

Models Àgils: SCRUM I AGILE



TABLE OF CONTENTS

01

Ford
(Sistema push)

02

Lean
manufacturing

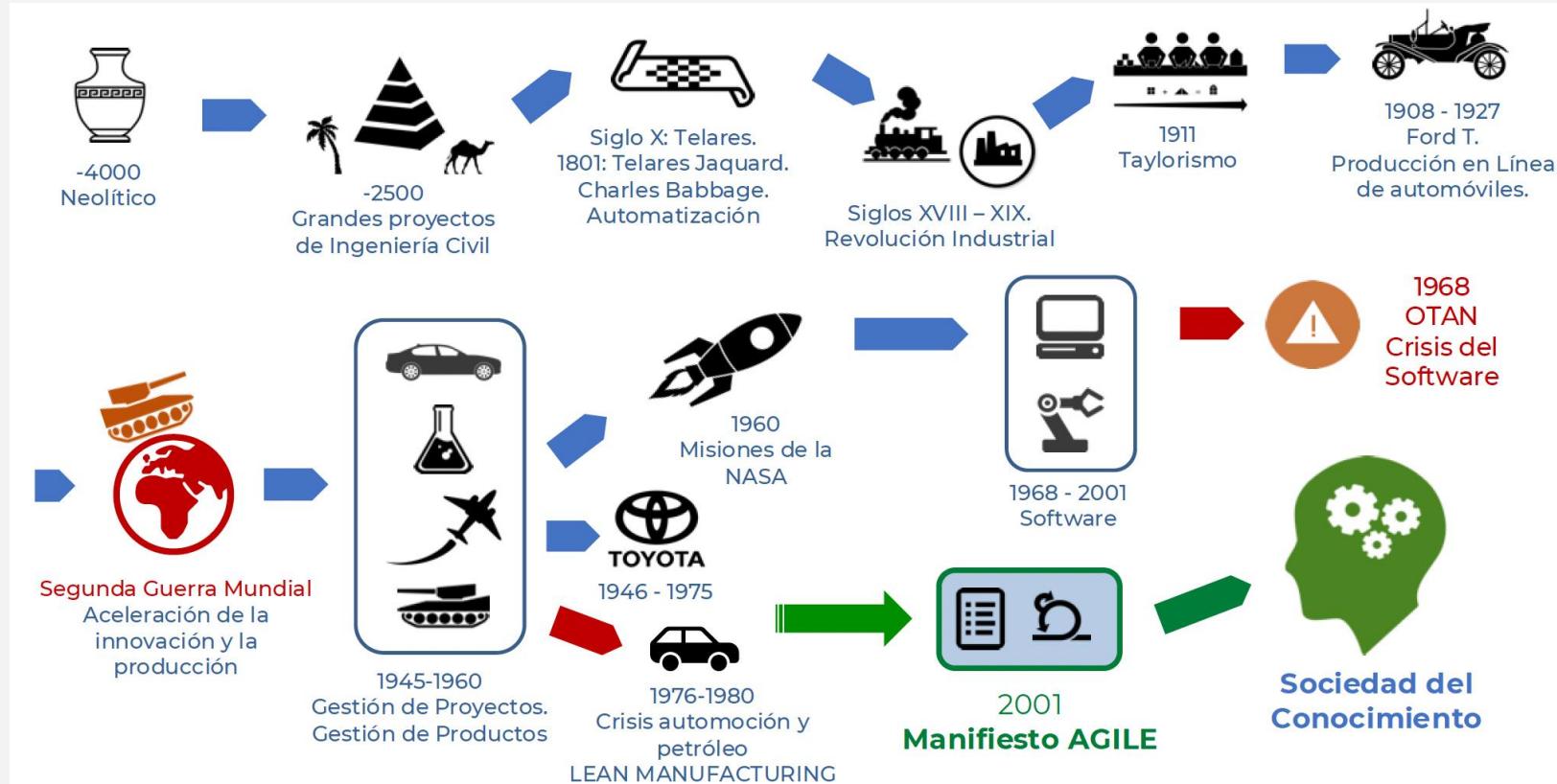
03

Manifest AGILE

04

Características de les
metodologies àgils







01

Ford (Sistema push)

Historia i context de les metodologies àgils

Què feia Ford?

Ford fabricava **grans volums d'un mateix producte estandarditzat**, especialment el **Model T**.

- **Poca varietat**
- **Configuració gairebé única**
- “Qualsevol client pot tenir el cotxe del color que vulgui, sempre que sigui negre”



Historia i context de les metodologies àgils

Què feia Ford?

Ford fabricava **grans volums d'un mateix producte estandarditzat**, especialment el **Model T**.

- **Poca varietat**
- **Configuració gairebé única**
- “Qualsevol client pot tenir el cotxe del color que vulgui, sempre que sigui negre”

L'objectiu era **reduir costos per economies d'escala**.

Ford va introduir la **línia de muntatge mòbil (1913)**:

- El producte es movia.
- Els treballadors feien tasques molt específiques i repetitives.
- Gran especialització.
- Temps de producció dràsticament reduït



Historia i context de les metodologies àgils

Què feia Ford?

Ford fabricava **grans volums d'un mateix producte estandarditzat**, especialment el **Model T**.

- Poca varietat
- Configuració ràpida
- “Qualsevol color, sempre negre”

L'objectiu era:

MODEL PUSH

Ford va introduir la **línia de muntatge mòbil (1913)**:

- El producte es movia.
- Els treballadors feien tasques molt específiques i repetitives.
- Gran especialització.
- Temps de producció dràsticament reduït



Historia i context de les metodologies àgils

Què feia Ford?

Ford va introduir la **línia de muntatge mòbil (1913)**:

- El producte es movia.
- Els treballadors feien tasques molt específiques i repetitives.
- Gran especialització.
- Temps de producció dràsticament reduït

Per què el model Fordista va començar a fallar als anys 70?.

- El mercat va començar a demanar més models, més opcions i més diferenciació.
- El sistema no podia adaptar-se sense disparar costos.
- Inventaris separats (costos augmenten)
- Errors de planificació fan canvis lents i costos elevats
- Producció en massa funciona molt bé amb **baixa entropia del mercat**.



Regles PUSH

Escenari

- Cada persona és una màquina en línia
- Cada post-it és un producte.

Regles

- Cada màquina ha de produir **exactament 5 unitats seguides**
- No pot passar a la següent operació fins acabar les 5.
- No pot esperar instruccions de la següent màquina



Regles PUSH

Resultats

- WIP elevat
- Colls d'ampolla si alguna màquina és més lenta
- Lead time llarg per a la primera unitat
- Inventari acumulat
- Problemes amagats





02

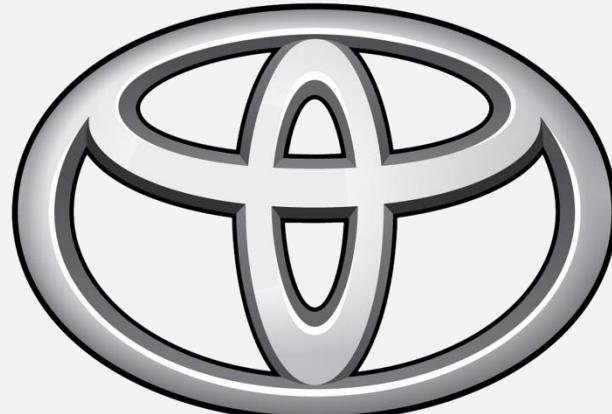
Lean Manufacturing (sistema pull)

Història i context de les metodologies àgils

"Lean Manufacturing uses the minimum necessary resources, removing all wastes and maximising the product value."



Historia i context de les metodologies àgils



TOYOTA

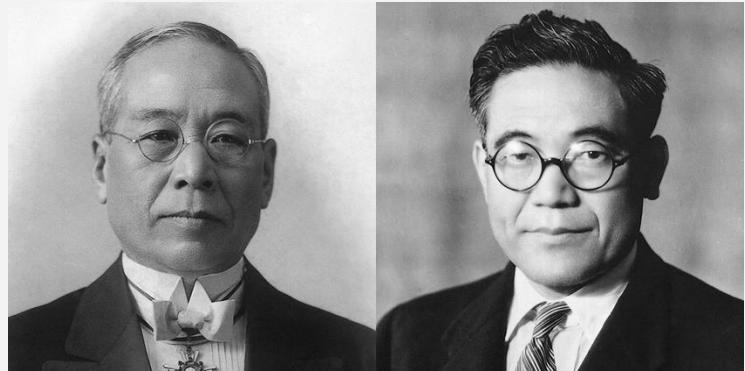


1. Utilitzar els recursos mínims necessaris per construir un producte
2. Evitar i/o eliminar temps morts
3. Reduir els inventaris
4. Millorar la eficiència
5. Millorar continua per incrementar la qualitat del producte

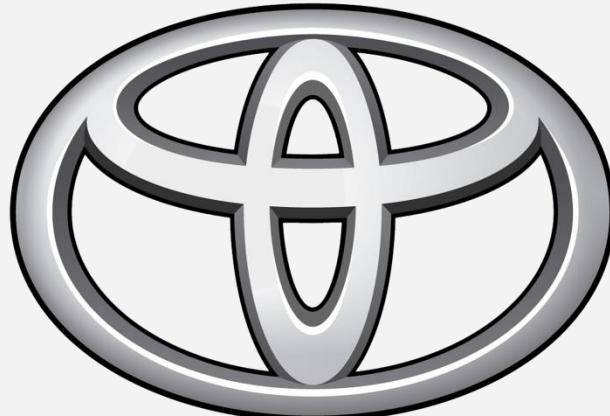
Historia i context de les metodologies àgils

Va començar amb Sakichi i Kiichiro Toyoda, i va ser perfeccionada per l'enginyer **Taiichi Ohno** a finals de la dècada de 1940

Va sorgir com una alternativa al model de producció en massa de **Ford**, buscant lliurar el que el client necessita exactament quan ho necessita, en lloc de produir grans estocks per vendre'ls després.



Historia i context de les metodologies àgils



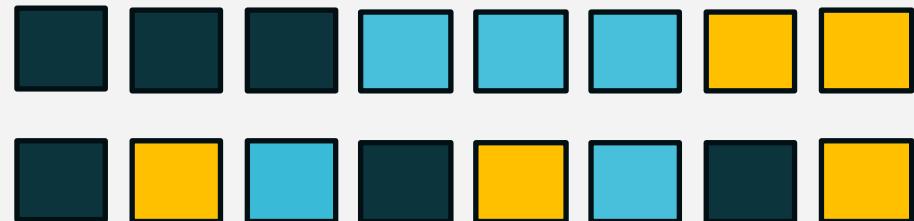
TOYOTA



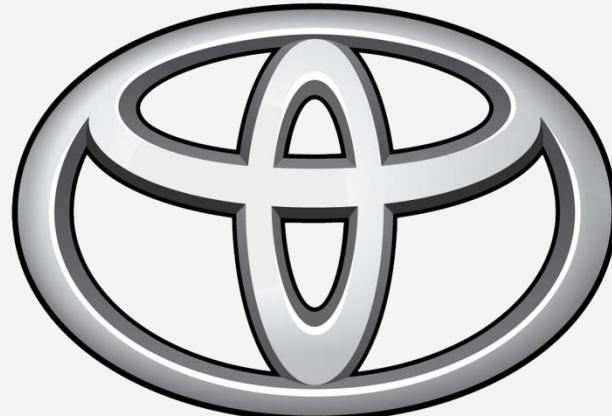
Lean no elimina la variabilitat sinó l'estructura.

Just-In-Time (JIT):

- Produir en lots petits
- Reposar sota demanda real
- Equilibrar càrregues



Historia i context de les metodologies àgils



TOYOTA



Com ho van fer?

- Reduir temps de canvi de configuració de 2 hores a 5 minuts
- Aplicar modularitat:
 - Plataformes comunes
 - Components compartits
 - Opcions finals diferenciades

Regles PULL

Escenari

- Cada persona és una màquina en línia
- Cada post-it és un producte.

Regles

- Un límit de WIP: màxim 1 o 2 post-its entre estacions.
- La última màquina només treballa si el client demana una unitat.
- Les altres màquines només treballen si la posterior li demana.
- No pot esperar instruccions de la següent màquina



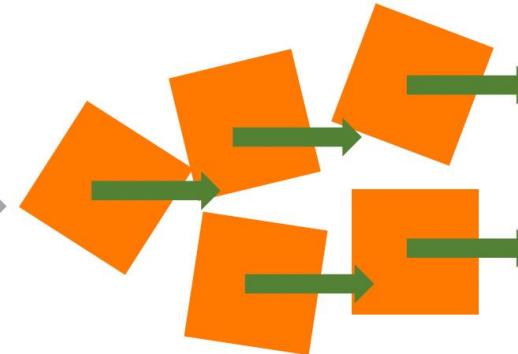
Regles PULL

Resultats

- WIP molt baix
- Problemes visibles immediatament
- Flux continu
- Lead time curt
- Sistema autoequilibrat



PUSH



PULL





I què passa amb
el software?



Historia i context de les metodologies àgils

La Crisi del Software

La **crisi del programari** és el conjunt de problemes detectats als anys 60 i 70 en el desenvolupament de programari, caracteritzats per retards, sobrecostos i baixa qualitat, que van motivar la creació de l'enginyeria del programari com a disciplina formal.

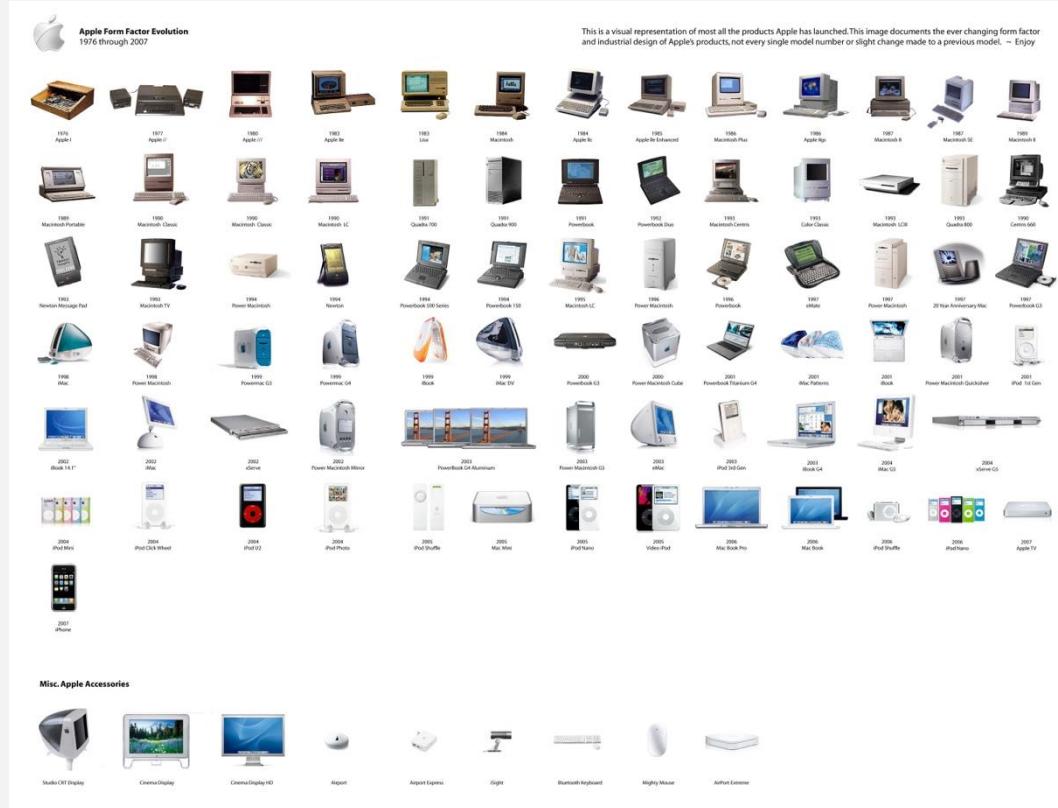
D'aquí van aparèixer els primers models iteratius i incrementals.

Historia i context de les metodologies àgils

1. Reducció dels defectes de codi
2. Evitar excessos de funcionalitats innecessàries
3. Reduir la falta de productivitat (excés de documentació)
4. Optimització del treball i el temps de treball
5. Feedback de millora continua per créixer com equip
6. **Evitar el risc del deute tècnic**

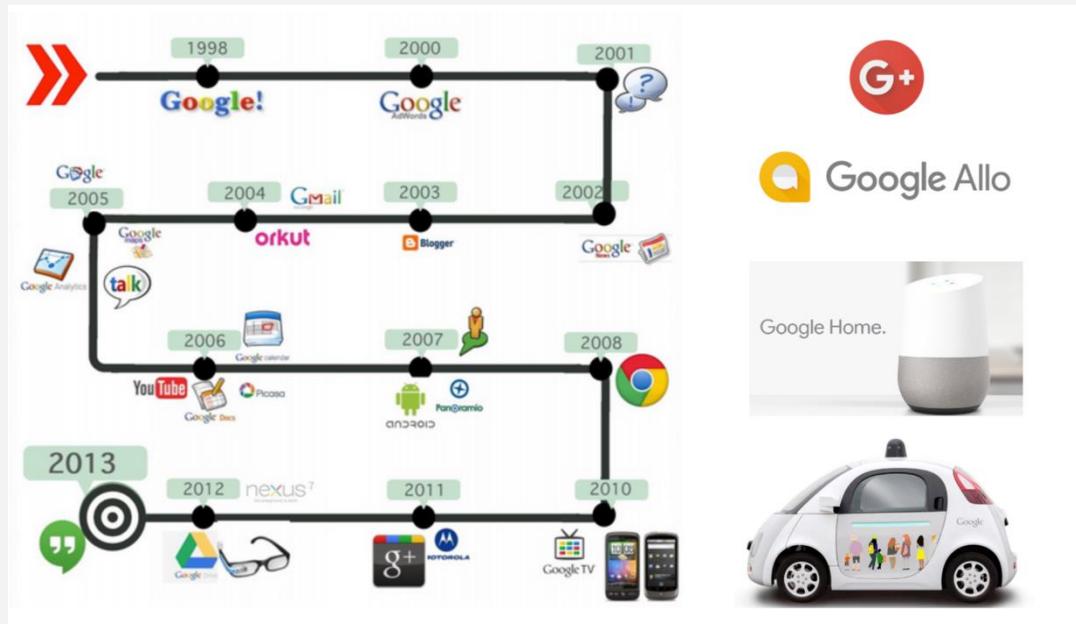


Historia i context de les metodologies àgils



*La velocitat dels canvis
del mercat obliguen a
les empreses tenir que
adaptar-se*

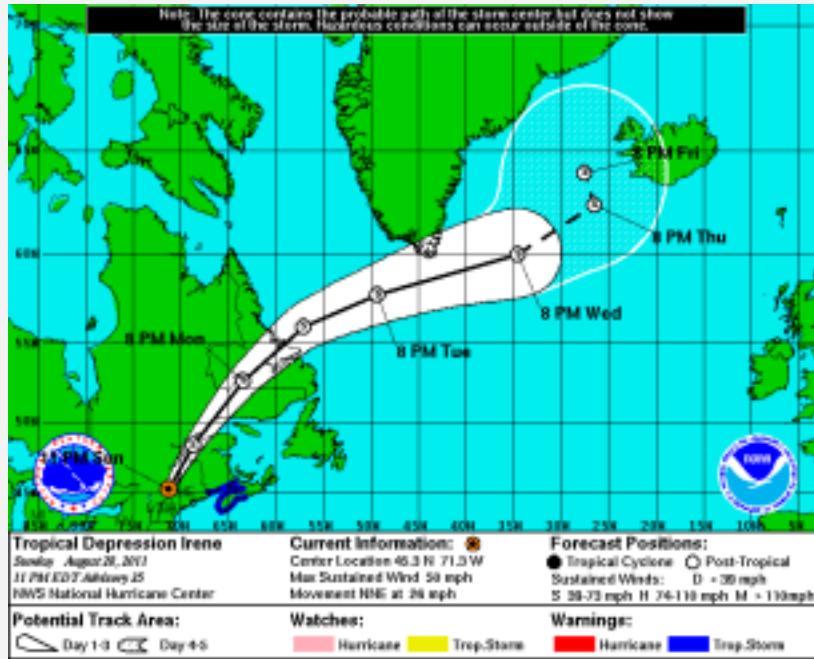
Historia i context de les metodologies àgils



*La desconeixença
implica cobrir el
màxim mercat
possible*



Historia i context de les metodologies àgils



Patró de previsió amb incertesa.

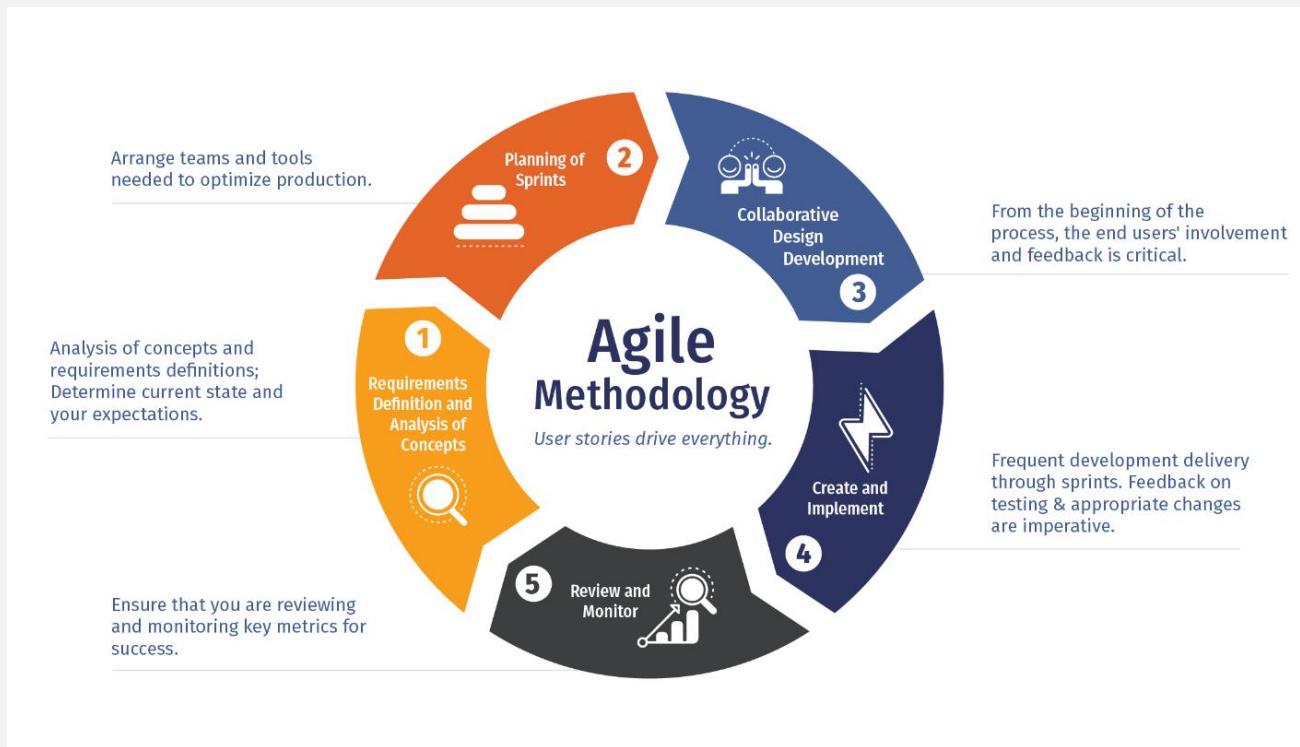
. En la gestió de projectes, el Con de la Incertesa descriu l'evolució de la quantitat d'incertesa durant un projecte



Manifest AGILE

El **Manifest Agile** és una declaració de principis publicada el 2001 per un grup de 17 professionals del desenvolupament de programari reunits a Snowbird (Utah).

Marc de gestió de projectes que divideix els projectes en varies fases dinàmiques, conegeudes com **SPRINTS**.



Manifest AGILE

Les metodologies àgils van néixer com a reacció als models tradicionals de desenvolupament (especialment el model en cascada).

Molt rígids

Fortament
documentals

Lents en
entorns
canviants

Les metodologies àgils van néixer com a reacció als models tradicionals de desenvolupament (especialment el model en cascada).



MANIFESTO AGILE

El manifest àgil estableix les principals regles de la metodologia àgil.

Principis darrere el manifest àgil

Nosaltres seguim els següents principis:

La nostra principal prioritat és satisfer al client mitjançant el lliurament primerenc i continu de programari que aporti valor.

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.

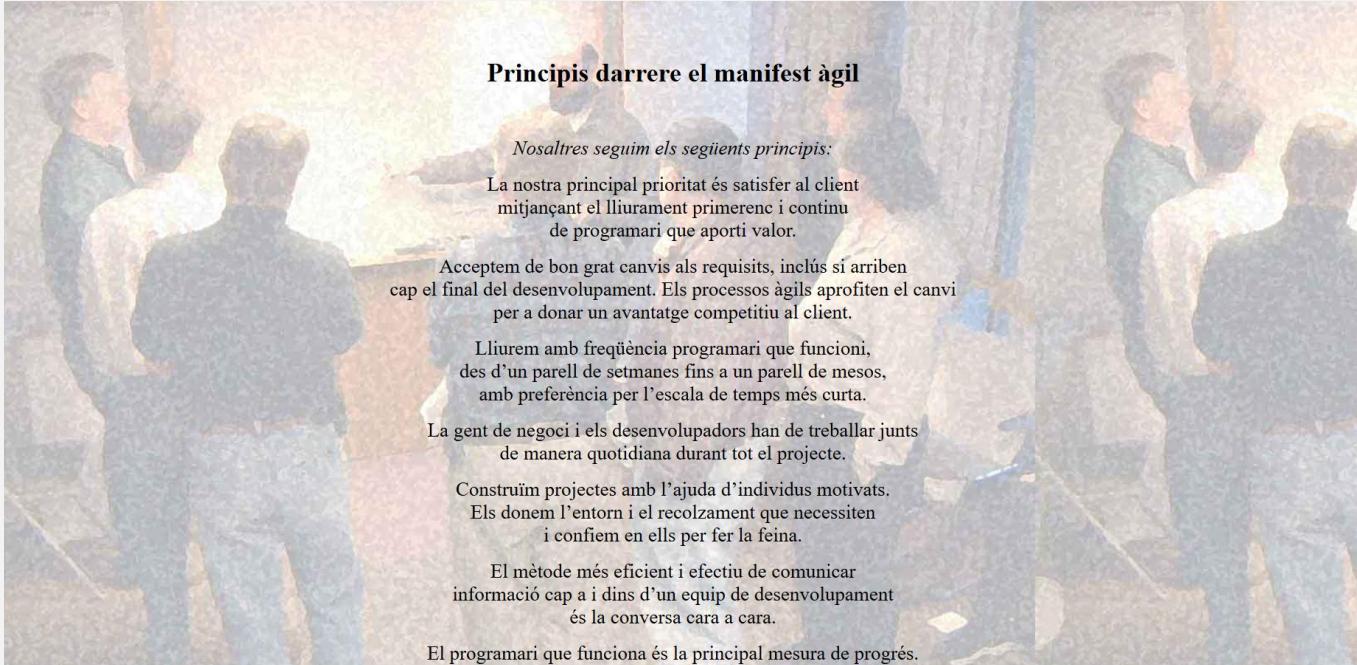
Lliurem amb freqüència programari que funcioni, des d'un parell de setmanes fins a un parell de mesos, amb preferència per l'escala de temps més curta.

La gent de negoci i els desenvolupadors han de treballar junts de manera quotidiana durant tot el projecte.

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.

El mètode més eficient i efectiu de comunicar informació cap a i dins d'un equip de desenvolupament és la conversa cara a cara.

El programari que funciona és la principal mesura de progrés.





Manifest AGILE

AGILE

TRADICIONAL

Interacció i Persones	Processos i Eines
Prototips funcionals	Documentació exhaustiva
Involucrar al client	Negociació contractual
Resposta al canvi	El Pla



Manifest AGILE

Els valors fonamentals

<u>Interacció i Persones</u>	Interacció del equip i client per construir el producte
<u>Prototips funcionals</u>	Software funcional per mostrar al client
<u>Involucrar al client</u>	El desenvolupament es centra en l'usuari (client)
<u>Resposta al canvi</u>	Capacitat d'afegir o eliminar requisits i funcionalitats sense impactar als costos del producte.

MANIFEST AGILE

12 PRINCIPIIS

Customer Satisfaction

Adaptation

Frequent Deliveries

Team Work

Motivation

Face to Face
Communication



MANIFEST AGILE

12 PRINCIPIS

Functional Prototypes

Constant Work

Technical Excellence

Simplicity

Self-Management

Quality
&
Improvement



Característiques de les metodologies àgils

PERSONES COM FOCUS PRINCIPAL

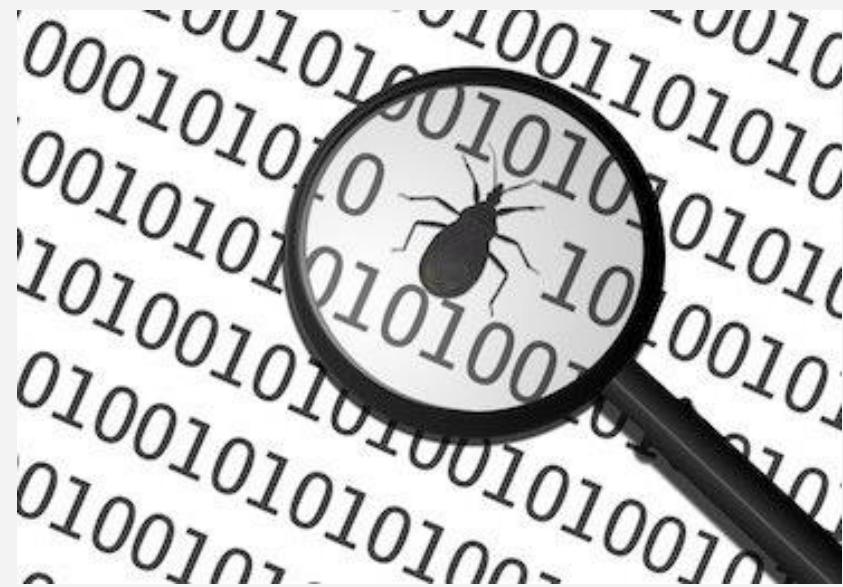
- 1. Equip.** Importància de tenir motivat i involucrat a un equip de treball.
- 2. Client.** Importància de tenir motivat al client per tenir naturalitat de comunicar les necessitats.



Característiques de les metodologies àgils

CALITAT DEL SOFTWARE

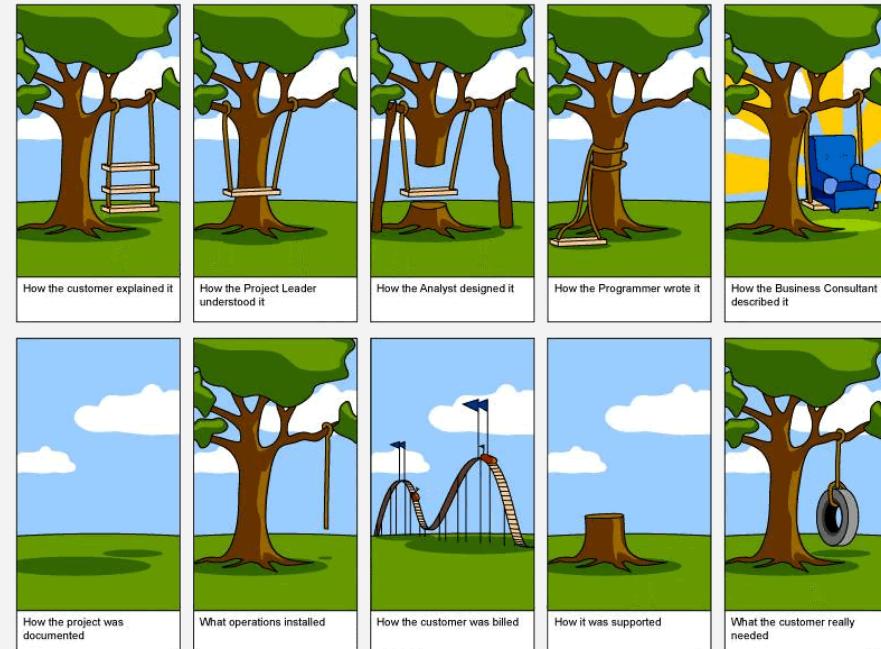
- 1. Minimitzar els errors.** Reduir i controlar els "bugs" dins el codi.
- 2. Qualitat de la documentació.** Qualitat de la documentació per a que els equips dins el client puguin utilitzar-la.



Característiques de les metodologies àgils

Desenvolupament centrat en el client

- 1. Entendre les necessitats del client.**
Importància de entendre les necessitats del client. Què es el que vol conseguir?
- 2. Desenvolupament basat en les “features” més rellevants per el client.** De més importància a menys.
- 3. Guia i Consell al client** per ajudar-lo a prendre una millor decisió.

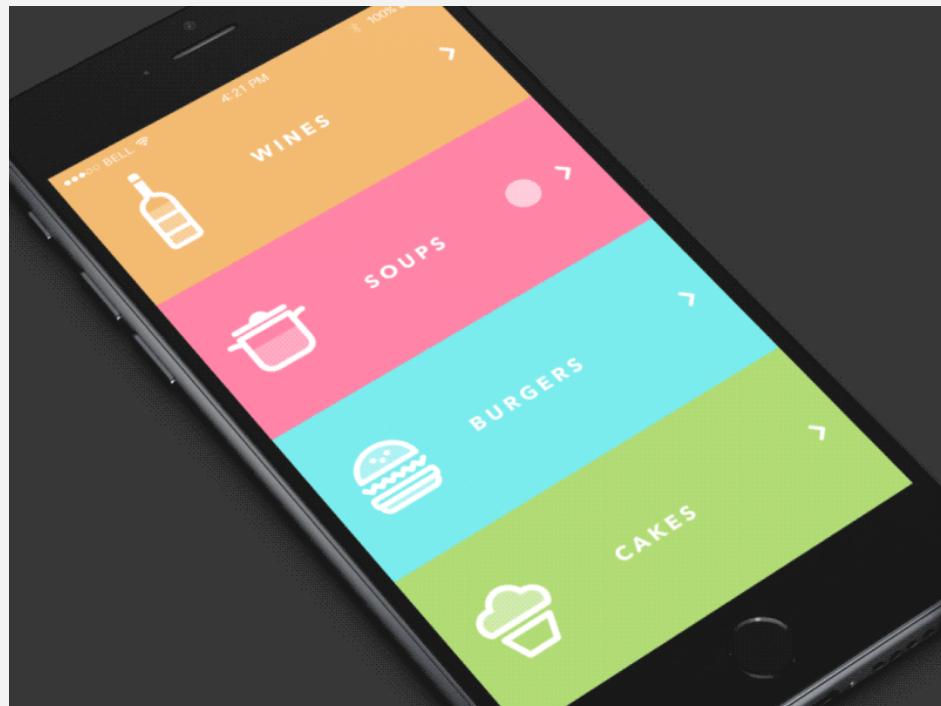


Característiques de les metodologies àgils

Simplicitat

- 1. Desenvolupar les intereficies.** Simplicitat en el desenvolupament de les parts visuals.
- 2. Desenvolupament del backend.** Simplicitat i modularitat en el codi.

Un codi més senzill es més entenedor



Característiques de les metodologies àgils

Anàlisis i Monitorització

- 1. No confondre monitorització amb control.**
Monitorització per aprendre i millorar, no per controlar al usuari.
- 2. Conèixer és saber les nostres debilitats** y això, significa, fer de les **nostres debilitats fortaleces**.

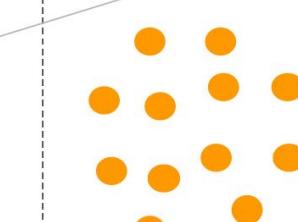


Què és Agile?

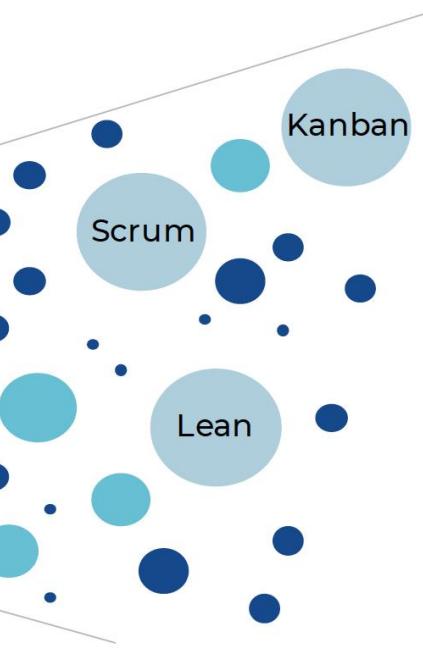


Agile és una
mentalitat

Descrita en
4 valors



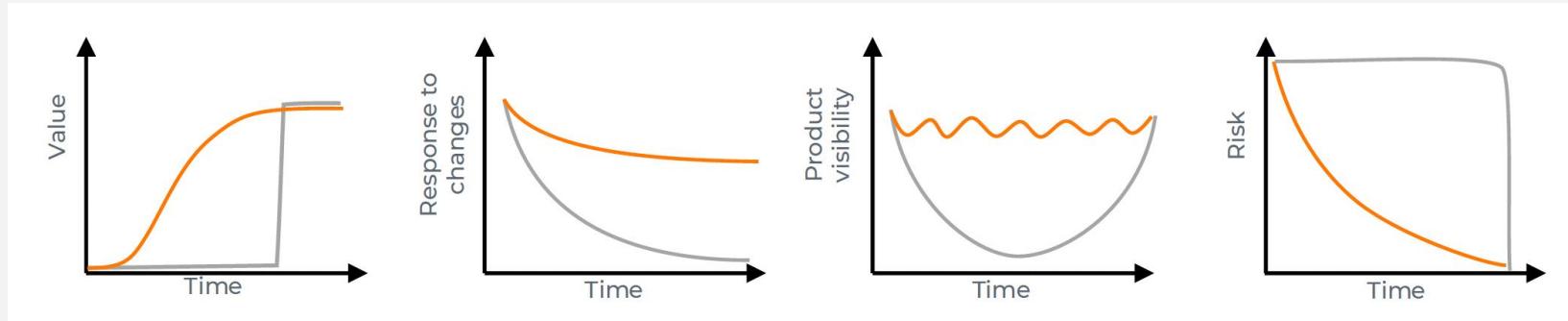
Definida per
12 principis



Manifestat a través de
diversos marcs de treball



Per què Agile?



Genera
valor més
aviat

Millora la
gestió de
canvis

Major
visibilitat per
l'usuari

Millor gestió
de riscos



Tipologies de metodologies àgils

- 1. SCRUM.** Consisteix en treballar per iteracions (**SPRINTS**).
- 2. Extreme Programming (XP).** Entregar al client el producte exactament com l'ha demanat basat en costos, calitat, temps i marge (Combinat amb SCRUM).
- 3. SAFE.** Adoptar metodologies àgils i Lean Project Management per grans projectes.
- 4. AGILE KANBAN.** Utilitzar model Kanban per saber en tot momento el estat de les activitats.
- 5. SCRUMBAN.** Combinació de lo millor de SCRUM i Kanban.





SCRUM i Metodologies àgils

INDEX

01

Técnicas adaptativas
vs predictivas

02

Models de
software
predictius

03

Models de
software
adaptatius

04

Extreme
Programming

05

SCRUM



Models de software adaptatius vs predictius

MODELS DE SOFTWARE PREDICTIUS

- Models lineals (després d'una etapa en realitzem la següent)
- Models dividits en fases o etapes, i el projecte es dona per finalitzat quan totes les etapes estan completades.
- Aquests models comencen amb una etapa de formalització de requisits per a tot el projecte.



Models de software adaptatius vs predictius



MODELS DE SOFTWARE ADAPTATIUS

- Models que no són totalment lineals, ja que s'adapten a les necessitats de l'entorn o a les necessitats del client.
- Models dividits en sprints i, en cada sprint, realitzem una versió o extensió del producte final (prototips funcionals).
- Cada sprint comença amb la selecció de funcionalitats clau per al client.

Models de software adaptatius vs predictius

1. Amb els models predictius, el client no veu la funcionalitat del producte fins que no finalitzen totes les fases.
2. En els models adaptatius, cada prototip funcional es va adaptant o incrementant fins a cobrir totes les funcionalitats desitjades.

Metodologías predictivas



Metodologías ágiles



Models de software adaptatius vs predictius

Models Predictius	Models Adaptatius
Lineal	Incremental / Iteratiu
Fases o Etapes	Sprints o Iteracions
Requeriments formals al principi	Requeriments en tots els sprints



Tipus de projecte i model de procés

Enfocament experimental o àgil

Context: Innovació alta, risc alt

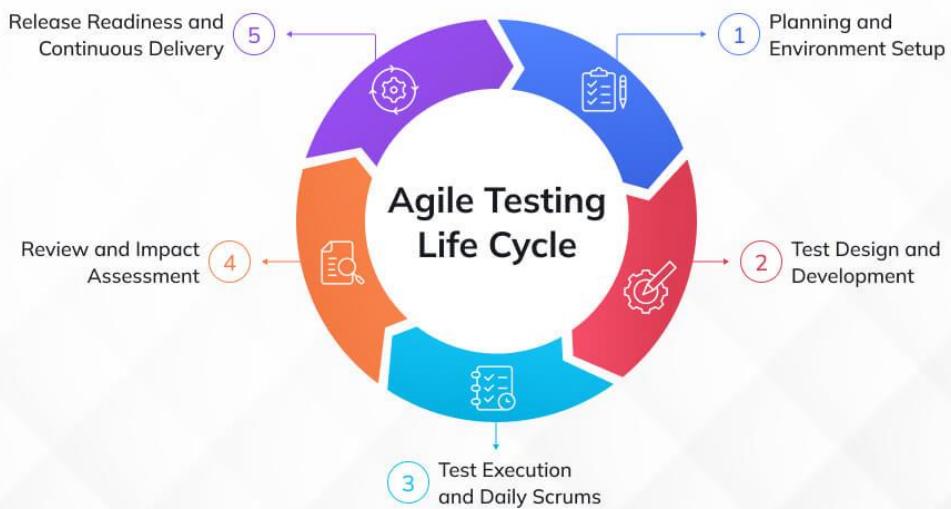
Quan la incertesa és alta, **planificar en detall deixa de ser efectiu.**

L'enfocament experimental assumeix que:

- No sabem què funcionarà
- Cal provar hipòtesis
- L'error és part del procés

La planificació és:

- Curta
- Flexible
- Orientada a l'aprenentatge



Discussió de models adaptatius i predictius

Volem planificar un viatge des de Barcelona a Londres i, amb una primera exploració, tenim els següents preus per activitat:

Item	Preu
Vols Barcelona – Londres (i viceversa). Dia 1 i Dia 4	200€/viatge
Hotel (Tots els dies)	100€ la nit
Visita London Bridge (Dia 1)	20€
Visita London Tower (Dia 2)	10€
Visita Backinham Palace (Dia 3)	20€

(L'estada serà de 4 dies. Com ho organitzaríeu de manera predictiva i com, de manera adaptativa?)



Discussió de models adaptatius i predictius

VIATJER PREDICTIU

Item	Preu
Dia 1: • Viatge Barcelona a Londres • Nits de Hotel • Visita London Bridge	200€ 400€ 20€
Dia 2: Visita London Tower	10€
Dia 3: Visita Backinham Palace	20€
Dia 4: Viatje Londres a Barcelona	200€
TOTAL	850€



Discussió de models adaptatius i predictius

VIATJER ADAPTATIU

Item	Preu
Dia 1: • Viatje Barcelona a Londres • Nit de hotel (Dia concret) • Visita London Bridge	200€ 100€ 20€
Dia 2: • Visita London Tower • Nit de hotel (Dia concret)	10€ 100€
Dia 3: • Visita Backinham Palace • Nit de hotel (Dia concret)	20€ 100€
Dia 4: • Viatje Londres a Barcelona • Nit de hotel (Dia concret)	200€ 100€
TOTAL	850€

Discussió de models adaptatius i predictius

QUE PASSARÀ SI S'ENDARREREIX EL VOL 1 DIA?

VIATGER ADAPTATIU

Item	Preu
Dia 1: • Viatge Barcelona a Londres • Nit de hotel (Dia concret) • Visita London Bridge	200€ 0€ (abans 100€) 0€ (abans 20€)
Dia 2: • Visita London Tower • Nit de hotel (Dia concret)	10€ 100€
Dia 3: • Visita Backinham Palace • Nit de hotel (Dia concret)	20€ 100€
Dia 4: • Viatje Londres a Barcelona • Nit de hotel (Dia concret)	200€ 100€

VIATGER PREDICTIU

Item	Preu
Dia 1: • Viatje Barcelona a Londres • Nit de hotel (Dia concret) • Visita London Bridge	200€ 400€ 20€
Dia 2: • Visita London Tower • Nit de hotel (Dia concret)	10€ 100€
Dia 3: • Visita Backinham Palace • Nit de hotel (Dia concret)	20€ 100€
Dia 4: • Viatje Londres a Barcelona • Nit de hotel (Dia concret)	200€ 100€



Discussió de models adaptatius i predictius

QUE PASSARÀ SI S'ENDARREREIX EL VOL 1 DIA?

Itinerari	Predictiu	Adaptatiu
Dia 1	620€	200€
Dia 2	10€	110€*
Dia 3	20€	120€*
Dia 4	200€	300€*
TOTAL	850€	730€



(*) Tenint en compte que no hi hagui nous imprevistos

Discussió models adaptatius vs predictius

Que es millor tenir un **model predictiu** o **adaptatiu**?

1. Si el projecte és tancat (requisits molt clars des del principi), és millor ser predictiu.
2. Si el projecte és obert (requisits canviant), és millor ser adaptatiu.



Esta foto de Autor desconocido se concede bajo licencia de CC

Discussió de models adaptatius i predictius

Exemples de projectes Adaptatius i Predictius

Models Predictius

Projectes per estaments públics

Projectes tancats de investigació

Projectes on s'ha fet un gran estudi de requeriments

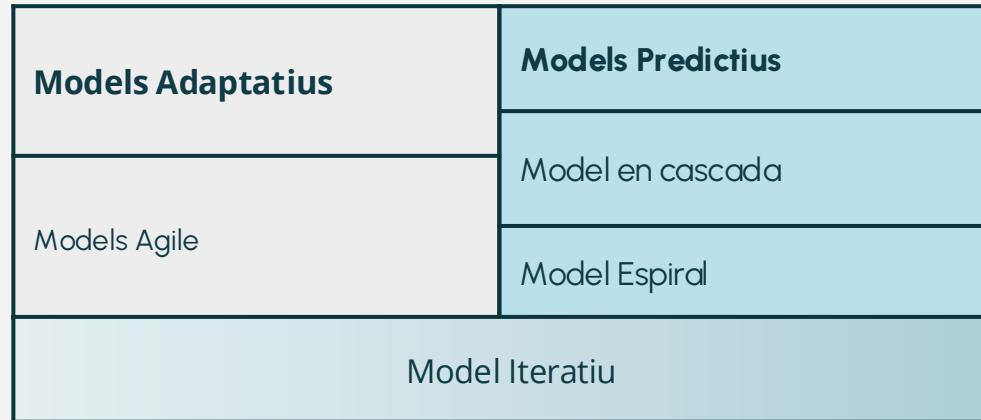
Models Adaptatius

Projectes a petita escala

Projectes on els requeriments no estan clars



Models de software adaptatius i predictius





KANBAN

Kanban és una paraula japonesa que significa **senyal visual** o **tauler de senyals**.

El sistema de targetes Kanban va ser inventat a finals de la dècada de 1940 per **Taiichi Ohno**, un enginyer industrial de Toyota que estava investigant mètodes per millorar el procés de fabricació de Toyota.

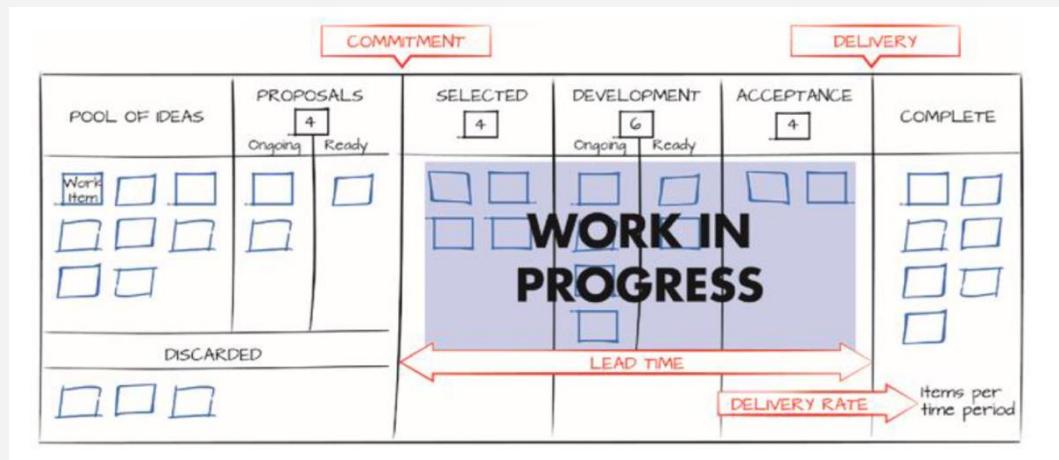




KANBAN

Proporciona un sistema de gestió de processos visuals que ajuda a la presa de decisions sobre què, quan i quant produir en projectes de coneixement.

- Promouen la transparencia.
- Optimitzen els fluxos de treball.
- Ajuden a detectar colls d'ampolla.
- Formen part de la millora contínua.





KANBAN

Proporciona un sistema de gestió de processos visuals que ajuda a la presa de decisions sobre què, quan i quant produir en projectes de coneixement.

- Promouen la transparència.
- Optimitzen els fluxos de treball.
- Ajuden a detectar colls d'ampolla.
- Formen part de la millora contínua.

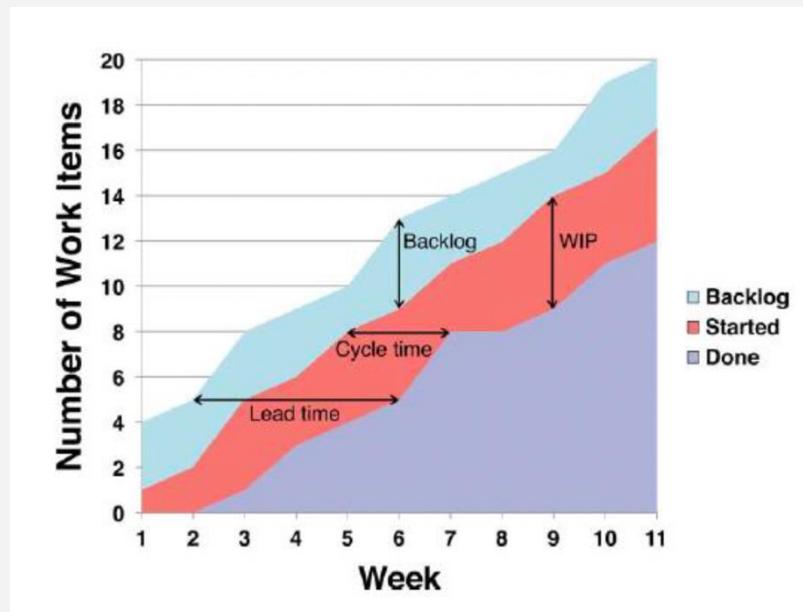
TO DO	IN PROGRESS	IN QA	DONE
Implement feedback collector NUC-205	Update T&C copy with v1.9 from the writers guild in all products that have cross country compliance NUC-213	Adapt web app no new payments provider NUC-346	Quick booking for accomodations - web NUC-336
Bump version for new API for billing NUC-206	Bump feedback icon version NUC-208	Purchasing error - edit fields NUC-338	Fluid booking on tablets NUC-343
Add NPS feedback to wallboard NUC-208	Tech spike on new stripe integration with paypal NUC-215	Change phone number field type to 'phone' NUC-217	Shoping cart purchasing error - quick fix required. NUC-354
Add analytics events to pricing page NUC-209			
Resize the images for the upcoming campaign NUC-210			



KANBAN

Diagrama de Flux Acumulatiu (Cumulative Flow Diagram)

- Es dibuixa comptant el nombre d'ítems per estat.
- Mostra el nombre acumulat d'ítems completats en un dia determinat.
- Identifica els colls d'ampolla i els problemes en el lliurament.
- Ajuda a comprendre com es comporta el flux i el *lead time* dins del sistema.

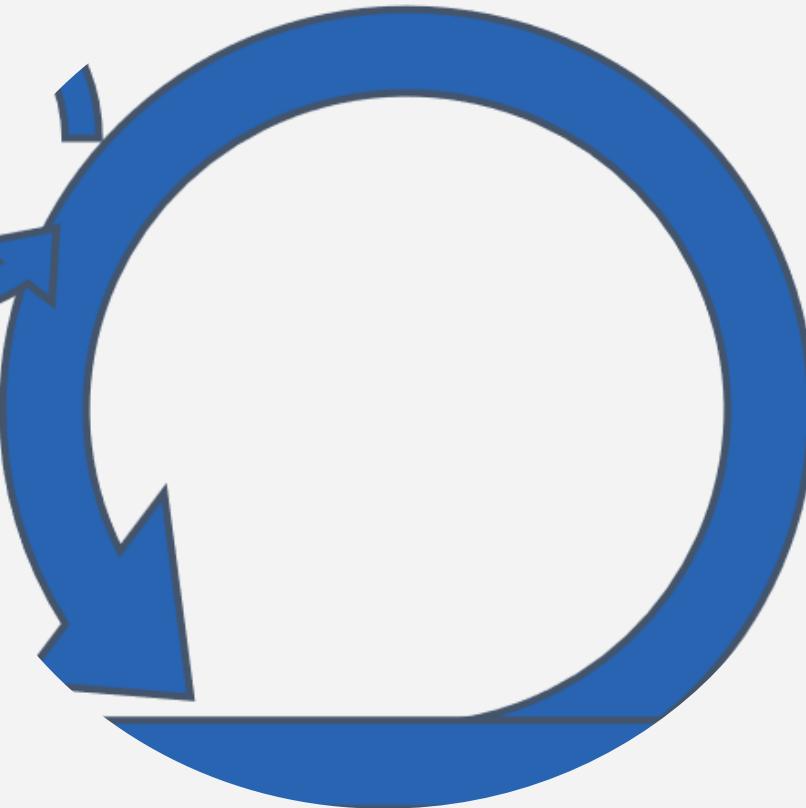




AGILE METODOLOGY



METODOLOGIES ÀGILS



SCRUM

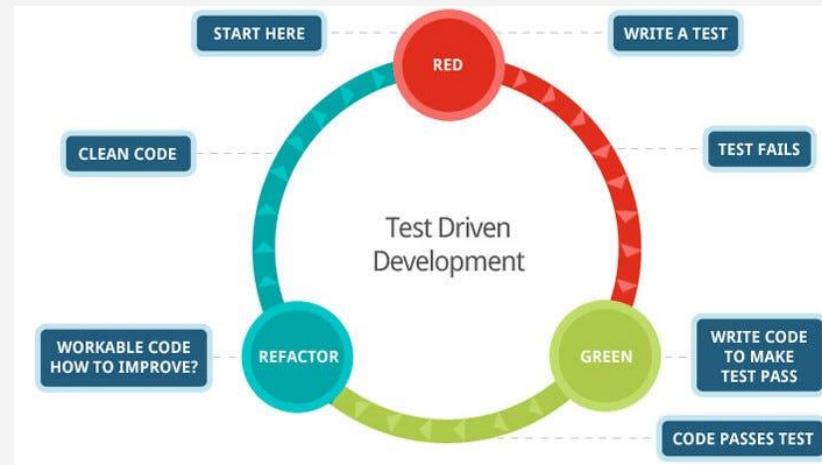
Tècnica que ajuda a maximitzar el valor del producte.

Consisteix a treballar mitjançant iteracions i, en cadascuna de les iteracions, es realitzen prototips funcionals.

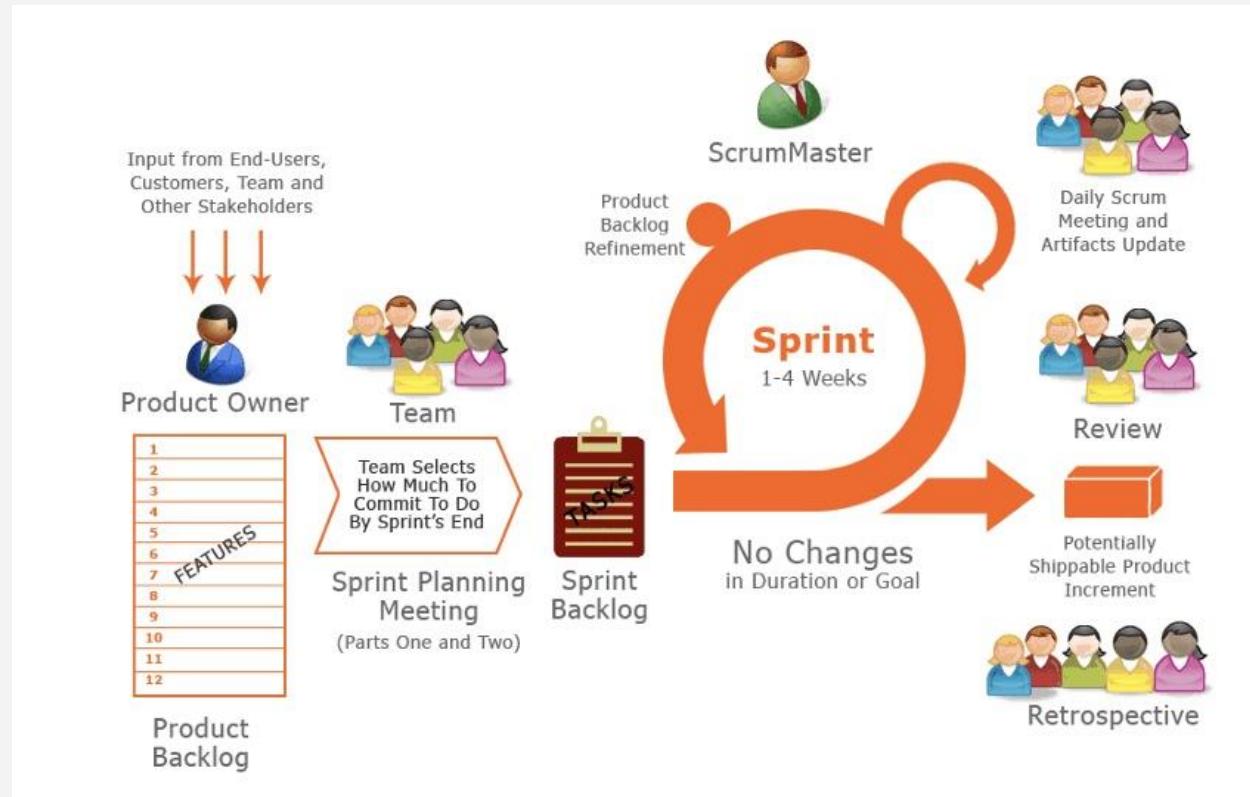
SCRUM: Test Driven Development

CARACTERISTIQUES PRINCIPALS

- **Reducció del temps de desenvolupament:** Productes que satisfan la majoria de les funcionalitats i els costos establerts.
- **Es pot llançar un producte en qualsevol moment:** Els prototips funcionals permetran llançar un producte al final de cada sprint (etapa del projecte).
- **Agilitat i flexibilitat:** Adaptabilitat als nous requisits i a les modificacions demandades.

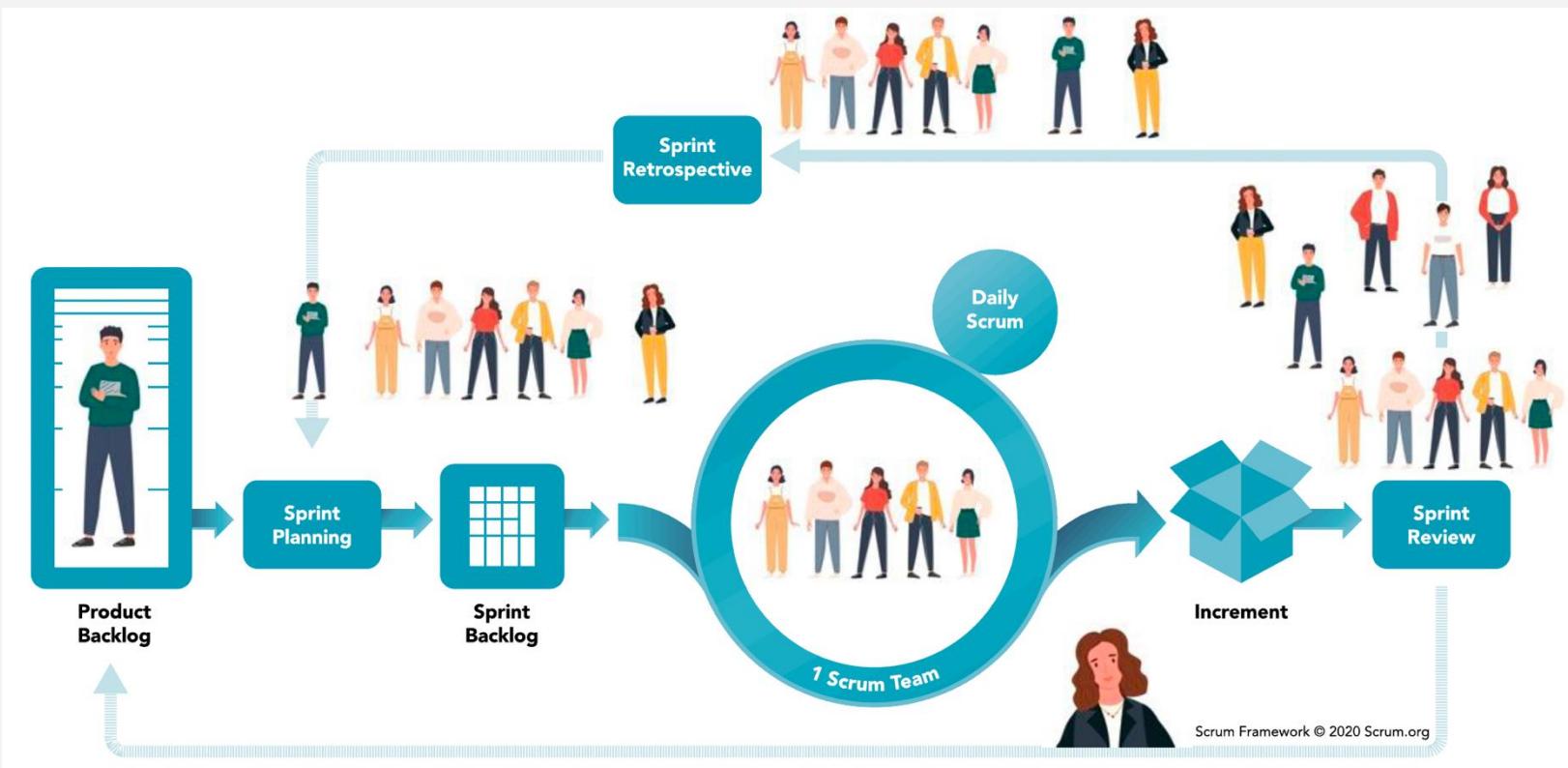


SCRUM: Distribució





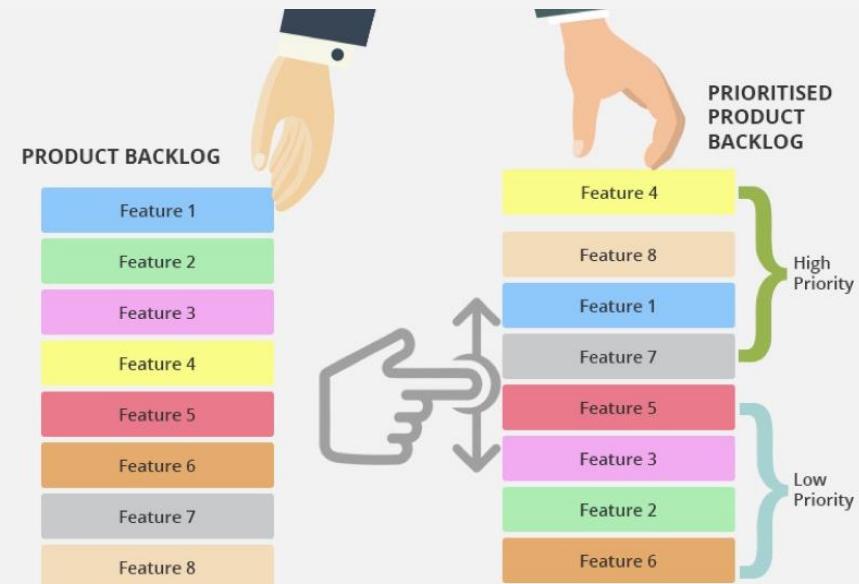
SCRUM: Distribució



SCRUM: Elements principals

Product Backlog

- **Llistat de característiques del producte.** Aquesta llista pot ser prioritizada o no prioritizada conjuntament amb el client.
- Aquesta llista **NO** està assignada ni **TEMPORIZADA**.



**As a <Role>
I Want to <function-description>
So I can: <value-statement>**

As a registered user,
I want to change my password,
so I can personalise it and better remember it.

SCRUM: Elements principals

User Stories (Histories de Usuari)

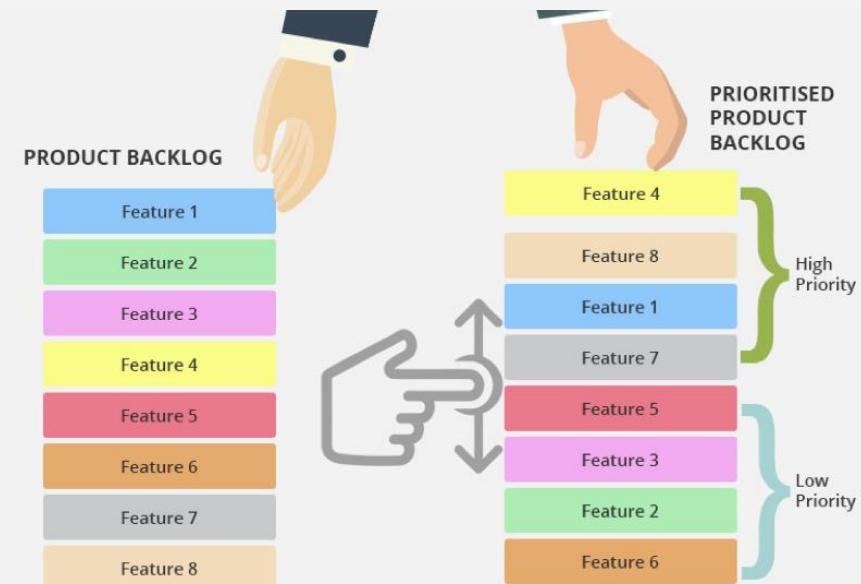
- Representació de les *features* o característiques del client.
- Aquests components han de ser clars i concisos, sempre representant una funcionalitat concreta.



SCRUM: Elements principals

Priorització del product backlog

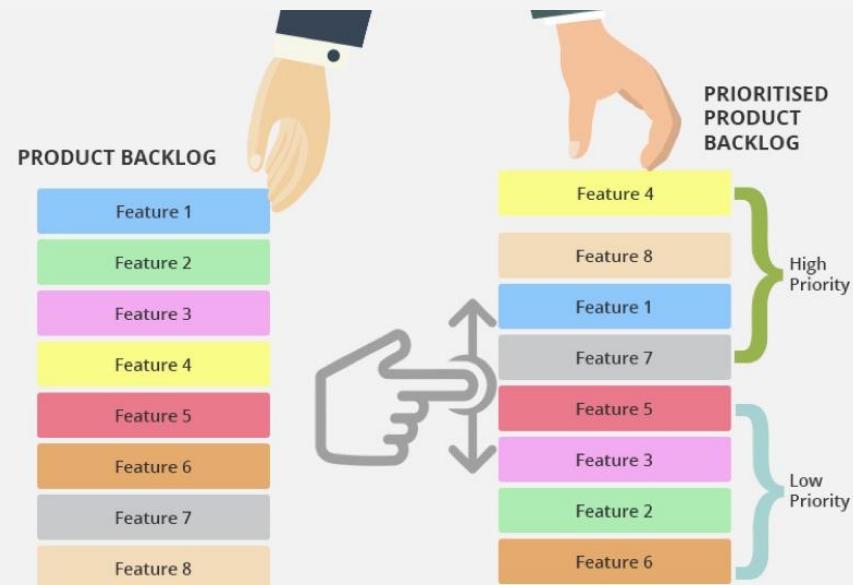
Balancejar entre la prioritat, el cost i el temps necessaris per realitzarlo.



SCRUM: Elements principals

Priorització del product backlog

1. Agrupar les tasques per prioritat.
2. Calcular la ràtio cost/hora.
 - **Cost:** valor en base a l'impacte en el projecte. Valor predeterminat.
 - **Hores:** estimació de treball segons la nostra experiència.
3. Ordenar de major a menor.





Scrum: Elements principals

Establir una priorització del Product backlog

User Story	Level	Hours	Value	Priority
Story A	MEDIUM	100	300	2
Story B	HIGH	30	200	1
Story C	MEDIUM	50	100	3
Story D	LOW	60	50	4

Scrum: Elements principals

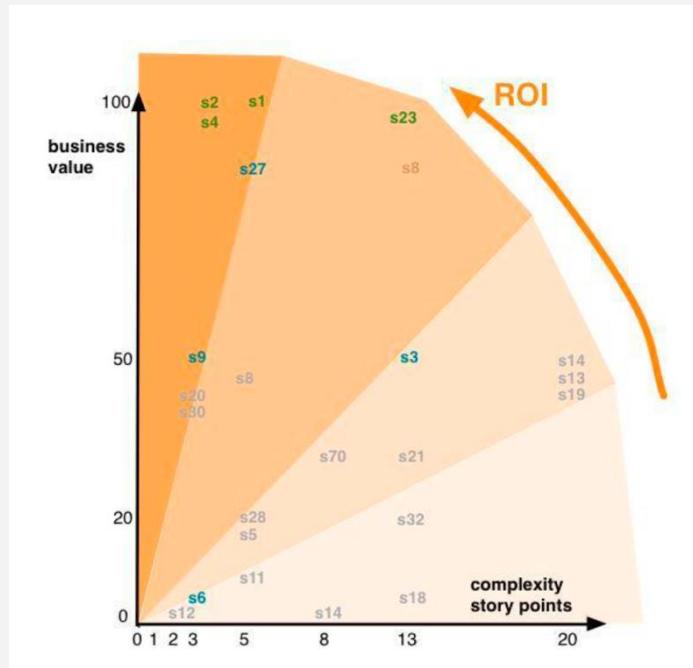
Establir una priorització del Product backlog



1. **Es realitza en cada *Sprint*.** És el que farem en aquella iteració.
2. **Divisió de les user stories en tasques més petites (*issues*).**
3. **Llistat priorititzat, assignat i temporalitzat.**

Scrum: Elements principals

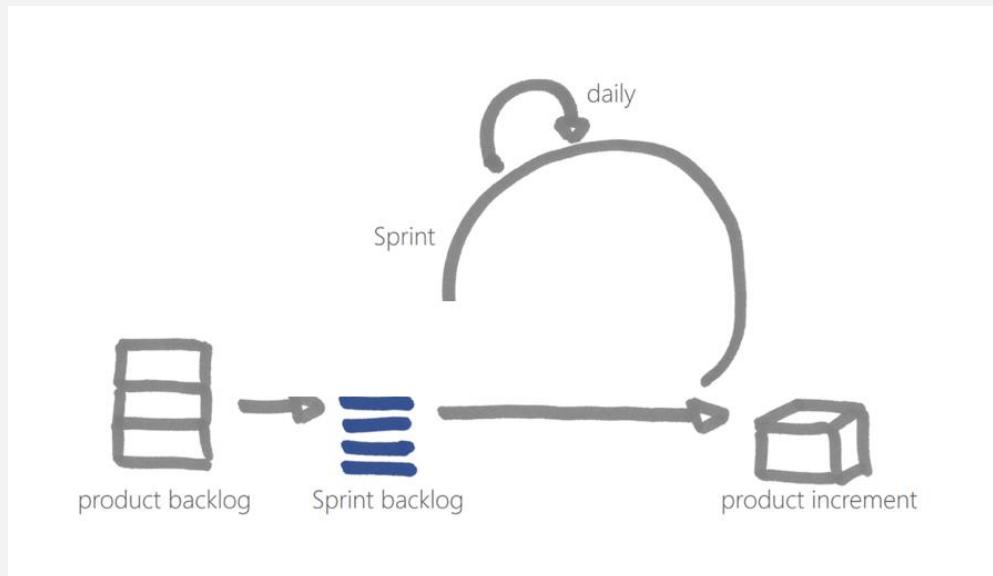
Establir una priorització del Product backlog



1. **Es realitza en cada Sprint.** És el que farem en aquella iteració.
2. **Divisió de les user stories en tasques més petites (*issues*).**
3. **Llistat prioritat, assignat i temporalitzat.**

Scrum: Elements principals

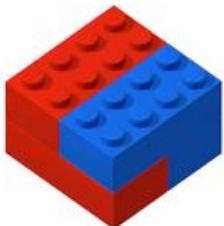
Característiques del sprint



- 1. Iteració de cada sprint.**
- 2. Definició del sprint backlog.**
- 3. Prototips funcionals al final de cada sprint.**

Scrum: Elements principals

Característiques dels increments



1. **Prototips funcionals incrementals**
2. **En cada Sprint, cal anar maximitzant les característiques del producte.**
3. **Al final, fent coses simples, s'acaba obtenint un programari complex**



Scrum: Elements principals

Característiques dels increments



1. **Prototips funcionals incrementals**
2. **En cada Sprint, cal anar maximitzant les característiques del producte.**
3. **Al final, fent coses simples, s'acaba obtenint un programari complex**

Scrum: Rols

Product Owner



Product Owner

1. És la veu del client dins de l'empresa.
2. Interacciona i gestiona el client.
3. Gestiona el product backlog.
4. Juntament amb el client, descobreix i identifica cadascuna de les necessitats i ajuda a transformar-les en requeriments.

Scrum: Rols

Development Team



1. Grup de treball encarregat de desenvolupar el producte.
2. Equip multidisciplinari.
3. Equip autogestionat.
4. Alta comunicació i transparència.

Scrum: Rols

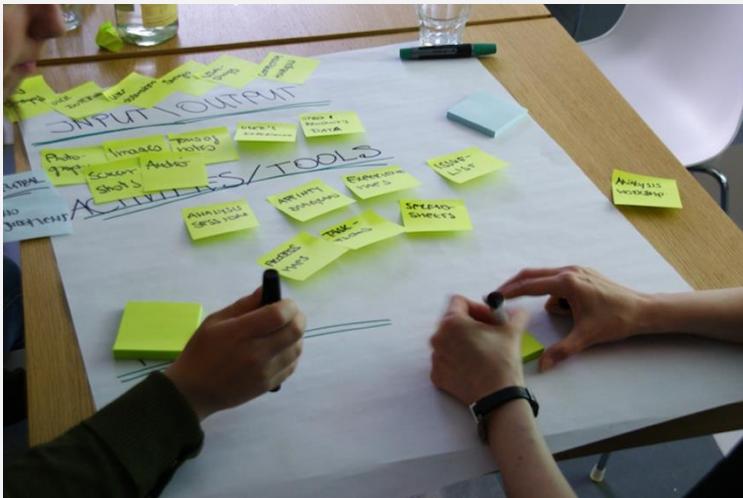
SCRUM Master



1. Assegurar el compliment de la metodologia.
2. Líder encarregada d'assegurar la qualitat del producte i els terminis d'execució.
3. Resol els conflictes entre l'equip de desenvolupament i el *product owner*.

Scrum: Meetings

Sprint Planning



1. Es realitza a l'inici de cada Sprint.
2. Durada: 1-2 h.
3. Objectiu: Planificar les funcionalitats que es realitzaran en la iteració.
4. Equips: Tots.



Scrum: Meetings

Daily Meetings



1. Es realitza a l'inici de cada dia.
2. Durada: 15-20 minuts.
3. Objectiu: Veure on som i què ens falta per fer. Desencallar problemes puntuals.
4. Equips: Scrum Master + Equip de Desenvolupament.

Scrum: Meetings

Sprint Review



1. Es realitza al final de cada *sprint*.
2. Durada: 1-2 h.
3. Objectiu: Ensenyar el producte al client. Recollir els nous canvis i funcionalitats.
4. Equips: Tots.

Scrum: Meetings

Sprint Retrospective



1. Es realitza al final de cada *sprint*.
2. Durada: 1-2 h.
3. Objectiu: Aprendre dels errors i dur a terme la millora contínua.
4. Equips: Equip de desenvolupament + Scrum Master.

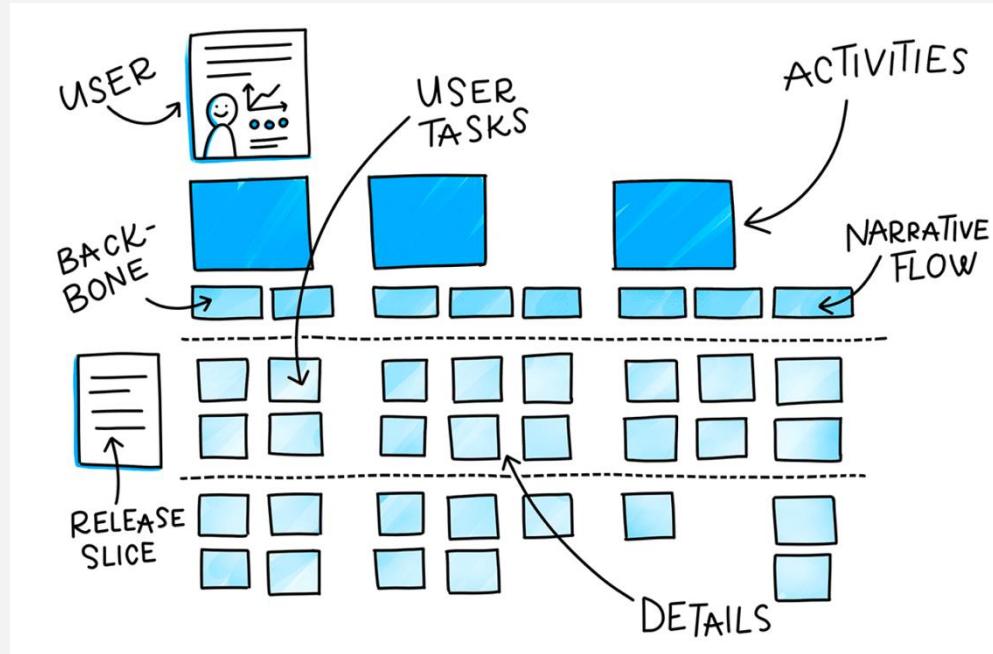
Scrum: Artifacts

Model Kanban

Example of a Kanban Board					
Backlog	In Progress (3)	Peer Review (3)	In Test (1)	Done	Blocked
Fast Track/ Defect					

1. Vista general del projecte i el seu estat.
2. Rol que el gestiona: Scrum Master.
3. Ús: Nivell intern per a l'equip de desenvolupament.

Scrum: User Story Mapping

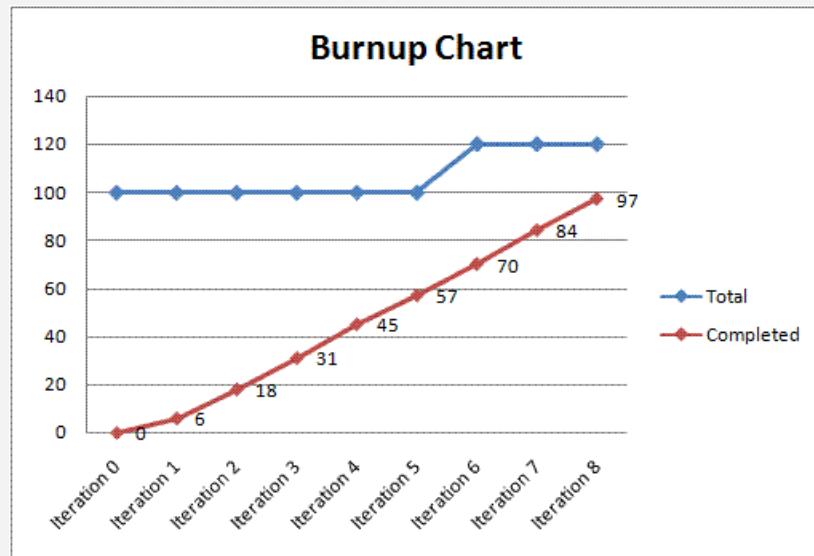


El **User Story Mapping** és una tècnica visual per organitzar i prioritzar el **Product Backlog** segons el flux real d'ús de l'usuari.

- Les **activitats principals** de l'usuari (a dalt, horitzontalment)
- Les **tasques o user stories** concretes (a sota, detallades)
- El **flux narratiu complet (narrative flow)** d'ús del producte
- **Els release slices** (talls verticals que defineixen què entra a cada versió)

Scrum: Artifacts

Grafic Burn-up

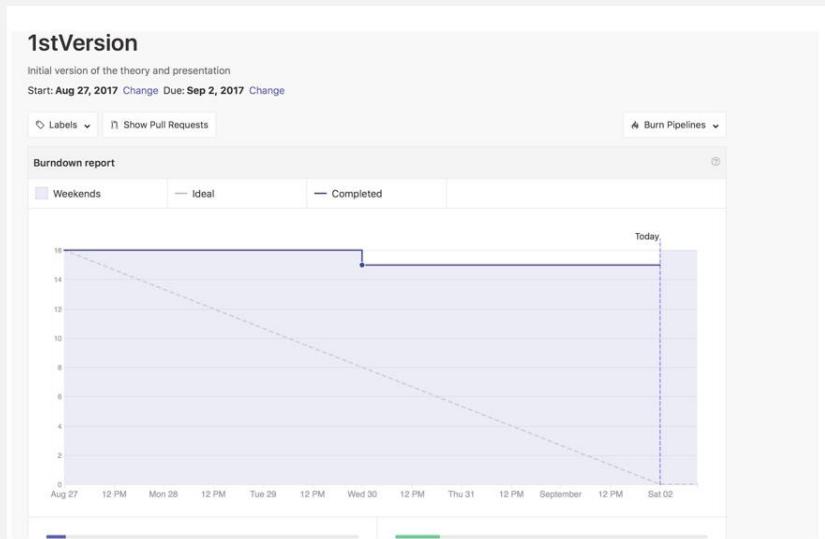


1. Veure la velocitat amb què tanquem *issues* al llarg de les iteracions.
2. Rol que el gestiona: Scrum Master.
3. Ús: Nivell intern per a l'equip de desenvolupament.



Scrum: Artifacts

Grafic Burn-down



1. Analitzar com gestionem la feina al llarg de l'sprint.
2. Rol que el gestiona: Scrum Master.
3. Ús: Nivell intern per a l'equip de desenvolupament.

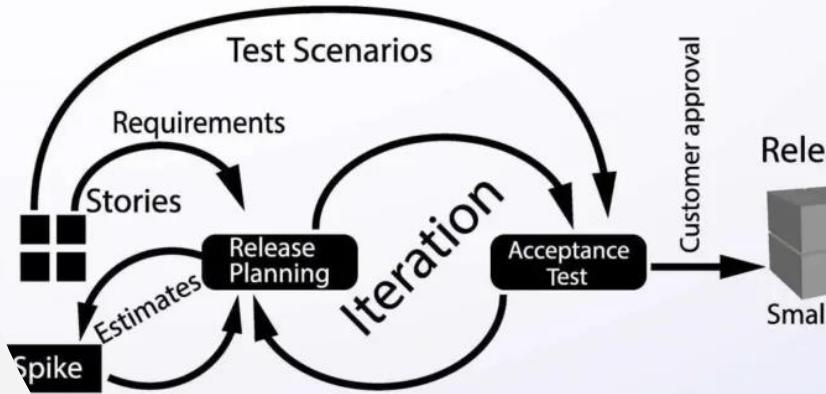
Extreme programming (XP)

Extreme Programming (XP) és un conjunt de tècniques per potenciar la satisfacció dels usuaris.

Tècniques orientades a gestionar el treball en equip per desenvolupar un programari alineat amb les necessitats i els requisits dels usuaris.



Extreme Programming



Pair Programming

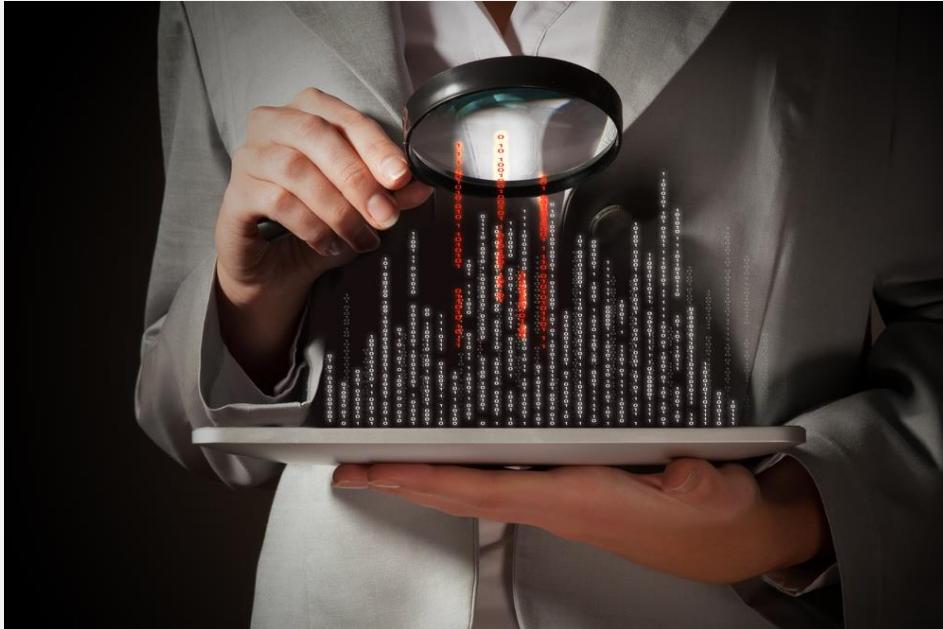


Dos desenvolupadors treballant junts en una mateixa màquina.

1. Programador. El que codifica.
2. Observador. Revisa el codi.

Cada cert temps, els rols s'intercanvien.

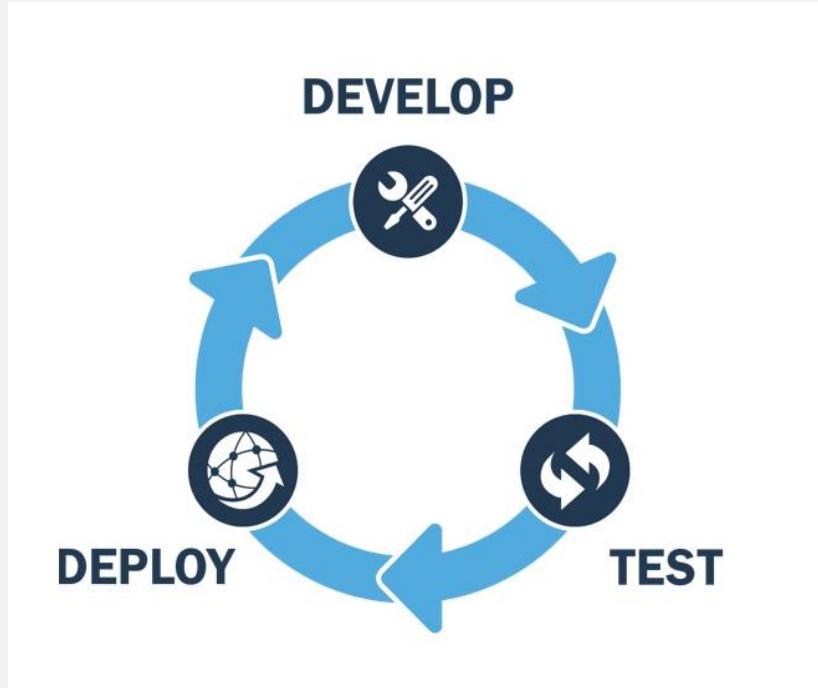
Peer Review



Revisió del codi per un equip de treball extern.

1. Equip de desenvolupament. Equip que desenvolupa el producte.
2. Equip de testing. Equip que prova i elabora l'informe de qualitat.

Continuos Integration (CI)



Pipelines i workflows per assegurar la qualitat del codi i desplegar automàticament.

1. **Desenvolupament.** Desenvolupament del codi i gestió sobre el repositori.
2. **Testeig.** Cada vegada que es fan *pull requests*, es realitzen els tests automàtics.
3. Si el testeig es realitza correctament, es fa un desplegament a l'entorn indicat.

Models de Extreme Programming: Continuos Integration (CI)

Jenkins



Travis



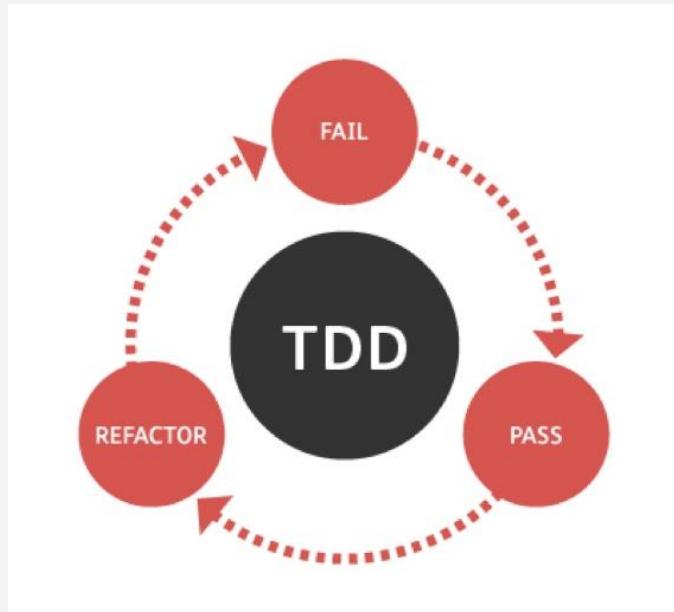
Circle
CI



Gitlab
CI



Models de Extreme Programming: Test Driven Development



Workflows per a la realització de testeig automàtic sobre el nostre producte.

1. **Codificació dels requisits.** Codificació dels requisits com a tests unitaris o històries d'usuari.
2. Generació del *pipeline* quan s'actualitza **GitHub**.
3. Si falla, ens envia un missatge amb els errors generats.

Models de Extreme Programming: Test Driven Development

JUnit *Cucumber*



Esta foto de Autor desconocido se concede bajo
licencia de [CC BY-SA-NC](#).

Esta foto de Autor desconocido se concede bajo licencia de [CC BY-SA](#).



