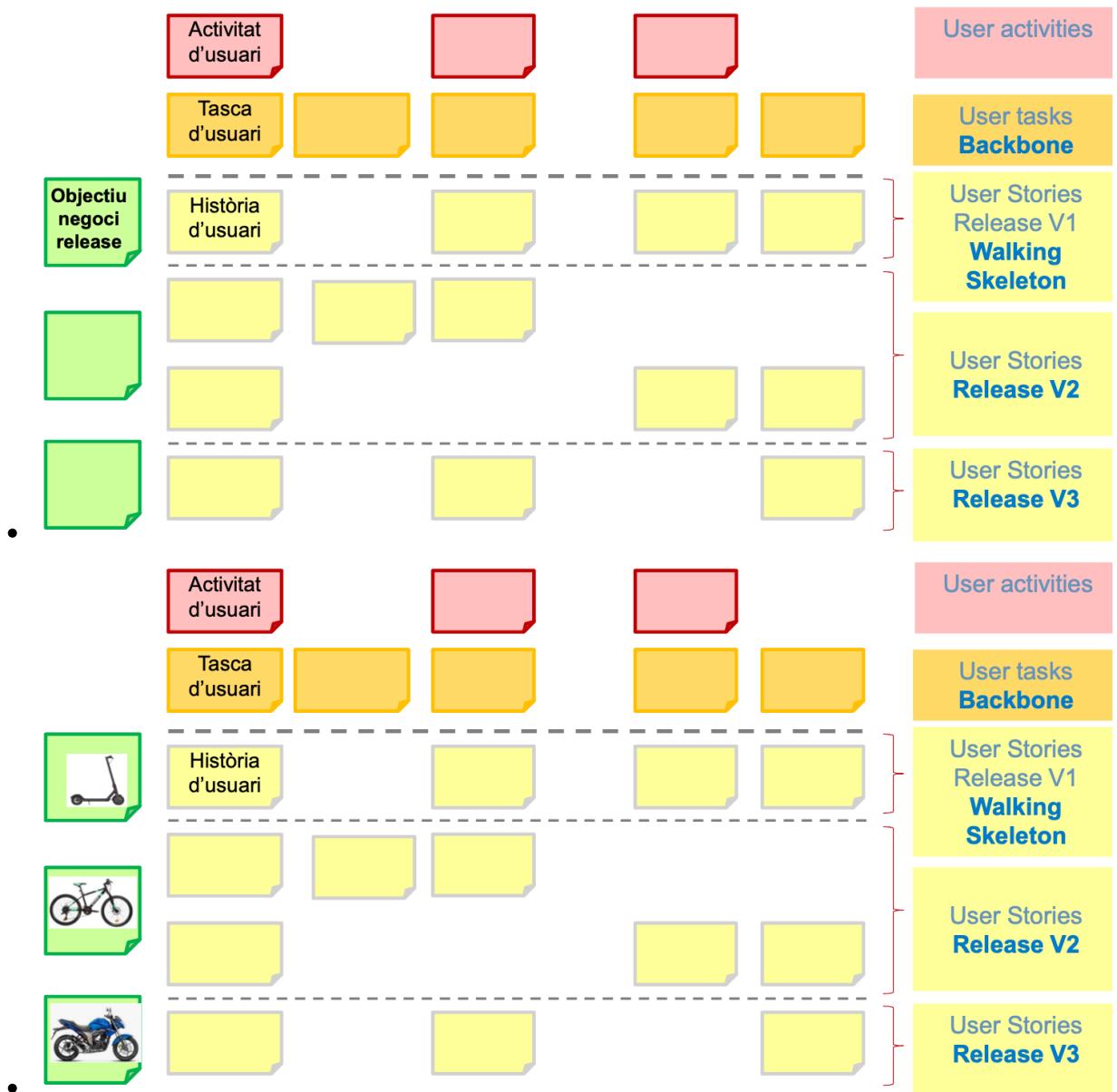


3. Afegir més històries d'usuari detallades, sota de cada tasca d'usuari (almenys una per tasca)



- Algunes consideracions
 - En format post-its per manca d'espai es pot evitar el format 'Com a ... Vull ... Per tal de' i posar sols el títol (facilita també la seva lectura) i usar un verb (Composar Correu, Afegir Usuari,...)

4. Estructurar en línies de versions el mapa preguntant com prioritzar les històries als usuaris (d'amunt a avall)



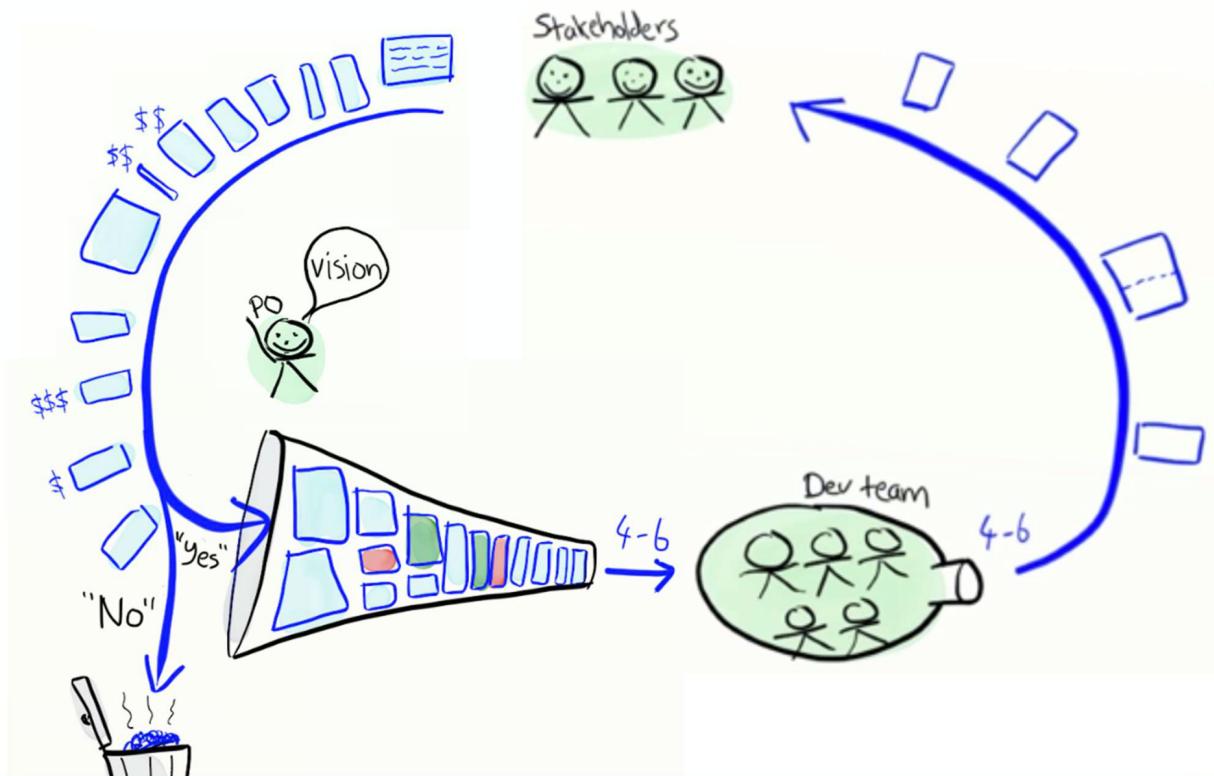
Estimació i planificació àgils

Estimació

def.

El principal propòsit de l'estimació del programari no és **predir** un resultat del projecte, és determinar si els objectius són realment **realistes** per poder controlar el projecte i aconseguir-ho.

Objetiu



Com fer planificació àgil?

- Estimar quant trigaré a llegir ...



Mida	→	Durada
------	---	--------

Unitats

- Es pot utilitzar qualsevol mètrica segons les necessitats concretes. Les més habituals són:
 - story Points

- Hores o dies ideals
- Scrum no dóna una manera concreta de fer les estimacions però es poden seguir alguns passos perquè l'estimació sigui el millor possible:
 - Estimar el temps disponible de l'equip de forma realista
 - Estimar cada element priorititzat del Product backlog correctament

Story Points

- Un Story Point és una mesura utilitzada en projectes Scrum
- S'utilitza per estimar la “dificultat” d’implementar un cas d’ús, tasca, etc
- En general, s’utilitzen les següents mètriques:
 - 1,2,4,8,16 o
 - X Small, Small, Medium, Large, Extra Large
- És una mesura abstracta i no té una correspondència directa a hores o dies però serveix per ajudar a quantificar l'esforç

Punts de Zoo

- Estimar el pes d'un mascle adult de:
 - Lleó
 - Cangur
 - Rinoceront
 - Ós
 - Girafa
 - Goril·la
 - Hipopòtam
 - Tigre

Hores ideals

- A l'hora d'estimar el temps d'un membre de l'equip en una tasca, no se li poden assignar totes les hores diàries de la seva jornada:
 - Es pot perdre temps en reunions (planejades o no)
 - Es pot perdre temps contestant correus o el telèfon
 - Es pot detectar algun error introduït anteriorment que s'ha de corregir
 - etc.
 - I també es perd temps, per exemple, fent el cafè
- Per a una estimació més realista, s'ha d'utilitzar el temps de la jornada laboral menys el temps dedicat a "altres tasques"

Estimacions

- Estimar el temps de cada element priorititzat del Product backlog:
 - És molt difícil estimar correctament una tasca si no s'ha fet mai
 - Pot ser útil involucrar gent amb més experiència en tasques similars
 - Tots els membres de l'equip han de participar en l'estimació
 - En general, a partir del 2n o 3r Sprint les estimacions cada vegada seran millors
- Els passos a seguir per estimar un element del product backlog podrien ser:
 - Entendre clarament el que es refereix el cas d'ús o requisit

- Fer totes les preguntes necessàries al product owner per tenir-lo totalment clar
- Dividir-lo en múltiples tasques (segons Scrum idealment cada tasca hauria de poder completar-se amb entre 4 i 20 hores)
- Estimar cada tasca individualment amb l'ajuda i col·laboració de tot l'equip
 - Un bon mètode pot ser utilitzar el “Planning Poker”



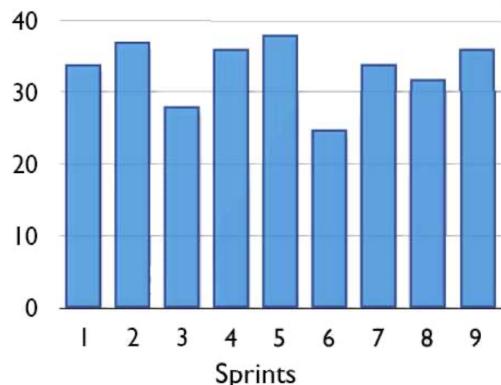
- Un cop realitzada l'estimació i començat l'Sprint, no es pot afegir més feina o canviar el treball assignat a l'equip
- Si no s'ha estimat bé i queda feina pendent, es passa al següent Sprint
- Els equips han d'auto-organitzar-se i hi ha d'haver transparència en la informació entre els membres

Velocitat

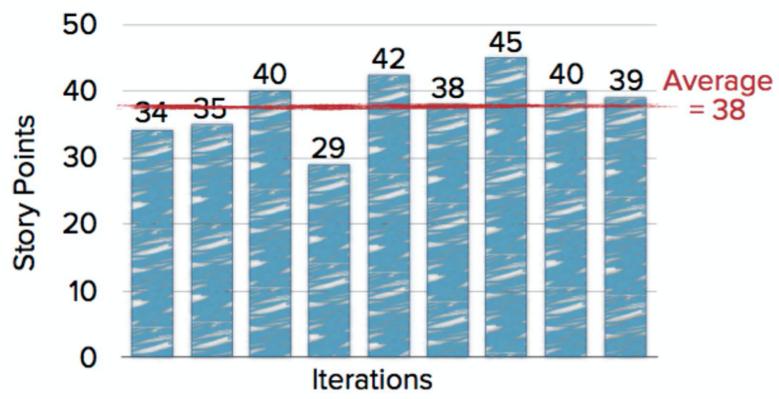
Velocitat = **Unitats de treball** que pot fer l'**equip** en una iteració

Suma total punts història completats durant una iteració

- Exemple:
 - (3 històries completades iteració * 5 punts) +
 - (2 històries completades iteració * 3 punts) +
 - (3 històries començades però no acabades * 8 punts)
 - = 21 punts (**velocitat**)
- La velocitat és la quantitat de treball completat a cada iteració

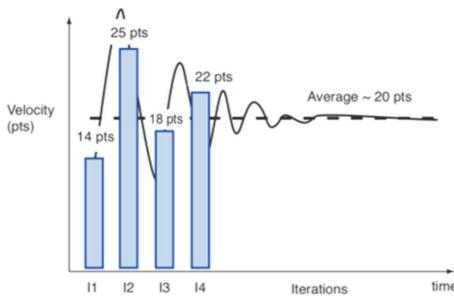


- S'utilitzen les mateixes unitats que per a les estimacions del product backlog
- Com utilitzar informació històrica?
 - Calcular un interval de confiança
 - Descartant algunes iteracions de cada extrem

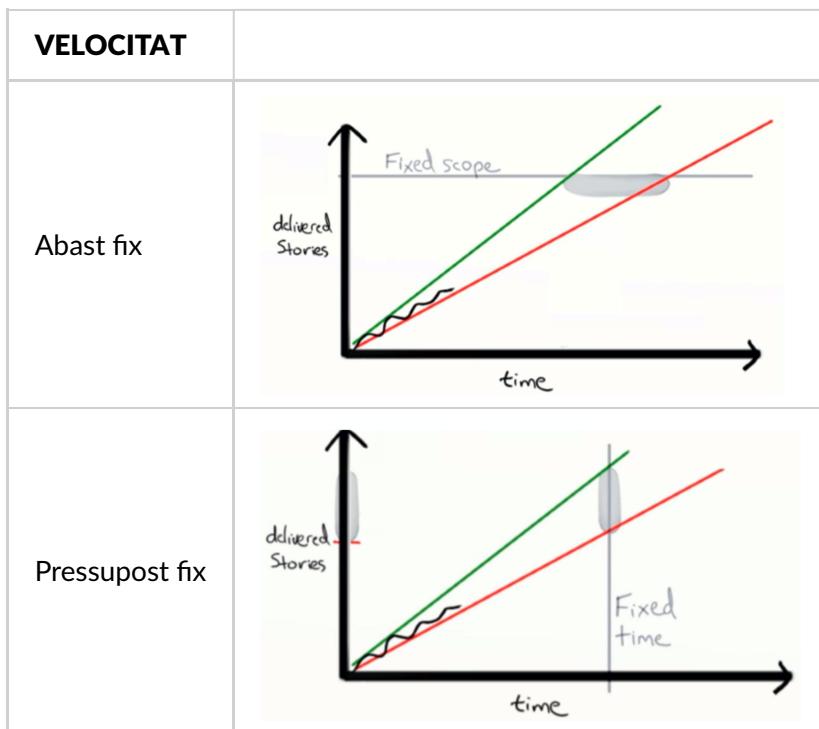


- Es necessiten varíes iteracions per estabilitzar-se
- No usar per mesurar el progrés o comparar equips

Velocity will vary



- Com utilitzar informació històrica?
 - Extrapolar amb els valors en els dos extrems del rang resultant
 - Velocitat històrica: 27 a 37
 - Amb un abast fix: Estimar data de finalització
 - Amb una data de lliurament fixa: Estimar abast
- Rangs
 - Ho tindrem, Podríem tenir-ho, No ho tindrem



Acceptació i control de qualitat en projectes àgils

Història d'usuari – Recordem...

- 3 seccions
 - **Card:** Identificació història
 - (Asa-Iwant-Sothatotextpla). 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

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

Com han de ser?

- Han d'expressar intenció, **no una solució**
 - "L'usuari pot escollir un compte" enllloc 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)
- 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

Exemples

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

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 (!,@,#,\$,%)

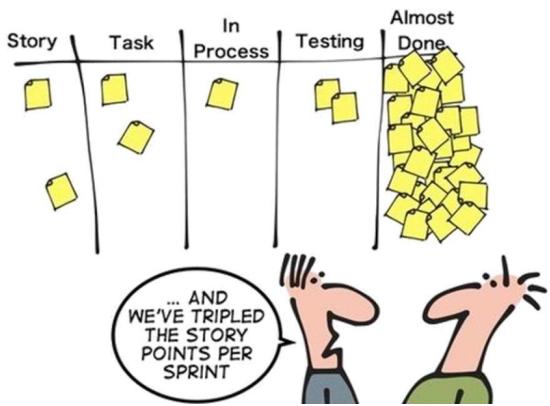
> Definició de Llest (Definition of Ready, DoR)

- Definida per l'equip Scrum
- Quan un ítem està preparat per entrar al sprint amb un risc assumible
- Inclou valor negoci que permet prioritització
- 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.

> Definició de Fet ()Definition of Done, DoD)

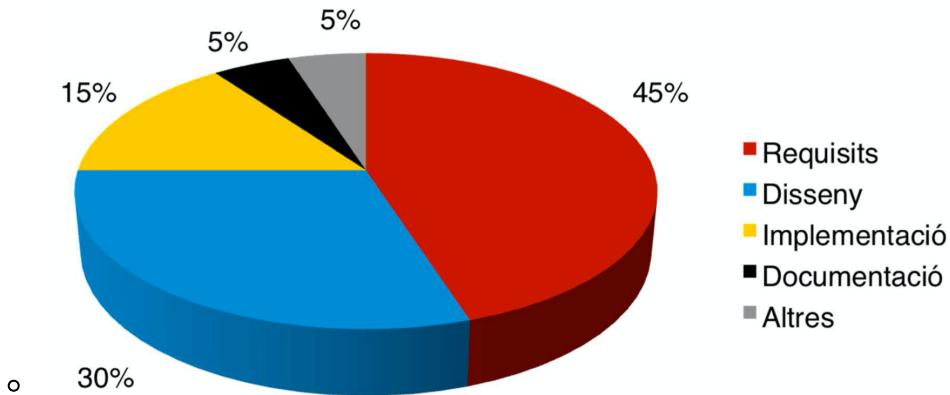
- Construir un enteniment comú dins l'equip sobre qualitat i completenessa
 - Quins mínims de qualitat i lliurables s'han de complir en tots els objectius/requisits que s'aniran acceptant durant cada iteració?
 - Com sap l'equip que la feina està feta?
- Tenir sempre un producte “potencialment entregable”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



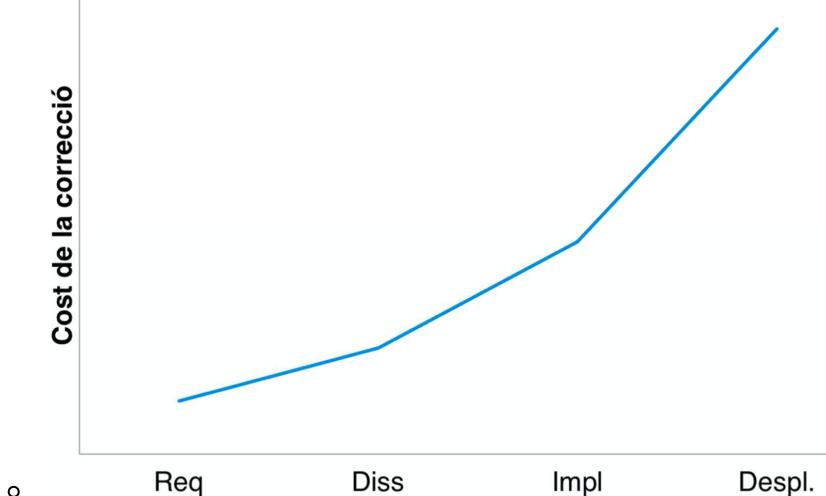
- Nivells
 - DoD d'una història d'usuari o ítem del product backlog
 - DoD de un sprint
 - DoD de una release
 - DoD d'una release a l'entorn d'integració
 - DoD d'una release a l'entorn de producció
 - ...
- Història d'usuari
 - Per qualsevol canvi al codi, s'ha escrit una prova que ho verifica
 - Proves 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
- 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
- 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
- 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*

Proves àgils

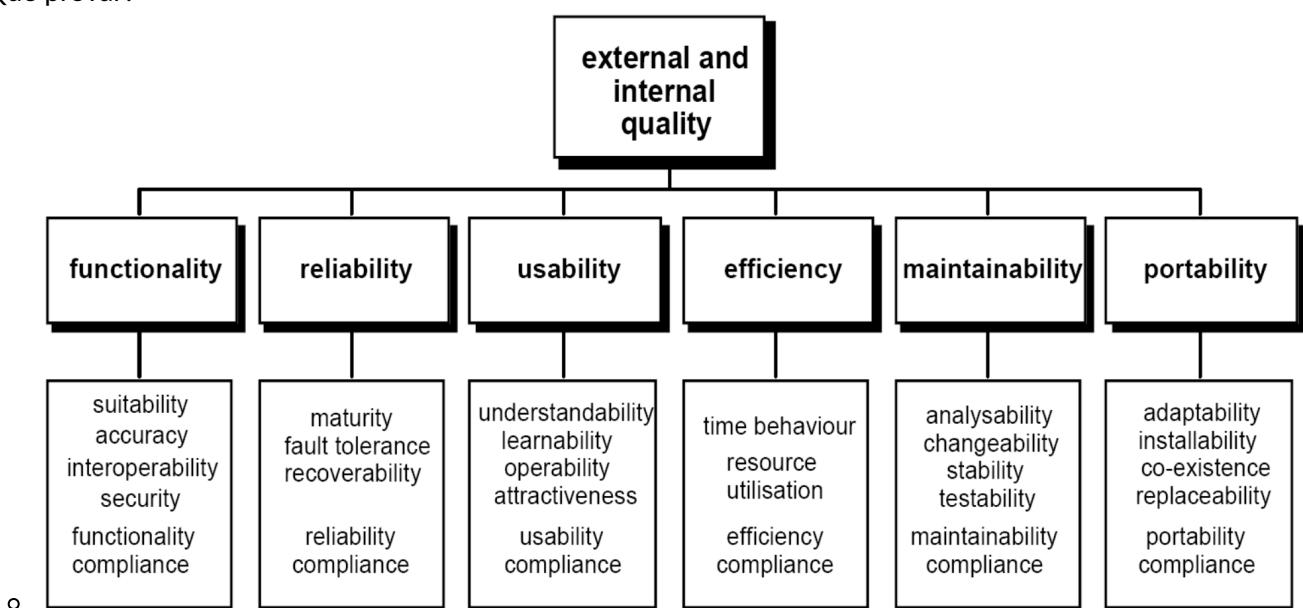
- Prova:
def.
Activitat dirigida a l'avaluació d'un atribut o capacitat d'un sistema software amb l'objectiu de determinar si satisfa els requisits [Hetzell]
- Motivació
- Defecte (bug)
 - Diferència entre el comportament del software i el comportament especificat
 - Diferència entre el comportament del software i el comportament esperat
- Causes dels defectes de software?
 - L'especificació canvia → pot ser que els tests no passin perquè ha canviat, s'han de tornar a fer
- 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 \$
- 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
- 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
- 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
- Origen dels defectes



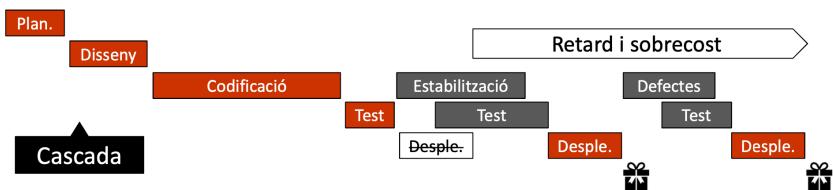
- Cost de la correcció de bugs



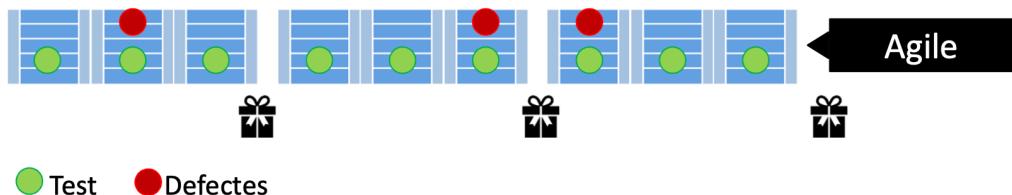
- Tipus de proves
 - Estàtiques (verificació) / dinàmiques (validació)
 - Unitàries / d'integració / de sistema / ...
 - Alfa / Beta / ...
 - En parlem més tard:
 - Manuals / Automatitzades
 - D'acceptació / de regressió / ...
- Què provar?



> 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.



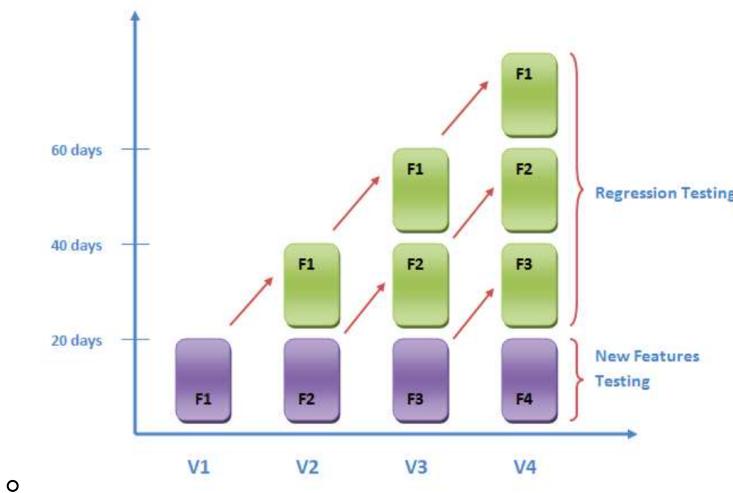
> Principis

Shifting Focus	<p>"Fear not! I'll protect you!"</p> <p>...from being the last line of defense...</p> <p>...to providing information, feedback, and visibility</p>
Les proves no són una fase	<p>Anàlisi → Disseny → Construcció i proves</p>
Tots proven	
Reducció latència feedback	Provem de forma primerenca i freqüent
Representen expectatives	Si el resultat no és l'esperat, no estan clares les expectatives, parlar amb els stakeholders o el product owner

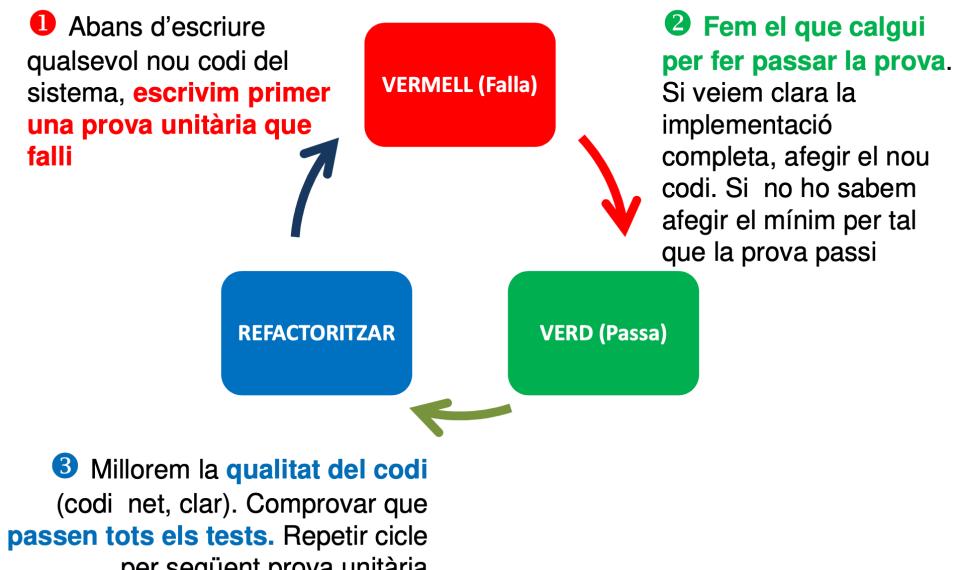
Mantenir el codi net	
Reduir documentació de proves	Llistes de verificació, Essència de la prova enlloc registre exhaustiu incidències, Eines lleugeres
Part del Done	

> Proves de regressió

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



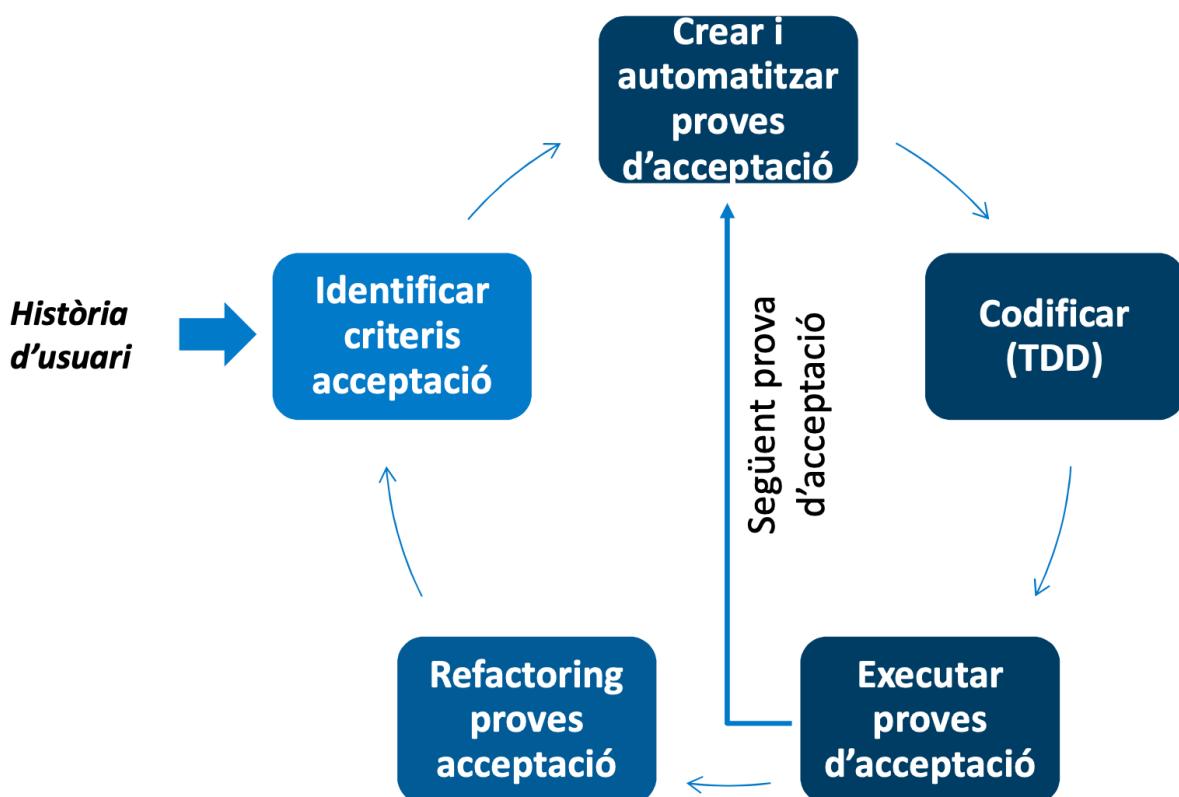
> Test Driven Development (TDD)



> Especificació per l'exemple

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

> Acceptance Test Driven Development

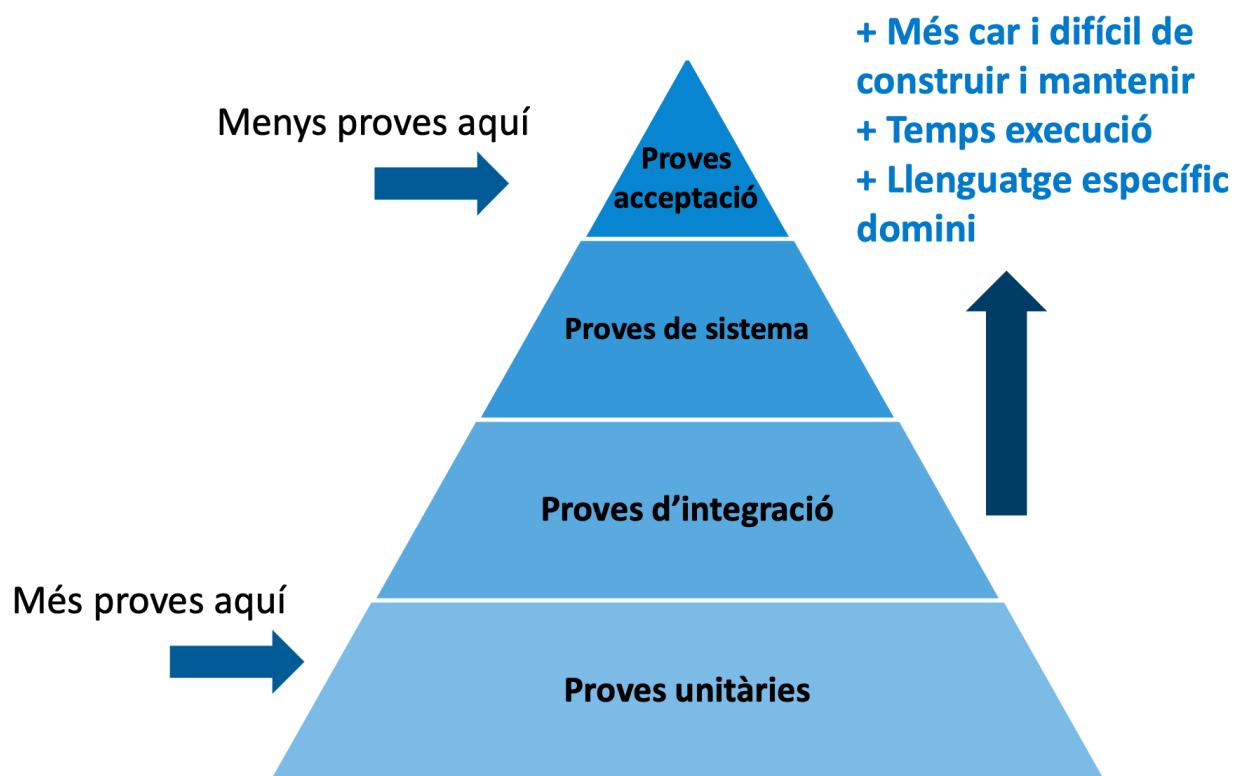


> Behaviour Driven Development

- 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**

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

> Piràmide d'automatització

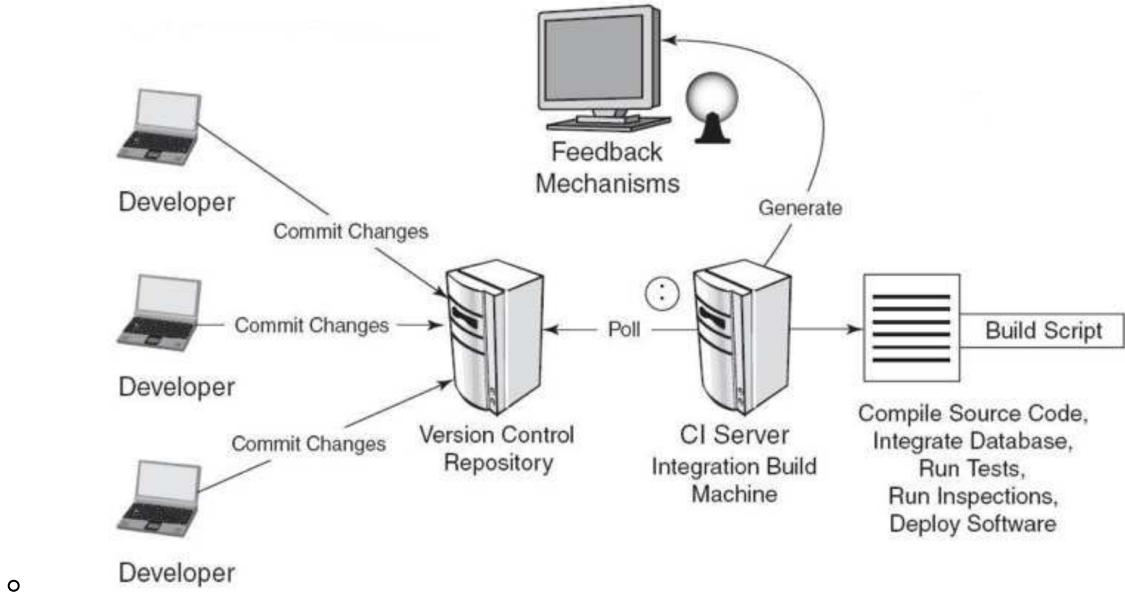


> Integració contínua

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.

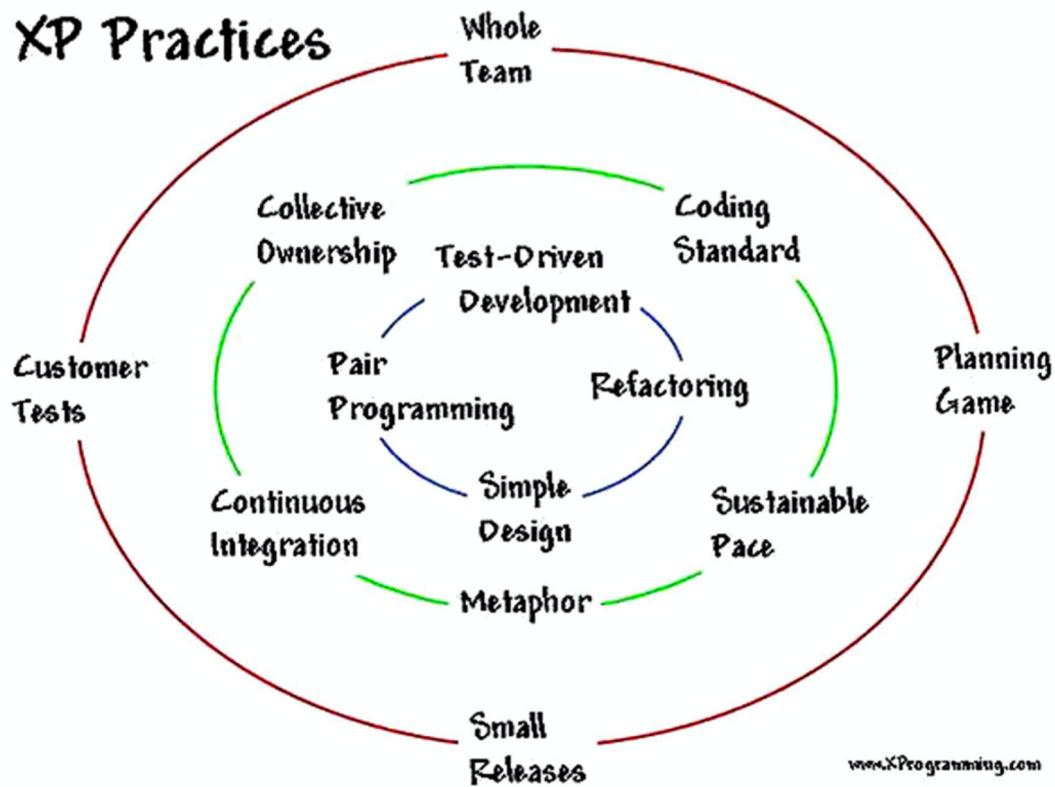
Each integration is verified by an automated build (including test) **to detect integration errors as quickly as possible.**

- El cicle de build i feedback continu de la qualitat



eXtreme Programming

XP Practices



Valors

- Comunicació
- Simplicitat
- Feedback
- Coratge
- Respecte

Comunicació

- Gestors, clients i tècnics formen part del mateix equip
- Comunicació cara a cara diària

- Un o més clients assignats com part de l'equip
- Less is more

Simplicitat

- DTSTTCPW: Do The Simplest Thing That Could Possibly Work
- YAGNI: You Ain't Gonna Need It

Feedback

- Al minut: Tests unitaris
- Al dia a dia: Les noves històries s'estimen quan s'escriuen
- A iteracions: Dues o tres setmanes

Coratge

- Per tirar codi desenvolupat a les escombraries
- Per canviar el disseny i l'arquitectura d'un projecte en desenvolupament en qualsevol moment (mitjançant refactoring)
- Però sempre d'acord amb els altres valors (coratge, no temeritat)

Respecte

- Cap als altres membres de l'equip, tècnics o de negoci
- Cap al projecte
- Cap a un mateix com a professional

Normes

- (*Pràcticament Scrum*) Planificació
- (*Pràcticamente Scrum*) Gestió
- Disseny
- Programació
- Proves

Planificació

- Històries d'usuari
- Planificació de lliuraments
- Lliuraments petits
- Iteracions
- Planificació de cada iteració

Gestió

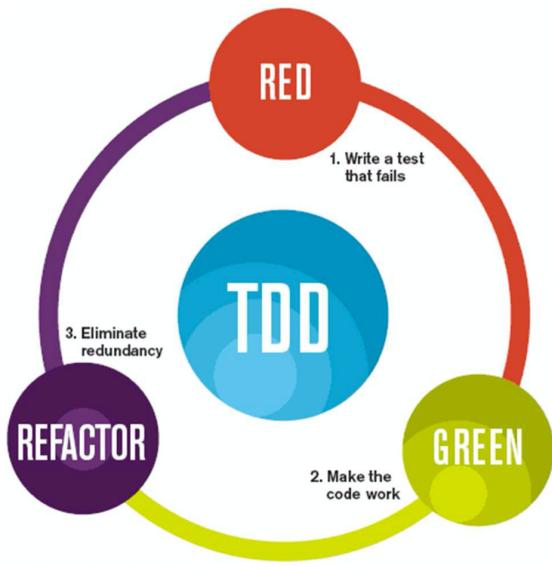
- Espai de treball obert
- Ritme sostenible
- Reunió diària de peu
- Velocitat
- Mou a la gent
- Arreglar XP

Disseny

- **Simplicitat**
 - DTSTTCPW
 - Codi TUBE
 - Testable
 - Understandable
 - Browsable
 - Explainable
- **Metàfora del sistema**
 - Estableix el vocabulari
- Targetes CRC per a les sessions de disseny
 - Class, Responsibilities, Collaboration
- Solucions spike per reduir el risc
 - Desenvolupar una prova per solucionar un dubte tecnològica
 - Timeboxed!
- No afegir funcionalitat de forma primerenca
 - Quanta de la funcionalitat afegida per un tècnic al que li demanen s'acaba fent servir?
 - 10%
 - Afegir flexibilitat més enllà del necessari en realitat afegeix complexitat
- Refactoring
 - Green to Green
 - Sense canvis de comportament

Codi

- **Tot l'equip**
 - El client no ajuda a l'equip sinó que forma part de l'equip
 - Escriu històries d'usuari, les prioritza, proporciona detalls durant el desenvolupament, realitza proves funcional
- **Adequació a estàndards**
- Pair Programming
- TDD: Desenvolupament guiat per les proves



○ The mantra of Test-Driven Development (TDD) is "red, green, refactor."

- Integració seqüencial
 - Només una parella integra
- Integració freqüent
 - Un únic repositori compartit
 - Commit al repositori compartit cada poques hores
- Ordinador d'integració
 - Un ordinador només per integració
- Propietat col·lectiva del codi



Proves

- Unitàries
 - Proves de tot el codi (excepte el més trivial)
 - Les proves formen part del repositori de codi
 - Totes les proves funcionen com a proves de regressió
- No lliurar codi si no passa totes les proves
- Si es troba un error, primer s'escriu una prova
- S'executen proves d'acceptació sovint
- El resultat d'executar les proves es publica

eXtrema??

- Kent Beck: És difícil tenir massa d'alguna cosa bona en desenvolupament de programari
 - La implicació del client és bona

- Converteix al client en un membre full time de l'equip
 - La comunicació és bona
 - Tothom al mateix lloc, pair programming, clients com part de l'equip, etc.
 - Les proves són bones
 - Proves unitàries de tot el codi
 - Proves d'acceptació per documentar requisits
 - La revisions de codi són bones
 - Revisar contínuament i en temps real (pair programming)
 - La integració de codi és bona
 - Integrar contínuament de forma automatitzada
 - Les iteracions curtes són bones
 - Fer iteracions d'1 o 2 setmanes
-

Pràctiques de l'equip àgil

Kanban

Scrum

Scrum: Una introducció



Processos iteratius i incrementals

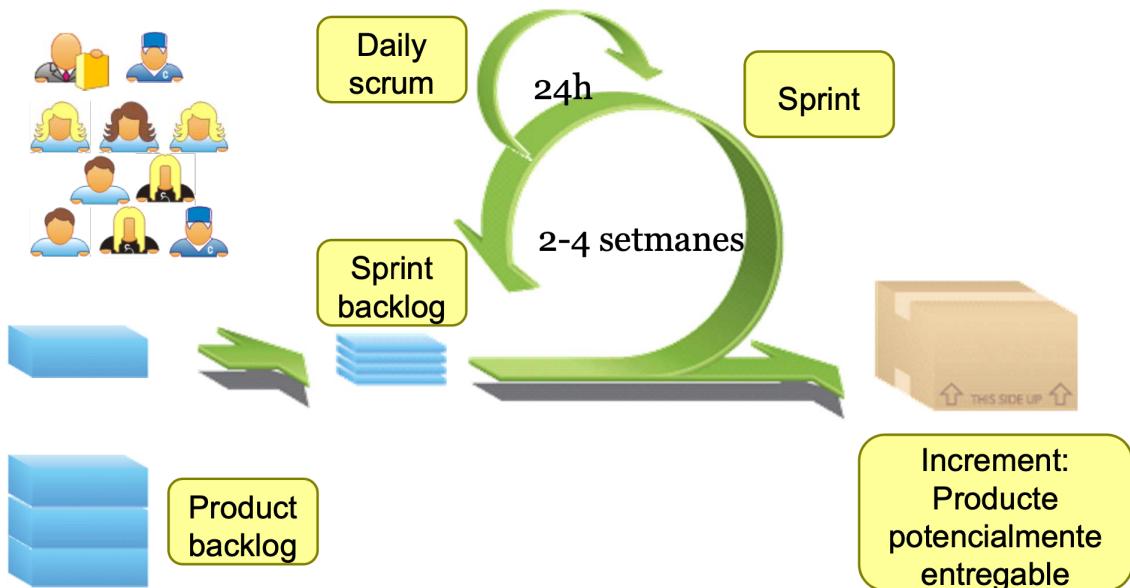
- L'enfocament de *cursa de relleus* en el desenvolupament de productes ... pot entrar en conflicte amb els objectius de màxima velocitat i flexibilitat.
- En el seu lloc, un enfocament holístic o estil 'rugbi' - on un equip intenta anar a la distància com una unitat, passant la pilota cap endavant i cap enrere - poden servir millor als actuals requisits competitius.

Scrum



Característiques

- Equips auto-organitzats
- El producte evoluciona a través d'una sèrie de “**sprints**”
- Els requisits es capturen com els elements d'una llista anomenada “**product backlog**”
- No prescriuen pràctiques d'enginyeria concretes
- Utilitza regles generatives per crear un entorn àgil per al lliurament de projectes
- Un dels “**processos àgils**”



3 Rols + 4 Cerimònies + 3 Artefactes

Rols
Product owner
Scrum Master
Equip de desenvolupament

Cerimònies
Sprint planning
Daily scrum

Cerimònies
Sprint review
Sprint retrospective

Artefactes
Product backlog
Sprint backlog
Increment

> Rols i responsabilitats

Rols

En Scrum s'identifiquen dos tipus d'usuari:

- Usuaris que formen part del procés:
 - Client
 - Equip
 - Scrum Màster
- Usuaris que no formen part del procés però s'han de tenir en compte:
 - Altres stakeholders
 - mànagers
 - etc.

En general, en les diferents reunions que es veuran més endavant, només intervé el **primer grup** d'usuaris però no es descarta permetre assistir algun dels altres si és només d'oient

Responsabilitats

Les responsabilitats d'un projecte Scrum es divideixen en 3 rols:

- Product Owner (PO, Client)
- Development Team (DT, L'equip de desenvolupament)
- Scrum Master (SM)

Tots tenen una implicació activa en el desenvolupament del projecte

Product Owner (Client)

- Responsable de maximitzar el valor del producte
 - Representa els interessos de tots els stakeholders en el projecte i en el sistema resultant
- Funcions
 - Defineix les funcionalitats del producte
 - Decideix les dates i continguts de les versions
 - És responsable de la rendibilitat del producte (ROI)
 - Prioritza funcionalitats d'acord amb el valor de negoci
- Les seves decisions són respectades per tota l'organització

- Són visibles en forma de Product backlog
- Gestiona el Product backlog:
 - Expressa clarament els seus continguts
 - Ho mana per aconseguir els objectius i missions del projecte
 - Assegura el valor del producte produït
 - Assegura que sigui visible, transparent i clar per a tots
 - Assegura que el DT entén els ítems que conté

Development team

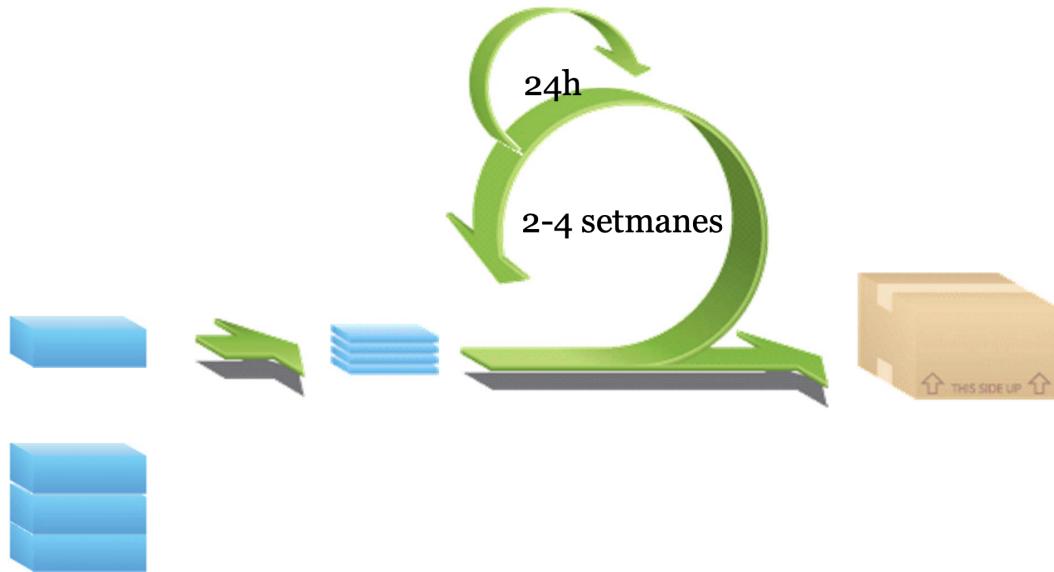
- Desenvolupa productes potencialment lliurables
- Característiques
 - Auto-organitzat
 - Crossfunctional
 - Sense títols ni categories
 - Un membre pot tenir punts forts i especialitats però tots responen com un de sol, com a equip
 - No hi ha sub-equips
- Mida: Entre 3 i 9 personnes
 - No canvia durant l'esprint

Scrum Master

- És el responsable del bon funcionament de tot el procés
- Característiques
 - Líder servent
- Serveis oferts al Product Owner
 - Trobar tècniques efectives de gestió del Product backlog
 - Comunicar clarament la visió, objectius i ítems del PB al DT
 - Ensenyar a crear ítems de PB concisos i clars
 - Entendre la planificació a llarg termini de producte en un entorn empíric
 - Entendre i practicar l'agilitat
- Facilitar els esdeveniments prescrits per Scrum segons sigui necessari
- Serveis oferts a l'equip de desenvolupament
 - Ajudar a aconseguir ser auto-organitzat i cross-functional
 - Ensenyar i liderar crear productes d'alt valor
 - Eliminar impediments
 - Coaching de l'equip en organitzacions en què Scrum no està plenament adoptat
- Serveis oferts a l'organització
 - Lideratge i coaching de l'adopció de Scrum
 - Planificació de la implementació de Scrum en l'organització
 - Ajudar a empleats i stakeholders a entendre i utilitzar Scrum
 - Causar el canvi que incrementi la productivitat de l'equip
 - Treballar amb altres Scrum Master

> Cerimònies

Esprints



Product backlog

Exemple de product backlog:

Backlog item	Estimate
Allow a guest to make a reservation	3
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run RevPAR (Revenue-Per-Available-Room) report	8
Improve exception handling	8
...	30

Sprint

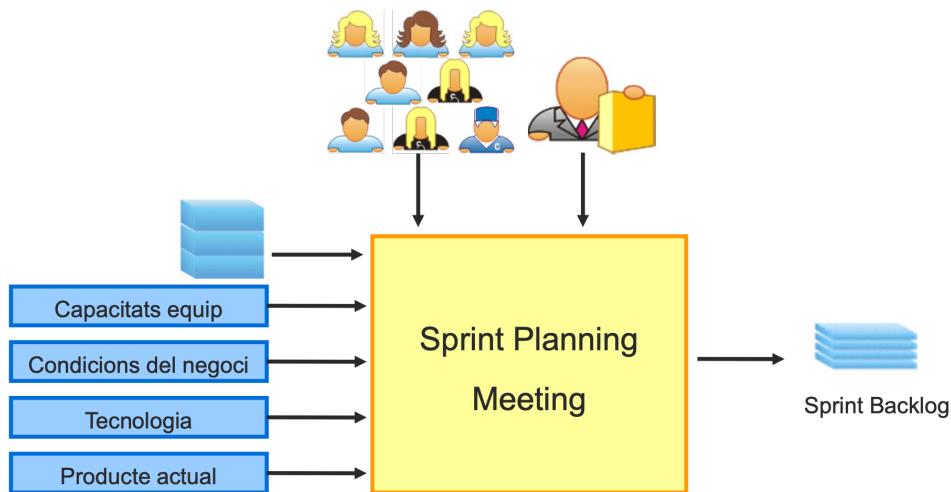
- Sprint Planning Meeting
- Desenvolupament
- Daily scrums
- Sprint Review Meeting
- Sprint Retrospective Meeting

STOP Canvis

Matisos

- Els requisits són vagues al principi
- Refinem els coneixements que tenim d'un requisit durant el treball diari
- Què és un canvi?

Sprint Planning Meeting



Sprint backlog

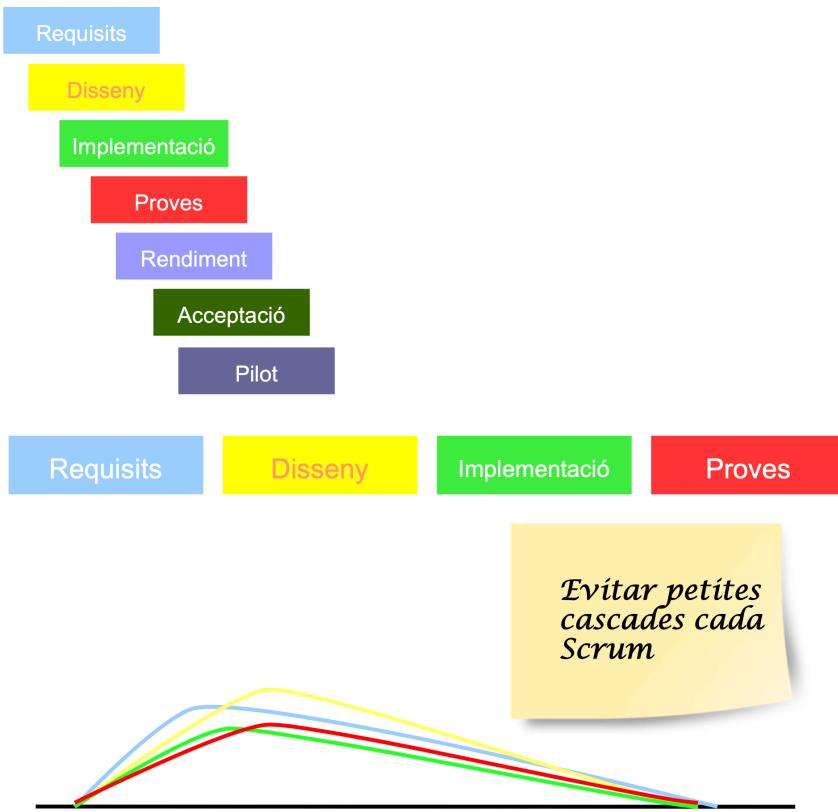
Objectiu de l'Sprint:

Tasks	Mon	Tues	Wed	Thurs	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	4	
Test the middle tier	8	16	16	11	8
Write online help	12				
Write the foo class	8	8	8	8	8
Add error logging			8	4	

Desenvolupament

- Fer tot el necessari per acabar el treball
- Portar com a mínim una entrada de backlog d'idea a implementació
- Podeu cercar consell, ajuda, ... fora, però ningú de fora pot donar instruccions, comentaris o intentar dirigir l'equip de desenvolupament
- Ningú pot canviar res del Product backlog durant el Sprint
- Si no es pot acabar tot el que s'ha acordat, treure funcionalitats a l'esprint actual (negociant amb el client)
- Si sobrarà temps, consultar al client per afegir funcionalitats al Sprint actual
- Responsabilitats:
 - Atendre cada dia al Daily Scrum
 - Mantenir el Sprint backlog actualitzat i visible per a tots els membres

Què significa acabar la feina?



Daily Scrum

- L'equip de desenvolupament sincronitza activitats i planifica el dia
 - Cada dia
 - 15 minuts
- 3 preguntes:
 - Què has fet des de l'última reunió Daily Scrum?
 - Què planeges fer fins al següent Daily Scrum?
 - Quins problemes has de puguir perjudicar el Scrum actual o al projecte?
- 2 preguntes addicionals:
 - Quines tasques haurien d'afegir l'Sprint backlog?
 - Què has après que sigui nou i rellevant per a algun membre de l'equip?
- Optimitza la probabilitat d'aconseguir l'objectiu de l'Sprint
 - L'entrenador replanifica l'Sprint després del Daily Scrum
 - L'entrenador és capaç d'explicar en qualsevol moment com s'auto-s'organitzarà per aconseguir l'objectiu del Sprint
- L'Scrum Master...
 - S'assegura que es faci ... però no el condueix
 - Ajuda a l'equip en el timeboxing
 - Assegura que no参与 ningú que no sigui DT

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
- 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

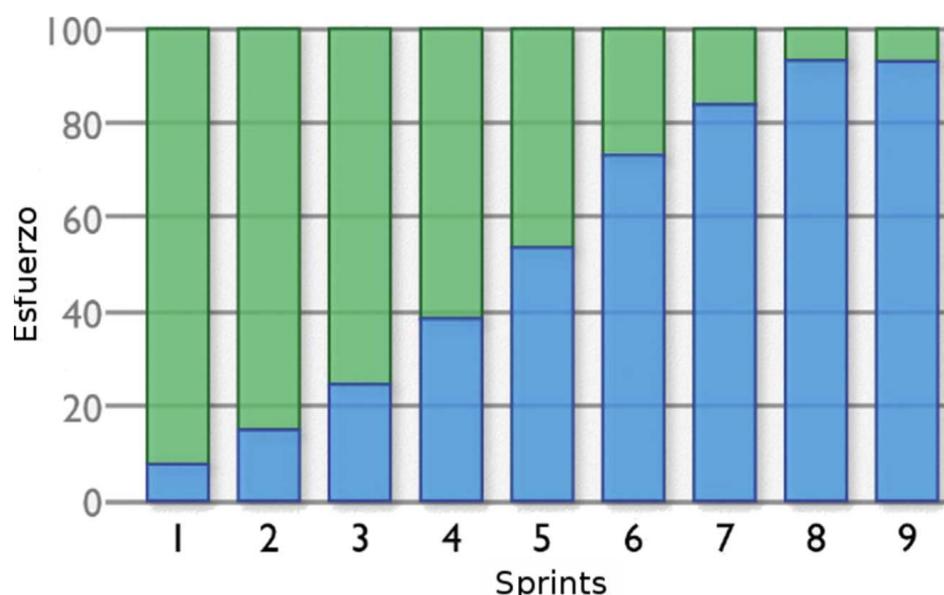
Sprint retrospective meeting

- 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 següent SPM
- Assisteixen el SM, el DT i (opcionalment) el PO
- 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 millors que es poden implantar per al següent Sprint

Sprint

- Hi ha un esprint d'anàlisi?
- Hi ha un esprint de proves?
- Hi ha un esprint de "estabilització"?

Sprint: Esforç en cada Sprint



- A la gràfica: verd representa l'arquitectura i blau funcionalitat visible per a l'usuari
- Al principi és més difícil però en cada Sprint sempre s'ha d'implementar funcionalitat visible per als usuaris

Release Sprint

- En acabar un Sprint, sempre s'ha d'ofrir un increment usable de la funcionalitat
- De totes maneres, pot haver Sprints especials anomenats Sprints de Release en els quals es fan tasques específiques com:
 - Proves MTBF (Mean time between failures)
 - Proves d'estrès, rendiment i usabilitat
 - Retocs en la documentació
 - etc
- En resum, tasques molt enfocades a fer una release final

Sprint: Terminació

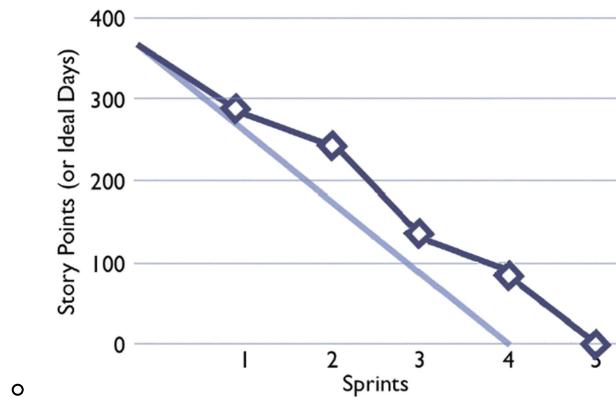
- Si un canvi no es pot mantenir fora de l'Sprint
 - Canvi tecnològic ineludible
 - Canvi de requisits dràstic
- L'entrenador ho pot sol·licitar, però ho decideix el PO
 - Circumstància extrema, infreqüent
 - Després
- El treball es rebutja
- El següent pas consisteix a iniciar un nou Sprint

> Artefactes

Product backlog

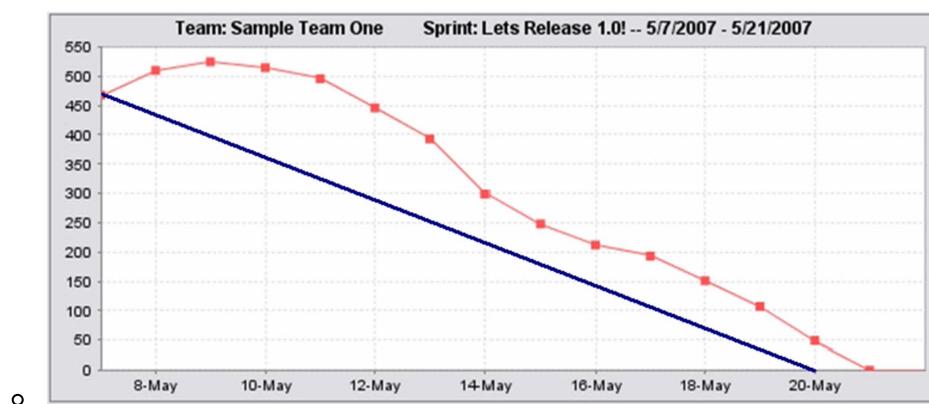
- Un projecte Scrum elcomençà el Product Owner generant el Product backlog. En el Product backlog:
 - Indica els requisits funcionals, no funcionals i casos d'ús
 - Prioritza perquè les funcionalitats que aporten més valor s'implementin primer
 - Divideix les diferents funcionalitats en versions
- Generar el Product backlog és el punt de partida encara que molt possiblement els requisits i prioritats poden canviar
- Si el product owner té problemes per generar el backlog, el Scrum Master pot ajudar
- Llista ordenada de tot el necessari i font única de requisits
 - El PO és la seva responsable del seu contingut, de fer-ho visible i d'ordenar
 - Mai està complet
- Mai està complet
 - Reflecteix el coneixement actual i en evolució dels requisits
 - Evoluciona i canvia constantment per ser apropiat, competitiu i útil
- Conté
 - Característiques, funcionalitats, requisits, millores, solucions a bugs ...
 - Que constitueixen el necessari per aconseguir un producte
- Ítems en el PB
 - Ordenats: Per valor, risc, prioritat i necessitat
 - Etiquetats: Descripció i estimació
 - Adequadament detallats: Els primers ítems estan més detallats
- Grooming
 - Afegeix detalls, estimacions i vista al PB
 - Procés continu (ongoing) entre PO i DT

- Realitzat en paral·lel a la resta d'activitats
- No consumeix més del 10%
- Monitorització del progrés



Sprint backlog

- Format per:
 - Objectiu
 - (+) Subconjunt del PB que es realitzarà en el Sprint
 - (+) Pla per fer-ho
- Monitorització



Increment

- Conjunt d'ítems del PB completats fins a la data
 - FETS: Definir FET
- Potencialment lliurable
 - Alta qualitat
 - Provat
 - Complet
 - El que fa, ho fa bé

Task boards

- Esquema:

Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Test the... 8 Code the... 2 Code the... 8 Test the... 8 Test the... 4	Code the... DC 4 Test the... SC 8	Test the... SC 6	Code the... D 4 Test the... SC 8 Test the... SC 8 Test the... SC 6
As a user, I... 5 points	Code the... 8 Test the... 8 Code the... 4 Code the... 6	Code the... DC 8		Test the... SC 8 Test the... SC 6

- Task board “manual”:

