
Metodologies àgils

PID_00249600

Isabel Polo de la Llana
Jorge Fernández González

Temps mínim de dedicació recomanat: 5 hores



**Isabel Polo de la Llana**

Enginyera en informàtica per la Universitat Politècnica de Catalunya (UPC) i certificada en Scrum Master. Actualment treballa com a desenvolupadora de programari per a dades massives i anàlisi de dades en el sector privat.

**Jorge Fernández González**

Enginyer d'Informàtica per la Universitat Politècnica de Catalunya (UPC). Actualment és director de consultoria de *business intelligence* en Abast Solutions. Professor associat al Departament d'Enginyeria de Serveis i Sistemes d'Informació (ESSI) i responsable de l'assignatura Sistemes d'informació per a organitzacions de la Facultat d'Informàtica de Barcelona, tasques que compagina com a col·laborador docent a la UOC.



Índex

Introducció	5
Objectius	7
1. Ser àgil	9
1.1. Què vol dir <i>àgil</i> ?	9
1.2. Manifest i principis	9
2. Scrum	12
2.1. Presentació/finalitat i cicle de vida	12
2.2. Producte mínim viable	14
2.3. Transparència, revisió i adaptació	15
2.4. <i>Sprint</i>	16
2.5. Principis INVEST	16
2.6. Criteris DoR i DoD	17
2.7. Spike	18
2.8. Estimació i velocitat	18
2.9. Cancel·lació de l' <i>sprint</i>	19
2.10. Lliurament continu	19
3. L'equip Scrum	22
3.1. Scrum <i>master</i>	22
3.2. <i>Product owner</i>	23
3.3. L'equip de desenvolupament	23
4. Cerimònies Scrum	26
4.1. Refinament del <i>backlog</i> del producte	26
4.2. Reunió de planificació d' <i>sprint</i>	30
4.3. <i>Daily Scrum</i>	31
4.4. Revisió de l' <i>sprint</i>	32
4.5. Retrospectiva de l' <i>sprint</i>	32
5. Artefactes Scrum	36
5.1. El <i>backlog</i> del producte	36
5.2. Ítems del <i>backlog</i> del producte (PBI)	37
5.3. Tasques de l' <i>sprint</i>	38
5.4. <i>Backlog</i> de l' <i>sprint</i>	39
5.5. Increment	39
5.6. <i>Sprint burn down chart</i>	40
6. Més enllà d'àgil	42

6.1. <i>Squads</i> - Cas pràctic: Spotify	42
6.2. Més pràctiques àgils: Kanban, eXtreme Programming, DevOps	44
Resum	49
Exercicis d'autoavaluació	51
Solucionari	53
Glossari	54
Bibliografia	56

Introducció

Les bases de la metodologia àgil es defineixen per ser col·laboratives, iteratives i incrementals. Intenten donar pas a una nova era que deixa enrere metodologies de fases grans i llargues, sense pràcticament contacte entre negoci (clients i/o usuaris) i desenvolupadors.

Ser àgil no és, de cap manera, ser desordenat, ni desenvolupar grans productes en temps mínims, sinó tot el contrari. El marc estableix un ordre clar i concís de com procedir a desenvolupar productes donant el màxim valor per al negoci en el mínim temps possible. Per demostrar com, presentarem en detall el marc de treball (*framework*) Scrum, una de les metodologies àgils més utilitzades avui dia.

En aquest mòdul es descriuen amb detall els **rols** que componen un equip Scrum, les **cerimònies** que es duen a terme durant tot el procés de desenvolupament del producte i els **artefactes** que es fan servir per a dur a terme el marc de treball Scrum.

Al llarg del mòdul se seguiran exemples concrets amb un fil conductor comú. Aquests exemples il·lustren com és el desenvolupament d'un projecte programari complex seguint metodologies àgils dins el marc Scrum.

Presentació del fil conductor: Diffus

Diffus és una coneguda companyia de la indústria ortopèdica amb seu a Dinamarca que fabrica ortesis i pròtesis. Produeix dispositius que s'apliquen externament sobre el cos humà per a modificar o completar el funcionament i l'estructura del sistema musculoesquelètic i també aparells externs usats per a reemplaçar membres absents o amb deficiències.

Després de saltar a la fama, el 2015 va reconèixer que estava tenint greus problemes per negociar algunes matèries primeres amb diferents proveïdors a escala mundial. L'empresa ha anat creixent exponencialment i el seu negoci s'ha ampliat any rere any, tot i que la seva gestió empresarial ha quedat una mica obsoleta per a la dimensió actual.

Les seves seves distribuïdes per tot el món han anat negociant de manera particular amb diversos proveïdors, però aquesta informació no s'ha centralitzat en cap sistema, ni en l'àmbit estatal ni en l'internacional, només en l'àmbit de la seu. A causa d'aquest fet, estan detectant que un mateix proveïdor factura diferents preus per als mateixos tipus de lots en seus diferents, o fins i tot per a seus localitzades dins del mateix país.

L'equip de direcció s'està plantejant renegociar determinats productes, però no té clar per quin començar. No volen malgastar temps ni diners a negociar el preu de matèries primeres que no siguin rellevants en el cost final del producte.

A més, en les cadenes de muntatge estan disponibles unes màquines dispensadores d'eines que els treballadors fan servir diàriament per completar tasques manuals molt específiques que requereixen precisió humana per a completar la producció d'alguns productes. Aquestes eines, realment cares majoritàriament, també es compren en lots i són facturades de diverses maneres. Així doncs, també estan interessats a controlar quines eines es fan servir més, cada quant són reemplaçades, a quines hores o torns es fan servir, quins treballadors les fan servir i qui les retorna a la màquina expendedora, etc.

La direcció s'ha plantejat un objectiu clar i concís: controlar els costos de facturació de qualsevol matèria primera o eina adquirida a Europa.

Si el projecte funciona com s'espera, no tenen dubte que invertiran més recursos per estendre el mateix sistema a escala mundial, ja que necessiten tenir tota la informació disponible per a prendre decisions informades.

Tenen en ment una idea clara: creuen que un quadre de comandament (*dashboard*) pot solucionar la centralització d'informació originària de múltiples fonts. El quadre de comandament hauria de complir diversos objectius:

- generació de gràfics
- interacció disponible per a analitzar dades
- anglès com a idioma central
- comparació personalitzada d'eines i matèries primeres per lots i geolocalitzacions
- moneda configurable (\$, £, €...)
- diferents accessos segons l'usuari

Aquest projecte ha estat assignat a un equip de desenvolupament que treballa amb el marc de treball Scrum. Al llarg d'aquest mòdul descobrirem com gestionen el projecte mitjançant Scrum: rols, cerimònies, artefactes, etc.

Objectius

L'objectiu d'aquest mòdul és adquirir coneixements sobre les metodologies àgils i tenir una comprensió més profunda del marc de treball Scrum. Més concretament, els objectius són els següents:

- 1.** Conèixer les bases del manifest àgil.
- 2.** Comprendre què és la metodologia Scrum i quins són els seus objectius.
- 3.** Conèixer el cicle de vida Scrum.
- 4.** Diferenciar els rols Scrum i quines són les seves responsabilitats.
- 5.** Identificar les cerimònies Scrum i per a què serveixen.
- 6.** Conèixer altres metodologies àgils.

1. Ser àgil

1.1. Què vol dir àgil?

El desenvolupament de programari àgil, de l'anglès *agile software development*, és un terme que s'utilitza per a descriure diversos mètodes de desenvolupament basant-se sempre en fases col·laboratives, iteratives i incrementals. Té com a objectiu donar valor al negoci com més aviat millor; és per això que les etapes solen ser curtes i intenten interaccionar amb el negoci per obtenir una retroacció activa i contínua al llarg de tot el desenvolupament.

Lectura recomanada

A l'enllaç següent es pot trobar informació addicional sobre el manifest àgil.

<http://agilemanifesto.org>

1.2. Manifest i principis

El febrer de l'any 2001, un grup de disset desenvolupadors de programari es va reunir per discutir sobre les tècniques i els processos de desenvolupament que s'havien seguit fins al moment. Com a resultat, es va definir llavors un concepte: el de les metodologies àgils, llunyanes dels procediments rígids i pesats que dominaven a l'època. S'assenten, com a resultat, les bases del manifest àgil:

Bases del Manifesto for Agile Software Development

Individuals and interactions over processes and tools.

Working software over comprehensive documentation.

Customer collaboration over contract negotiation.

Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.

A continuació, es descriu cadascun d'aquests valors amb més detall.

1) Més valoració dels individus i de les seves interaccions que no dels processos i les eines

Es busca comptar amb persones que participin en equips motivats i autònoms, que interactuïn efectivament entre ells i que obtinguin resultats singulars que quedin lluny del que es pot arribar a obtenir seguint altres procediments.

2) Programari que funciona per sobre de documentació extensa

El plantejament se centra a satisfer els *stakeholders* amb programari que funciona, que dona valor al negoci per mitjà de petits increments que són íntegrament funcionals. Només és estrictament necessària la documentació que dona valor als *stakeholders* i que, per tant, ha de ser actualitzada a mesura que el producte adquireix noves característiques.

Stakeholder

Persona que té un interès a obtenir el desenvolupament d'un producte, bé perquè desitja usar-lo o perquè serà afectat per ell.

3) Col·laboració amb el client abans que negociació d'un contracte

S'entén que la relació entre client i proveïdor és enterament cooperativa i que aquesta és l'única manera d'obtenir un producte desenvolupat col·laborativament de manera efectiva. Òbviament, hi ha negociacions i temes legals que estableixen de manera formal el contracte, però el seu objectiu se centra a satisfer les dues parts donant lloc a la prioritització de necessitats i a la revisió del producte.

4) Resposta davant els canvis per sobre del seguiment d'un pla

Per diversos motius, les necessitats i les prioritats del producte canvien constantment, ja sigui abans de començar el desenvolupament, durant algunes iteracions i fins i tot al final. Gràcies a l'adaptació als canvis durant tot el cicle, s'evita desenvolupar un producte que pugui quedar lluny de les necessitats dels *stakeholders* i que finalment no sigui utilitzat.

Si ens basem en els anteriors valors descrits al manifest àgil, s'estableixen dotze principis bàsics:

Principis bàsics

- 1. Satisfacció del client:** la major prioritat és satisfer el client mitjançant el lliurament de productes que aportin valor des de l'inici del projecte.
- 2. Els canvis de requisits són benvinguts:** es dona cabuda que l'usuari provi el producte al llarg del seu desenvolupament i al fet que es modifiquin els requisits si, d'aquesta manera, el producte aportarà més valor per al negoci.
- 3. Lliuraments freqüents:** es fan lliuraments funcionals tan freqüents com sigui possible.
- 4. Treball col·laboratiu:** l'equip de desenvolupament i el grup responsable del negoci mantenen una relació propera i habitual.
- 5. Confiança i motivació:** els productes són elaborats per individus motivats dins d'un entorn en què se sentin protegits i confortables.
- 6. Comunicació freqüent i presencial:** la comunicació tant dins l'equip de desenvolupament com amb els responsables del negoci es fa cara a cara, tan freqüentment com sigui possible.
- 7. Progrés estimable:** l'increment que afegeix funcionalitats i valor a un producte és el principal indicador de progrés.
- 8. Desenvolupament sostenible:** l'equip en conjunt ha de mantenir un ritme constant indefinidament.
- 9. Atenció a l'excel·lència:** l'excel·lència tècnica i la qualitat del disseny es tenen en compte al llarg de tot el procés, i això en millora l'agilitat.
- 10. Simplicitat:** és essencial a l'hora de minimitzar tasques.
- 11. Autoorganització:** els equips autoorganitzats originen millors arquitectures, requisits i dissenys.
- 12. Adaptació:** l'equip delibera regularment sobre la seva efectivitat i sobre com es pot millorar de cara al futur.

2. Scrum

El terme *scrum* (comunament pronunciat tant *scrum* com *scram* en català, fonèticament representat com /skr^m/) apareix per primer cop a l'article «The New New Product Development Game» (el nou nou joc del desenvolupament de productes), redactat per Hirotaka Takeuchi i Ikujiro Nonaka i publicat a *Harvard Business Review* el 1986. S'hi comparen equips funcionals amb la formació melé, de l'anglès *scrum*, que es fa servir en rugbi. L'article es considera la inspiració del marc de treball Scrum, encara que no hi fa referència directa.

Més tard, l'any 1993, Jeff Sutherland defineix *scrum* com a procés a Easel Corporation, una empresa de desenvolupament de programari. I només dos anys més tard, el 1995, juntament amb Ken Schwaber, presenta el marc de treball Scrum a la conferència OOPSLA'95, acrònim d'*object oriented programming systems, languages and applications*.

2.1. Presentació/finalitat i cicle de vida

Scrum és un marc de referència usat per al desenvolupament de programari des de principis dels anys noranta. No es tracta d'un procés que s'ha de seguir, sinó d'un marc que dona cabuda a diferents tècniques. Aquest marc es caracteritza per definir etapes incrementals, ser flexible i respondre a situacions canviants. S'usa comunament per a projectes complexos en què no és possible predir totes les situacions, funcionalitats i necessitats que poden esdevenir en el futur. Es recomana per a equips multifuncionals i autogestionats d'entre tres i deu persones cadascun.

Scrum està compost per:

- Tres tipus de **rols** dins d'un equip: Scrum *master*, *product owner* (client) i desenvolupador.
- Una sèrie d'**artefactes**, com el *backlog* del producte.
- Un conjunt de **cerimònies**: refinament del *backlog*, planificació de l'*sprint* (iteració), *daily Scrum*, revisió de l'*sprint* i retrospectiva de l'*sprint*.

Tots aquests elements es descriuen amb més detall a les seccions següents.

En Scrum es defineix una durada fixa per a cada etapa, anomenada **sprint** o **iteració**, després de la qual s'ha d'haver desenvolupat un increment del producte que és potencialment lliurable. La durada de cada etapa o iteració pot

Lectura recomanada

Hirotaka Takeuchi; Ikujiro Nonaka (1986, gener). «The New New Product Development Game». *Harvard Business Review*.

Qui fa servir Scrum?

Empreses conegudes i de renom reconeixen que segueixen i fins i tot innoven dins el marc Scrum:

- Spotify
- Toyota
- Booking.com
- Amazon
- Google

ser d'entre una i quatre setmanes. Les regles que regeixen el marc Scrum ajuden a unir tots els elements (rols, cerimònies i artefactes) de manera simple d'entendre, tot i que difícil de dominar.

Scrum es caracteritza per ser flexible i adaptar-se als canvis, però al seu torn hi ha establerts punts de control diaris i punts de control després de cada iteració, per així assegurar el resultat del producte.

El cicle de vida Scrum comença amb la redacció d'una llista de necessitats organitzades per ordre de prioritats, el **backlog** del producte. S'hi indica què cal emprendre, però no s'hi indica com. Quan la llista està finalitzada amb tots els requisits necessaris, comença el primer *sprint*. Dins de cada *sprint* es desenvolupen alguns dels ítems del *backlog* del producte. Quan un *sprint* s'acaba, en comença un de nou, ja que la base d'Scrum és la iteració. A grans trets, se segueix el procés següent:

- a) El **product owner** redacta completament el *backlog* del producte.
- b) Durant la reunió de **refinament de l'*sprint***, l'**equip de desenvolupament** analitza el **backlog del producte** i selecciona quins ítems es poden incloure en l'increment d'una versió potencialment lliurable al final del primer **sprint**. Es negocia entre ambdues parts, **product owner** i **desenvolupadors**, si hi ha algun conflicte a causa de prioritats o dependències.
- c) L'**sprint** comença i l'equip de desenvolupament s'autogestiona per desenvolupar els ítems inclosos a la iteració. Assisteixen a **standups** diaris juntament amb el **product owner** i l'**Scrum master** per aclarir dubtes, resoldre impediments, proposar alternatives més adequades, etc.
- d) Al final de la iteració té lloc la **revisió de l'*sprint***, on l'equip sencer es reuneix per fer una demostració de l'increment que ha estat desenvolupat durant l'*sprint*.
- e) En darrer lloc es fa una **retrospectiva de l'*sprint***, que dona lloc a millorar el procés seguit en el desenvolupament, en la gestió, en la demostració, en la comunicació o en qualsevol altre element que l'equip vulgui valorar.
- f) El procés sencer a partir del punt *b* es repeteix tantes vegades com sigui necessari, fins a obtenir un producte que compleixi els resultats esperats pel negoci.

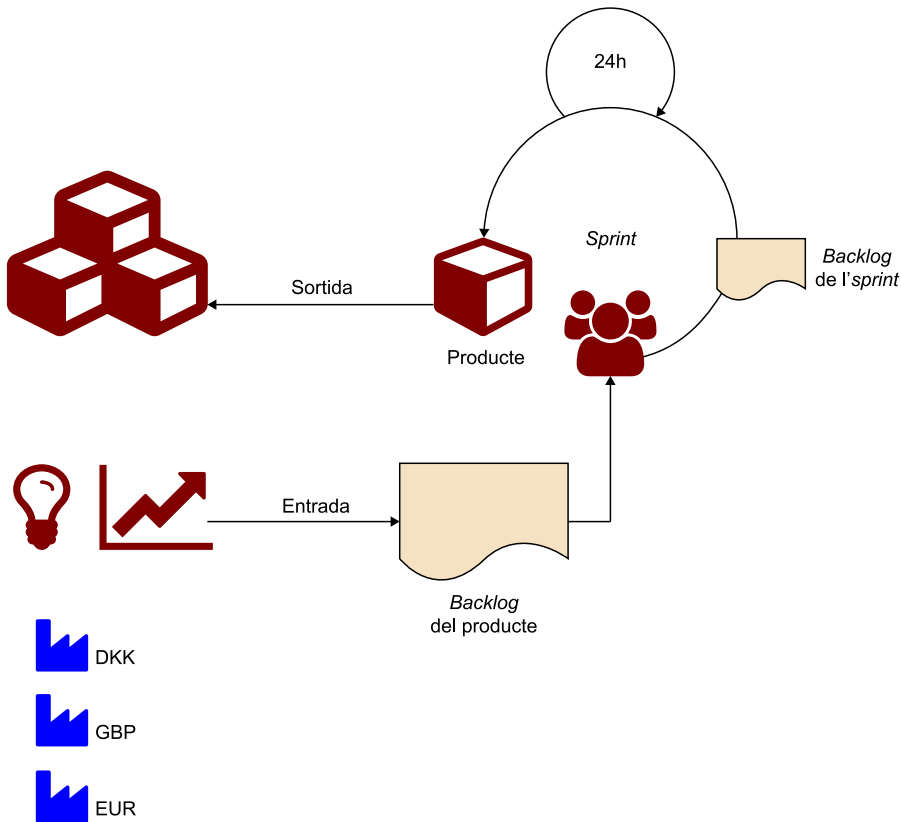
Product owner

Aquest paper és l'encarregat de representar les necessitats del negoci. Està definit amb més detall a l'apartat 3.2.

Standup

Reunió diària, de quinze minuts com a màxim, en què tots els membres de l'equip parlen de les tasques en què estan treballant i si tenen impediments per a dur-les a terme. Això està definit amb més detall a l'apartat 4.3.

Figura 1. Procés d'Scrum genèric



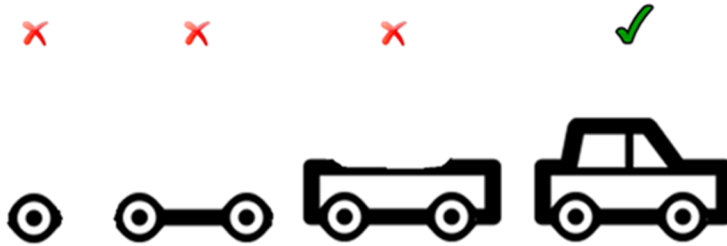
D'aquesta manera, Scrum permet a l'equip de desenvolupament gestionar la seva pròpia feina. Quan s'és responsable d'escollir i de prendre decisions pròpies, el treball es converteix en un desafiament que se salda amb idees interessants i originals que no podrien haver-se planificat des d'un inici.

2.2. Producte mínim viable

Scrum se centra a lliurar constantment un producte mínim viable (de l'anglès *minimum viable product*, MVP). Això vol dir que a cada iteració se li lliura al negoci un producte que compleix íntegrament algunes de les necessitats que han detectat i que desitgen cobrir.

Figura 2. Com desenvolupar un producte mínim viable

Evitar models



Potenciar models



Scrum no només se centra a lliurar productes que donin valor al negoci com més aviat millor, sinó que també té en compte quines són les prioritats del negoci. Una de les accions que es fan a Scrum és elaborar una llista de necessitats i funcionalitats, i l'equip de desenvolupament decideix quins seran incloses i desenvolupades en cada etapa per maximitzar el valor lliurat. Gràcies a la retroacció constant del negoci, es realitzen contínues redefinicions de les necessitats per a ajustar-se completament a l'objectiu que realment es vol obtenir.

2.3. Transparència, revisió i adaptació

Scrum té com a pilars bàsics:

- la transparència
- la revisió
- l'adaptació

1) Transparència

L'entorn en què treballa l'equip de desenvolupament ha de facilitar la interacció i la comunicació entre els membres perquè tots siguin conscients del volum de treball, de la dificultat de les tasques, de qui és responsable de cada element i de quins problemes i impediments s'estan tenint. Per això, se segueixen uns estàndards definits dins el propi equip, com ara detallar què es considera com a tasca finalitzada o quan una tasca està a punt per a ser executada. El llenguatge usat ha de ser compartit i entès per tots els participants de l'equip.

2) Revisió

Scrum anima a inspeccionar el progrés diari per detectar desviaments de l'objectiu desitjat. Gràcies a les reunions diàries i a les revisions de *sprint*, l'equip no només millora la comunicació i la transparència, sinó que s'hi afegeixen punts de revisió, fet que dona lloc a una millora contínua del procés i del producte. L'equip detecta en cada revisió si s'està duent a terme un desenvolupament que dona valor al producte o, al contrari, si s'allunya dels objectius sense donar cap valor i es necessita alguna mena de rectificació.

3) Adaptació

Si durant alguna de les revisions (*daily Scrum*, revisió de l'*sprint* o retrospectiva de l'*sprint*) es detecta que el desenvolupament es desvia de l'objectiu de l'*sprint* i no es dona el valor esperat al producte, cal adaptar-s'hi tan aviat com sigui possible. D'aquesta manera, s'evita malbaratar recursos i temps per a obtenir un resultat que no és l'esperat.

En aquesta anàlisi no només es readapta el desenvolupament segons les necessitats, sinó que es redefeixen processos si, amb això, s'obté un benefici dins de l'equip. De vegades, també poden redefinir criteris o necessitats a mesura que el producte pren forma.

2.4. *Sprint*

El nucli d'Scrum és l'*sprint*. Un *sprint* és una fase de durada fixada, normalment entre una i quatre setmanes, en què l'equip desenvolupa part d'un producte que és potencialment lliurable. L'*sprint* també s'anomena comunament **iteració**. Un *sprint* sempre té la mateixa durada al llarg de tot el desenvolupament del producte i després d'acabar un el següent comença sense demora l'endemà.

2.5. Principis INVEST

Un *sprint* pretén desenvolupar un o més dels ítems descrits al *backlog*, anomenats **story** o *product backlog item* (PBI). L'acrònim INVEST ens permet recordar els criteris de qualitat que hem de tenir en compte per definir una *story*:

INVEST

- *Independent*
- *Negotiable*
- *Valuable*
- *Estimable*
- *Small*
- *Testable*

Story

Ítem del *backlog* que descriu un escenari amb criteris d'acceptació. Definit amb més detall a l'apartat 5.2.

A continuació, es descriuen cada un dels principis INVEST:

- Independent: cada *story* ha de ser autocontinguda, és a dir, independent una de l'altra.
- Negociable: la funcionalitat descrita ha de ser negociable i flexible, i ha de donar lloc a la discussió en cas que calgui.
- Valuós: l'*story* ha d'aportar valor quantificable al producte.
- Estimable: l'esforç de desenvolupament d'una *story* ha de ser mesurable.
- Petit: l'*story* ha de ser prou petita per a poder predir la seva complexitat i completar-la en una sola iteració.
- Testeable: ha de ser factible comprovar si s'ha completat satisfactòriament o no.

2.6. Criteris DoR i DoD

L'equip ha de decidir quins criteris han de ser considerats i complets perquè una *story* sigui apta per a ser escollida i inclosa en alguna de les iteracions. Aquests criteris es coneixen com DoR, de l'anglès *definition of ready* (definició de preparat). Hi ha equips que consideren útil tenir una DoR al projecte i una altra DoR al *sprint*.

Alguns exemples de DoR en l'àmbit de projecte podrien ser:

- Fonts de dades identificades i amb accés permès. En el cas que hi hagi *stories* que requereixin accés a dades, no podran realitzar-se si no es té accés a les dades.
- *Stakeholders* identificats. Alguns *stakeholders* solen participar en les reunions de revisió de l'*sprint* i utilitzar el producte que es desenvolupa. Si estan identificats, seran convidats i podran tenir accés als increments del producte.
- Calendari de reunions acceptat per tot l'equip. Tot l'equip ha de ser conscient de quan tindran lloc les cerimònies i haurà de planificar-se en conseqüència.
- Eines per al desenvolupament instal·lades i funcionant.
- Accés als entorns de desenvolupament permès.

Alguns exemples de DoR en l'àmbit d'*sprint* poden ser:

- històries d'usuari revisades i catalogades segons esforç requerit

- tasques i subtasques creades i estimades
- calendari de l'*sprint* actualitzat

DoD, de l'anglès *definition of done* (definició d'acabat), consisteix en una llista d'activitats que, un cop dutes a terme, fan que es compleixin tots i cadascun dels criteris d'acceptació de l'usuari. S'ha d'aconseguir la DoD per donar valor al producte i assegurar que compleix la qualitat. Si al final de l'*sprint* algun dels punts de DoD està incomplet o sense complir, l'*story* es considera no finalitzada.

Alguns exemples de DoD podrien ser:

- test unitari funcionant
- revisió del codi realitzada
- desplegament en entorn de test realitzat
- funcionalitat verificada pel *product owner*

Tant els criteris DoR com els DoD han de ser acordats entre els membres de l'equip i estar explícits en algun lloc comú de l'entorn on l'equip treballi. Les llistes poden ser tan detallades com l'equip ho consideri, i se solen redactar (o actualitzar si és un equip que ha treballat prèviament en altres projectes) abans de començar el projecte o *sprint*.

2.7. Spike

A vegades hi ha necessitats específiques que difícilment poden ser desenvolupades donada la seva complexitat o el desconeixement sobre el tema. Una *spike story* és una *story* que requereix una investigació tècnica o funcional (durant un temps limitat i prefixat) per a poder tenir una estimació de la seva durada i/o poder dur-la a terme.

2.8. Estimació i velocitat

Mike Cohn, un dels grans col·laboradors d'Scrum, va crear el terme *story points*. La mida de cada *story* s'estima amb **punts depenent de l'esforç** que l'equip necessitarà fer per a poder completar l'*story*, no en la complexitat que presenti.

«Story Points are not about the complexity of developing a feature; they are about the effort required to develop a feature.»

Mike Cohn

Assignant punts a cada *story* es busca relativitzar l'esforç requerit per a cadascuna d'elles. Normalment s'assignen punts a les stories seguint la seqüència de Fibonacci: com menys esforç, menys valor i, per tant, menys grandària.

Seqüència de Fibonacci

Seqüència de nombres enters els dos primers termes de la qual són 0 i 1. Cada terme a partir del tercer s'obté com la suma dels dos termes anteriors: 0, 1, 1, 2, 3, 5, 8, 13...

Taula 1. Assignació de punts en funció de l'esforç

Story points	Esforç
0	story realitzada
1, 2, 3, 5, 8, 13	story petita (S), mitjana (M), gran (L)
20, 40, 100	story molt gran que necessita ser dividida en stories més petites per poder dur-se a terme en diferents increments
?	manca de coneixement per a estimar
∞ (infinit)	massa gran, no hi ha capacitat

Al final de l'*sprint* és idoni comptabilitzar els punts de totes les *stories* lliurades complint la DoD per l'equip de desenvolupament. Aquest còmput s'anomena **capacitat d'entrega de l'equip** o **velocitat**. Si aquest càlcul es fa *sprint* rere *sprint*, pot ajudar a fer prediccions sobre la capacitat que l'equip sol tenir. És una predicció que pot ser completa o no segons les condicions del projecte i l'entorn en què se situa, òbviament: mai s'ha de prendre com a referència fixa.

2.9. Cancel·lació de l'*sprint*

A vegades passa que l'*sprint* és cancel·lat. Les raons poden ser diverses:

- L'objectiu de l'*sprint* ha quedat obsolet.
- El *product owner* considera que no es pot afegir més valor al producte.
- No hi ha més finançament per a la realització de l'*sprint*.
- Els ítems que queden al *backlog* del producte han deixat de tenir valor.
- Part considerable de l'equip ha estat reassignat a un altre projecte.

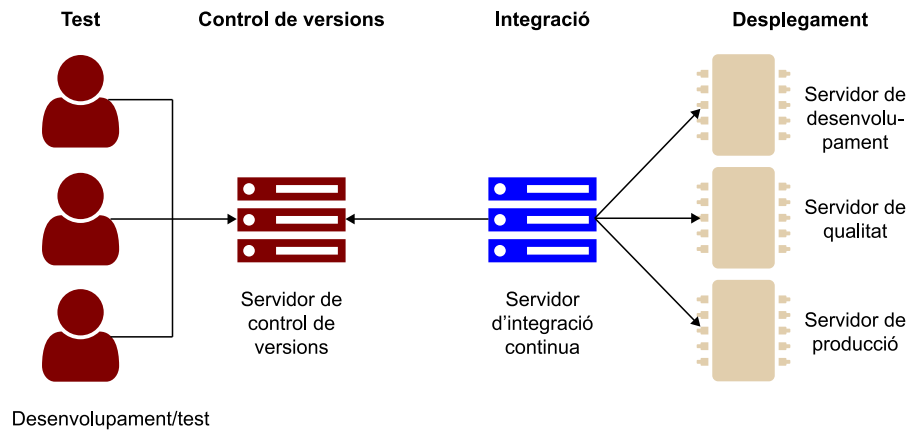
Només el *product owner* té autoritat per a cancel·lar un *sprint*, tot i que pot ser degut a influències dels clients finals del producte.

2.10. Lliurament continu

Amb l'objectiu de millorar la qualitat del programari i reduir el risc relacionat, el lliurament continu (en anglès, *continuous delivery* o CD) és un concepte fonamental que pren importància dins dels desenvolupaments àgils. Aquesta pràctica requereix la incorporació de codi en un espai comú diverses vegades al dia, perquè sigui verificat i finalment empaquetat i apte per a ser lliurat tan freqüentment com sigui necessari.

El lliurament continu està basat en quatre pilars:

Figura 3. Exemple de procés d'integració contínua



a) **Test:** cada vegada que s'afegeixen nous desenvolupaments es modifica part del codi actual o s'eliminen funcionalitats obsoletes, etc., és ideal verificar que cada component del sistema funciona correctament, de manera autònoma i íntegrament en conjunt. Cal comprovar que no ha deixat de funcionar res previ al desenvolupament actual, i també que el lliurable funciona correctament en diferents entorns, etc. Per això, l'etapa de desenvolupament es complementa amb la de test, i fins i tot en processos més estrictes (coneguts com *test-driven development* o TDD) es defineixen i implementen primer o en paral·lel els tests per desenvolupar així codi que compleixi amb ells des d'un principi. És primordial que aquesta etapa sigui robusta per a evitar el descobriment d'errors en etapes més avançades sempre que sigui possible, ja que en aquest cas es complica el desenvolupament, cosa que fa que calgui tornar enrere per fer-hi correccions que al seu torn poden causar nous defectes.

b) **Control de versions:** també és important mantenir un versionat del desenvolupament realitzat. Normalment, això és gestionat gràcies a un repositori de control de versions que permet la comparació, fusió, ramificació, etc. de codi segons sigui definit per l'equip de desenvolupament o l'entorn on treballen. El control de versions permet versionar tant desenvolupaments en paral·lel com desenvolupaments comuns, i facilita un historial clar i concís del codi afegit.

c) **Integració:** la fase d'integració és primordial per a la verificació de programari amb diferents components, que han d'interactuar i que poden haver de desplegar-se en diferents entorns. L'execució dels mateixos tests de desenvolupament és positiva i fins i tot pot afegir-se un conjunt nou de tests per verificar les noves característiques tenint en compte l'entorn on residirà el codi.

d) **Desplegament:** el procés d'integració finalitza amb la preparació d'un paquet, un conjunt de fitxers o l'artefacte que sigui convenient per a poder realitzar, *a posteriori*, el desplegament del producte en els diferents entorns que siguin necessaris.

És beneficiós que aquests quatre passos s'automatitzin tant com sigui possible, ja que això facilita que el procés de lliurament sigui tan freqüent com convenient, mentre que es minimitza el risc de lliurar una versió amb errors o amb alguna característica que deixi de funcionar pels canvis al codi.

3. L'equip Scrum

L'equip Scrum és multifuncional, està autoorganitzat i es compon de tres únics rols: Scrum *master*, *product owner* i equip de desenvolupament. En total, solen ser entre tres i deu persones, cadascuna amb un únic paper. L'equip Scrum s'encarrega de produir resultats per als *stakeholders*, més coneguts com els interessats (o part interessada del negoci), que solen ser inversors, clients o usuaris finals. Els *stakeholders* no formen part de l'equip Scrum.

Ja que cada perfil de l'equip Scrum té diferents responsabilitats, és altament recomanable que cada membre de l'equip tingui un únic rol, ja que en cas contrari els interessos de cada un es poden veure afectats si hi ha conflictes entre diferents rols per al mateix membre. Generalment, l'equip sencer està assignat al mateix projecte i pràcticament tots els membres estan designats a contribuir-hi a temps complet.

3.1. Scrum *master*

El paper de l'Scrum *master*, o SM, és primordial per a mantenir un equip que segueixi valors i principis Scrum. És conegut com el «facilitador», ja que:

- instrueix l'equip sobre Scrum,
- assegura la correcta execució d'Scrum,
- facilita la comunicació entre l'equip,
- protegeix l'equip del desenvolupament de possibles intrusions o interrupcions,
- manté un entorn de treball adequat fent visibles tots els artefactes,
- organitza i controla les cerimònies Scrum,
- recull informació al llarg dels *sprints* per realitzar prediccions,
- resol impediments,
- no té autoritat sobre l'equip,
- ajuda i instrueix l'organització a adoptar o potenciar Scrum.

Scrum *master* a Diffus

En Klaus ha treballat en diferents perfils i a diverses empreses tant com a desenvolupador com a Scrum *master*. En el transcurs del seu pas per Diffus ha treballat en dos petits projectes –dins d'altres àrees de l'empresa. Ha tingut prou temps per a conèixer el funcionament de l'empresa i quins productes fabrica, quines eines i tecnologies fan servir els desenvolupadors i quin és el problema a què s'enfronta l'empresa.

3.2. Product owner

El *product owner*, o PO, és el perfil encarregat de maximitzar el valor del producte. És el paper que comprèn les necessitats dels *stakeholders*, busca finançament i decideix quines característiques i funcionalitats són indispensables perquè el producte sigui desenvolupat i pugui ser lliurat. És a dir, és el vincle directe entre *stakeholders* i l'equip de desenvolupament.

PBI

De l'anglès *product backlog item*. Definit amb més detall a l'apartat 5.2.

Les responsabilitats d'aquest paper són les següents:

- Crear el *backlog* del producte.
- Organitzar per ordre de prioritat els ítems del *backlog*, de manera que es maximitzi el valor del producte.
- Assegurar que el *backlog* és clar i és entès per tots els membres.
- Informar i actualitzar el *backlog* del producte amb noves necessitats o canvis de criteris d'acceptació.
- Mantenir sempre en ordre de preferència els PBI per ser inclosos en cada increment.
- Aclarir dubtes a l'equip de desenvolupament respecte a funcionalitats o característiques del producte.
- Decidir si els PBI han estat completats satisfactòriament i compleixen amb el que s'esperava.

Product owner a Diffus

La Fiona ha estat assignada *product owner* del projecte. Fa molt de temps que treballa a Diffus i té un ampli coneixement de l'empresa. Ha treballat en diferents llocs, tots ells relacionats amb la part financera. Comprèn a la perfecció com funcionen totes les seus ubicades a Dinamarca i, a més, té contactes propers a les altres seus europees.

Tot i que Diffus té repartides actualment seus per tot el món, ha decidit centrar-se en l'àmbit europeu per a aquest projecte. Si els resultats donen un valor notable a l'empresa, el projecte rebrà més recursos per dur-se a terme a escala mundial. Fins ara, els projectes d'anàlisis financeres que s'han dut a terme en l'àmbit de la seu són bastant deficients, ja que no tenen informació estandarditzada i són clarament insuficients per a contrastar dades. La Fiona té nombroses esperances de descobrir, gràcies a aquest projecte, què està passant en les finances actuals.

3.3. L'equip de desenvolupament

L'equip de desenvolupament, o ED, és responsable de produir un increment en cada iteració d'acord amb la descripció de DoD (*definition of done*). És el mateix equip el que decideix quins PBI són inclosos en cada *sprint* i com s'escometen. L'equip de desenvolupament és multifuncional i autoorganitzat. Es recoma-

nen **equips** més aviat petits, no menys de tres, ja que és fàcil que a l'equip de desenvolupament li falti experiència en algun àmbit, i al seu torn no més de vuit, ja que requeriria massa coordinació i gestió. Així doncs, l'equip de desenvolupament inclou entre tres i vuit persones, i té les característiques següents:

- Està compost per diferents rols, tots els necessaris per a crear apropiadament un increment.
- Tots els membres són considerats per igual com a desenvolupadors, no hi ha jerarquia interna a l'equip.
- L'equip és autònom i s'autogestiona internament. Ningú decideix quins PBI són escollits en cada *sprint*, ni qui els executa, ni com, excepte el mateix equip de desenvolupament.
- Negocia amb el *product owner* quins PBI són inclosos en cada *sprint*.
- Manté una comunicació oberta entre tots els membres, sense jerarquies ni subgrups.

Desenvolupadors a Diffus

S'han assignat cinc desenvolupadors a aquest projecte. Així doncs, tenim els desenvolupadors següents:

- John: amb àmplia experiència en l'anàlisi de dades i en el testeig d'aplicacions.
- Manuel: la seva experiència anterior es basa en desenvolupament web, sempre *front end*.
- Zenzie: sènior en el sector de les tecnologies de la informació, sense experiència en l'anàlisi de dades.
- Xin: està finalitzant els seus estudis universitaris, ha participat a temps parcial en diversos projectes de BI.
- Louise: situada físicament a Bèlgica, a una oficina remota, té àmplia experiència amb projectes similars.

Com es pot veure, l'experiència de l'equip és variada i en diferents sectors, i fins i tot en diferents localitzacions. Tot i això, tots participaran a les reunions que més endavant es descriuen, ja que cada un dels membres aporta diferents experiències i idees, a més de diferents perspectives, la qual cosa és clau per a l'èxit del projecte.

Figura 4. Equip de desenvolupament multifuncional a Diffus

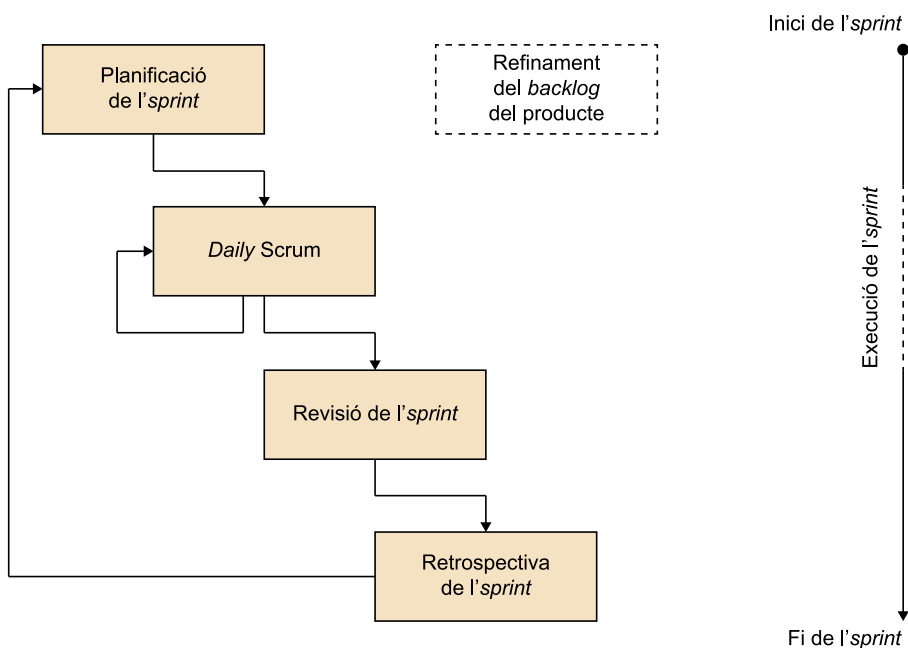


4. Cerimònies Scrum

Dins el marc Scrum hi ha un conjunt de reunions que es desenvolupen al llarg del tot el procés de creació d'un producte. Totes les reunions, més conegudes com cerimònies, són organitzades per l'*Scrum master*, tot i que aquest no té autoritat a l'hora de prendre decisions en elles.

El flux de cerimònies es pot veure al gràfic següent:

Figura 5. Flux de cerimònies Scrum dins d'un *sprint*



4.1. Refinament del *backlog* del producte

A vegades els PBI són força genèrics, no tenen detalls en els criteris d'acceptació o varien de prioritat depenent de les necessitats del negoci. Durant el refinament del *backlog* del producte (també anomenat *backlog refinement*, *backlog grooming*, *backlog maintenance* o *story time*, ja que no té nom oficial), l'equip es reuneix per a realitzar les tasques següent:

- Redefinir cada PBI.
- Constatar que l'equip de desenvolupament entén tots i cadascun dels requisits descrits.
- Assumir en què consisteixen els criteris d'acceptació.

- Reordenar el *backlog* del producte si es considera que les prioritats en realitat són unes altres.
- Eliminar els PBI que ja no són necessaris si és el cas.

Diffus i el seu backlog

La Fiona, la *product owner* del nostre projecte, ha fet un gran esforç durant les últimes setmanes. Ha contactat amb diferents companys, amb responsables de diverses àrees i, com no, amb tots i cadascun dels *stakeholders* implicats i interessats en l'èxit del projecte. Ha identificat les necessitats del negoci i ha redactat una a una les *stories* que representen aquestes necessitats. A continuació, es pot veure una de les *stories* que requereix la visualització d'un gràfic representant el cost de les matèries primeres:

Figura 6. Gràfic de matèries primeres

Gràfic de matèries primeres

Qui?
L'usuari amb perfil de mànager d'àrea.

Què?
Vol veure un gràfic que contingui la quantitat de matèries primeres adquirides.

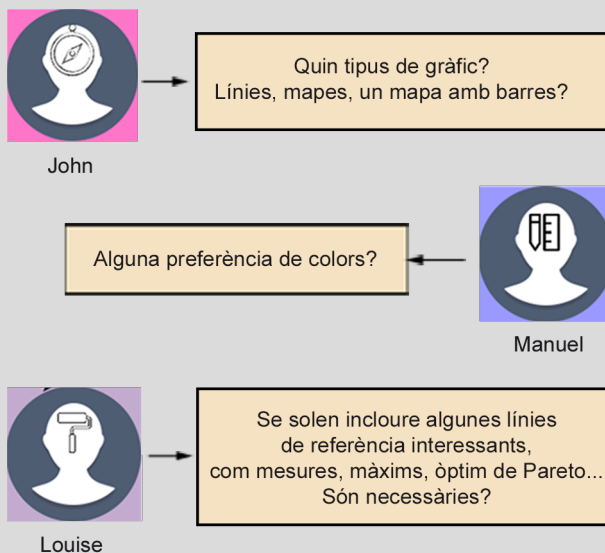
Amb quin objectiu?
Per poder valorar quines matèries són les més sol·licitades.

Criteris d'acceptació:

- Ordenació de més a menys quantitat
- Filtres geolocalitzadors habilitats
- Tipus de moneda seleccionable

Després de llegir aquesta descripció, l'equip de desenvolupament comença a plantejar-se dubtes, ja que la definició planteja clarament l'objectiu de l'usuari, però no té detalls que poden determinar l'èxit o el fracàs de la informació que es mostra al gràfic.

Figura 7. Exemple de preguntes de desenvolupadors cap al *product owner*



Després de diverses preguntes realitzades per l'equip de desenvolupament, respostes del *product owner*, recomanacions, etc., la Fiona actualitza la descripció de la mateixa *story* amb alguns detalls que seran decisius per a donar-li el valor esperat.

Figura 8. Gràfic de matèries primeres

Gràfic de matèries primeres

Qui?
L'usuari amb perfil de mànager d'àrea.

Què?
Vol veure un gràfic que contingui la quantitat de matèries primeres adquirides.

Amb quin objectiu?
Per poder valorar quines matèries són les més sol·licitades.

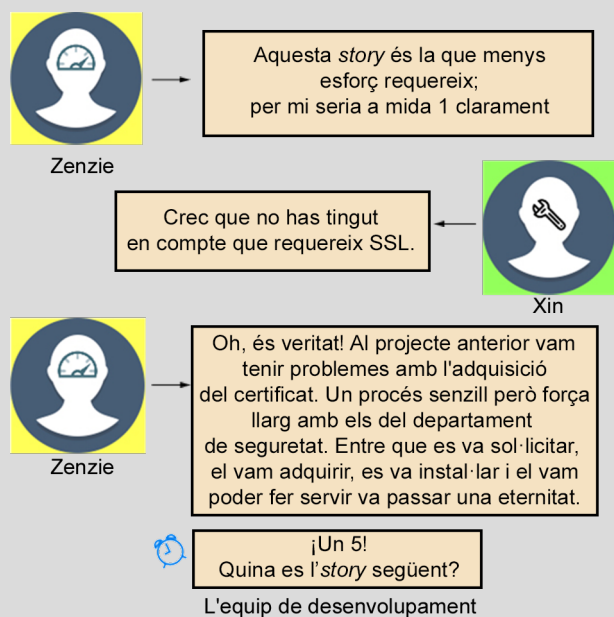
Criteris d'acceptació:

- Ordenació de més a menys quantitat
- Ordenació per matèria
- Línies de referència: cost mínim, cost màxim, cost mitjà
- Línia de Pareto que indiqui la relació 80/20
- Disponible filtre per a unitats/litres/quilos
- Filtres geolocalitzats habilitats
- Tipus de moneda seleccionable
- Selecció per any/trimestre/mes/setmana/dia
- Color gradient de vermell a blau (vermell: cost màxim; blau: cost mínim), línies de referència en gris, a excepció de la del cost mitjà, en vermell
- Gràfic de barres
- Indicació del cost de la matèria i el percentatge de la despesa total en cada barra

Quan l'equip sencer té clara aquesta *story*, passen a examinar la següent del *backlog* del producte fins a finalitzar l'anàlisi de totes i cadascuna d'elles.

Un cop acabada l'anàlisi total del *backlog* del producte, l'equip es troba en una posició adequada en la qual poder assignar punts, relativitzant així l'esforç requerit entre totes elles.

Figura 9. Exemple d'assignació de punts per a una *story*



4.2. Reunió de planificació d'*sprint*

A l'inici de cada *sprint* l'equip es reuneix durant l'*sprint planning*, o reunió de planificació d'*sprint*, de durada limitada com a màxim a vuit hores per a *sprints* de quatre setmanes. L'objectiu de la cerimònia consisteix a decidir quins ítems del *backlog* poden ser inclosos dins l'*sprint*. Ja que el *product owner* decideix quines prioritats tenen cada un dels ítems, l'equip té autoritat per a escollir quants seran inclosos a l'*sprint* per a ser desenvolupats i incrementar el producte.

En finalitzar la reunió de planificació d'*sprint*, es té com a resultat el **backlog de l'*sprint***, de l'anglès *sprint backlog*.

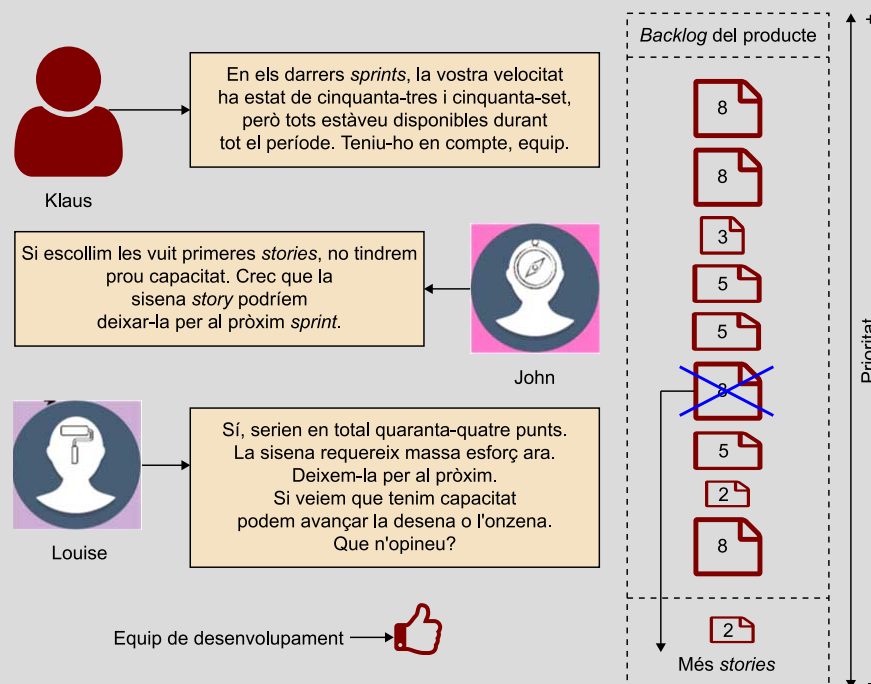
Diffus i la seva planificació d'*sprint*

En Klaus té molta experiència com a Scrum *master* i sap que, per a realitzar qualsevol planificació, és indispensable saber de quins recursos es disposen. Després de preguntar a cada un dels desenvolupadors, obté la informació següent:

- La Zenie i en Manuel estaran de formació durant un dia la primera setmana de l'*sprint*.
- En John té planificat agafar-se dos dies de vacances al final de l'*sprint*.
- La resta de l'equip, desenvolupadors, *product owner* i Scrum *master* inclosos, estan disponibles la resta de dies.

Després de posar en comú aquesta informació, l'equip decideix quantes *stories* són capaços de lliurar tenint en compte les vacants.

Figura 10. Reunió de planificació d'*sprint* a Diffus



4.3. Daily Scrum

El *daily Scrum*, o comunament conegut com a *daily standup*, és una reunió de durada limitada a quinze minuts on es tracten els ítems següents: què es va fer el dia anterior, quin pla hi ha per al dia actual i quins impediments hi ha. L'objectiu és compartir amb la resta de l'equip quines tasques estan en procés i resoldre els possibles problemes que n'impedeixin el desenvolupament. És comú que en aquestes reunions el *product owner* hi estigui present també, ja que a vegades ajuda a resoldre dubtes, dona informació que pot incorporar-se, o simplement veu com a poc a poc pren forma el producte. Aquestes reunions milloren tant la comunicació interna entre els desenvolupadors com la comunicació amb el client, que veu clarament com els seus PBI més prioritaris es van materialitzant dia rere dia.

Un standup a Diffus

En Klaus, l'*Scrum master*, ha planificat *standups* diaris durant tot l'*sprint*, que comencen a les 10.00 h cada matí. Tant la Fiona, la *product owner*, com els desenvolupadors hi assisteixen sens falta, inclosa la Louise, que es connecta remotament amb la resta de l'equip des de la seu on treballa habitualment, a Bèlgica. Mantenen tant el contacte visual com l'auditiu, ja que l'equip va valorar a l'inici del projecte que seria positiu veure's diàriament.

A l'*standup* parlen tots i cada un dels membres, tot i que el temps és limitat a quinze minuts, així que procuren ser puntuals a l'inici i ser concisos en la descripció de les tasques en què estan treballant, impediments, etc.

Es reuneixen en una àrea de la sala on solen treballar, dempeus, i amb un equip de videoconferència que enfoca tot l'equip.

Klaus: bon dia, equip! Louise, ens sents i ens veus correctament? Nosaltres a tu sí.

Louise: sí, tot ok.

Klaus: fantàstic, comencem... Avui és l'últim dia de mes, així que començarem per l'última lletra de l'abecedari... la zeta! Zenzie, avui et toca començar la primera al *standup*. Quan vulguis...

Zenzie: bon dia. Ahir vaig acabar amb els tests unitaris dels panells de navegació, vaig pujar el codi a la branca creada per a aquesta *story*, però vaig tenir problemes per a barrejar el codi. La Louise em va ajudar i finalment ho vam resoldre; gràcies de nou, Louise. Avui he començat amb la meva *story* següent, els filtres de geolocalització. De moment no tinc impediments.

Klaus: següent?

Manuel: continuo jo, per exemple. He estat treballant als gràfics de l'*story* DIFVIS-7502. Fiona, a l'*story* hi ha definit uns gradients de colors, de blau a vermell, i de groc a vermell. Em preguntava si necessiten seguir algun codi RGB concret.

Fiona: no, l'únic color estàndard que necessitem és el de la capçalera, juntament amb el logotip.

...

A vegades sorgeixen dubtes o discussions que requereixen més temps, però en aquest cas en Klaus (*Scrum master*) controla el temps i proposa als membres de l'equip afectats que es reuneixin *a posteriori* per resoldre'l o investigar-ho en privat.

4.4. Revisió de l'*sprint*

Després de la finalització de cada *sprint* hi ha una reunió de revisió en la qual participa l'equip sencer juntament amb la part interessada del negoci: els desenvolupadors, l'*Scrum master*, el *product owner* i els *stakeholders*. La durada és limitada com a màxim a quatre hores per *sprints* de quatre setmanes. L'objectiu de la trobada és mostrar l'evolució del producte als interessats, mostrar clarament com avança el projecte i quines novetats han estat implementades amb èxit i complint amb la DoD (*definition of done*). En molts casos, es comparteix amb els *stakeholders* el producte que ha estat desenvolupat fins al dia de la finalització de l'*sprint*, amb l'objectiu que aquests el facin servir i puguin compartir la seva opinió per així incorporar-la al *product backlog*.

La primera demo per a Diffus

Després de finalitzar el primer *sprint*, l'equip de desenvolupament fa una demostració de l'aplicació en la qual han estat treballant durant dues setmanes. Com que és el primer *sprint*, moltes de les tasques que s'han desenvolupat no són visuals, són més aviat extracció de dades de diferents fonts, transformació de dades i algunes tasques de disseny de l'aplicació. Després d'explicar breument la part tècnica completada, han realitzat una petita demo que mostra:

- El panell genèric amb les diferents mètriques, de moment deshabilitades, ja que no es manipulen les dades encara.
- La plantilla amb la capçalera, la barra de navegació i el panell de filtres.
- Com s'ha d'utilitzar l'aplicació a grans trets, com funcionen els filtres i com es consulta la data d'actualització de l'última càrrega de dades.

Els desenvolupadors han facilitat accés als *stakeholders* a l'aplicació que han ensenyat perquè comencin a familiaritzar-s'hi des de l'inici.

4.5. Retrospectiva de l'*sprint*

Després de finalitzar cada *sprint* l'equip torna a reunir-se, aquest cop sense *stakeholders*, per analitzar com ha anat el transcurs de l'*sprint* i quines millores poden afegir-se al pròxim.

La taula següent representa la durada màxima d'aquest tipus de reunions dependent de la longitud de l'*sprint*:

Taula 2. Durada màxima de la retrospectiva de l'*sprint*

Durada de l' <i>sprint</i>	Durada de la retrospectiva de l' <i>sprint</i>
Quatre setmanes	Fins a 3 h
Tres setmanes	Fins a 2 + 1/4 h
Dues setmanes	Fins a 1 + 1/2 h
Una setmana	Fins a 3/4 h

Hi ha moltes tècniques per a realitzar i guiar la retrospectiva de manera dinàmica i productiva de cara a les pròximes iteracions. En general, totes elles tenen en comú donar resposta a algunes preguntes genèriques, on tots i cada un dels membres de l'equip tenen veu i expressen la seva opinió. Normalment consisteix que, després d'uns minuts per a la reflexió, cada membre de l'equip contesta les qüestions següents segons la seva pròpia percepció de la iteració finalitzada:

- Què ha funcionat correctament?
- Què podria haver-se fet millor?
- Què es pot millorar en el futur?
- Què hem après?
- Què no ha aportat valor afegit?

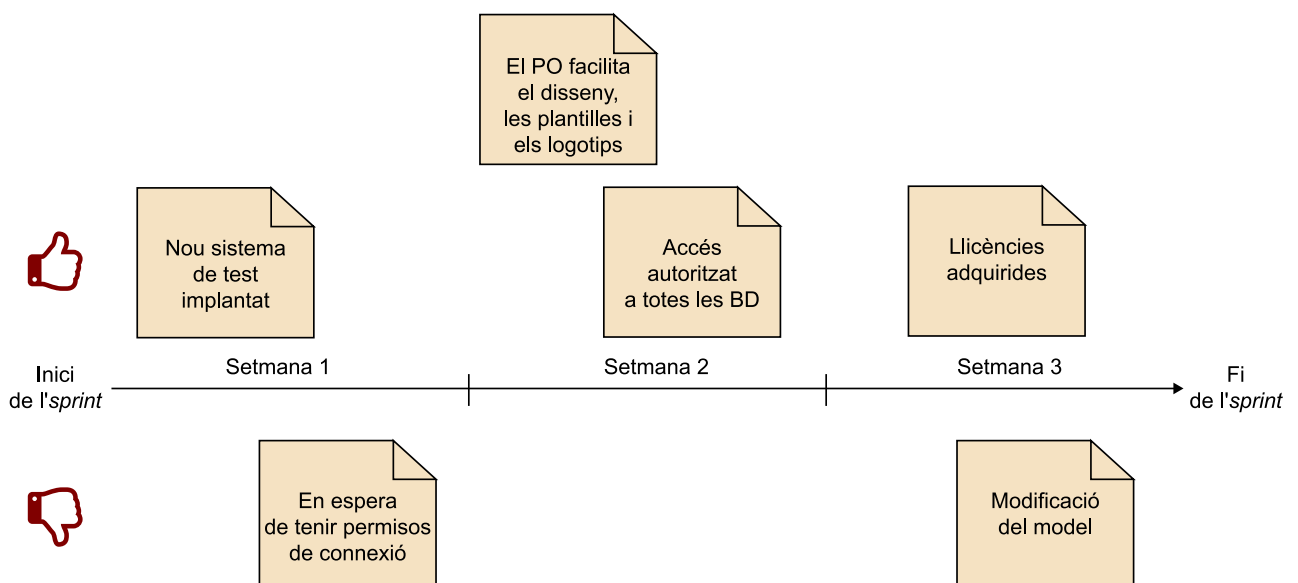
Després de la discussió oberta entre tot l'equip amb cadascuna de les opinions, s'anota si cal prendre alguna acció per a l'*sprint* següent.

La retrospectiva no ha de ser avorrida per a l'equip: és una reunió interactiva on els membres han de sentir-se còmodes per a expressar opinions i crítiques constructives. A continuació, es mostren algunes tècniques per a realitzar la retrospectiva de manera més activa i dinàmica.

1) Línia del temps

La tècnica de la línia en el temps (*timeline*) consisteix a col·locar indicadors d'esdeveniments que hagin ocorregut durant l'*sprint*. Aquests indicadors, que poden ser notes de paper, es col·loquen al llarg d'una línia temporal que representa el transcurs de l'*sprint*. Els esdeveniments han de ser classificats com a positius o negatius.

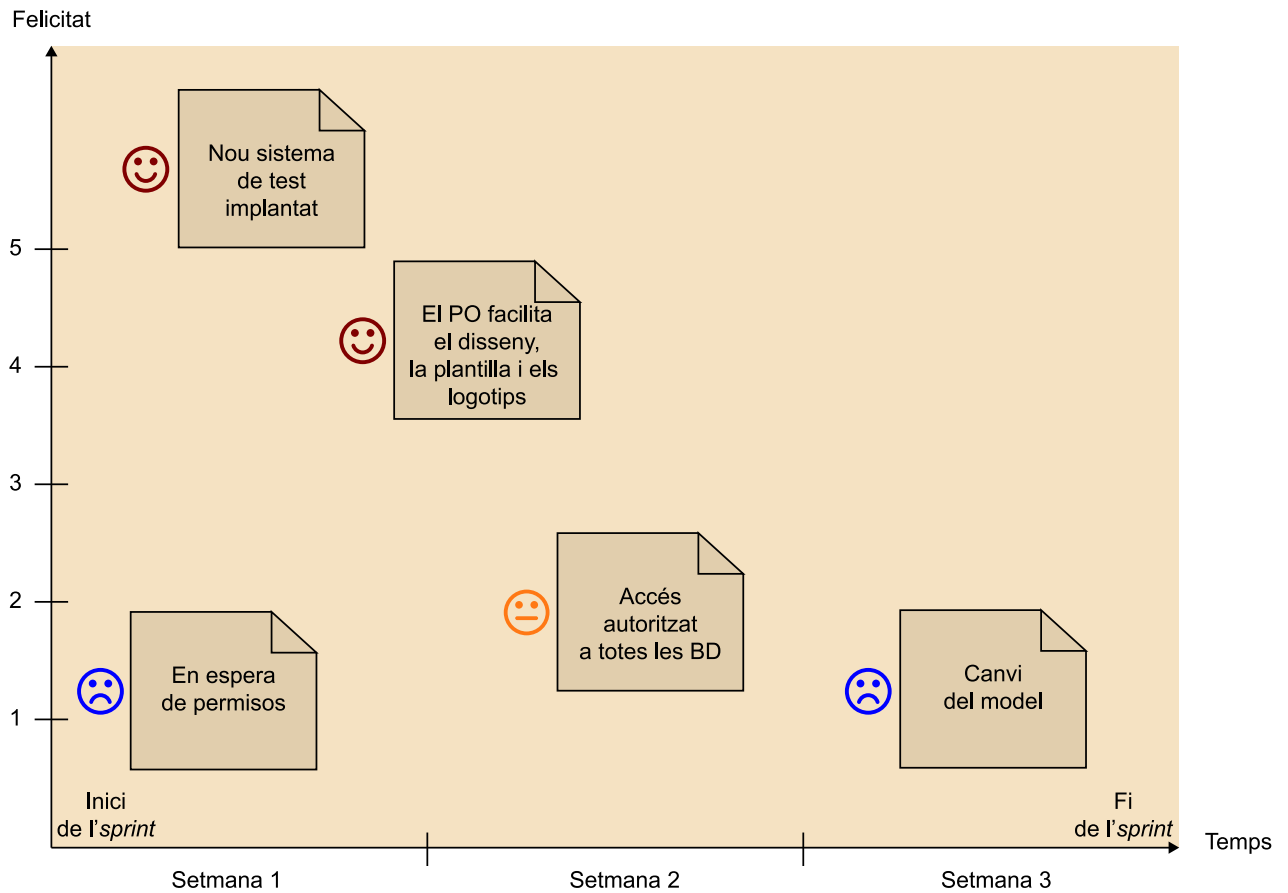
Figura 11. Exemple de *timeline*



2) Happiness metric

La mètrica de la felicitat (*happiness metric*) també és una tècnica molt usada en les retrospectives d'*sprint*. La tècnica consisteix a mostrar l'estat emocional de cada membre de l'equip; s'atorguen valors de l'1 al 5 al costat d'una breu explicació de l'avaluació. El gràfic resultant és guardat per realitzar posteriors comparacions.

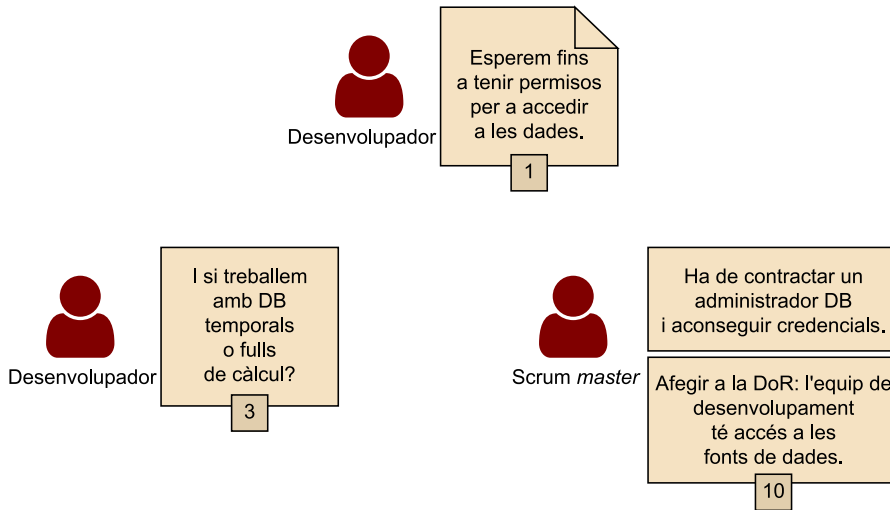
Figura 12. Exemple de *happiness metric*



3) El joc de la perfecció

Aquesta tècnica consisteix a puntuar idees entre l'1 i el 10. L'objectiu és maximitzar la crítica constructiva i saber com millorar qualsevol plantejament fins a ser «immillorable», donats els recursos i les circumstàncies.

Figura 13. Exemple de situació amb el joc de la perfecció



5. Artefactes Scrum

Amb l'objectiu de donar transparència i possibilitar l'adaptació i la millora, Scrum posseeix una sèrie d'artefactes que faciliten aquestes tasques.

5.1. El *backlog* del producte

El *backlog* del producte (*product backlog*) consisteix en una llista ordenada d'ítems (*product backlog items*, PBI o també *stories*) on es descriuen detalladament les funcionals i característiques del producte que són desitjades. La llista és visible per a tots els *stakeholders* i pot ser incrementada per qualsevol membre de l'equip, però només és el *product owner* qui la reordena contínuament decidint quins ítems són més prioritaris.

Figura 14. *Backlog* del producte a l'esquerra, del qual es negocien diversos PBI per realitzar a l'*sprint* i es desglossen en diverses tasques més concretes

<i>Backlog del producte</i>	<i>Backlog de l'sprint</i>
AD usuaris	AD usuaris
Connexió de dades	Connexió de dades
Modelatge de dades	Modelatge de dades
Navegació	
Filtres temporals	
Filtres de geolocalització	
Selector de moneda	
Selector del volum de matèries	
Gràfic de matèries primeres	
Taula amb detall matèries/localització/preus	
Gràfic de l'ús de les eines enfront del cost de les eines	
Gràfic ús/quantitat enfront del total - Incloure Pareto	
Gràfic d'eines enfront de treballadors	
Taula amb el detall de les eines	
KPI despesa per unitat de negoci	
Descàrrega visual de gràfics	
Descàrrega de dades gràfiques	
...	

5.2. Ítems del *backlog* del producte (PBI)

En un ítem del *backlog* s'especifica quines característiques són necessàries per al producte. Normalment sol ser redactat com una història d'usuari, de l'anglès *user story* o simplement *story*, i pot tenir especificats diversos criteris d'acceptació. L'esforç requerit s'estima en unitats relatives per l'equip i no sol excedir de dos o tres dies de treball per a dues o tres persones. En cada PBI s'especifica què cal, però no es detalla com ha de ser realitzat.

Figura 15. Exemple de PBI

Gràfic d'eines: ús enfront de cost

Qui?
L'usuari amb perfil de mànager d'àrea

Què?
Vol veure un gràfic que contingui l'ús de les eines enfront del cost d'aquestes eines.

Criteris d'acceptació:

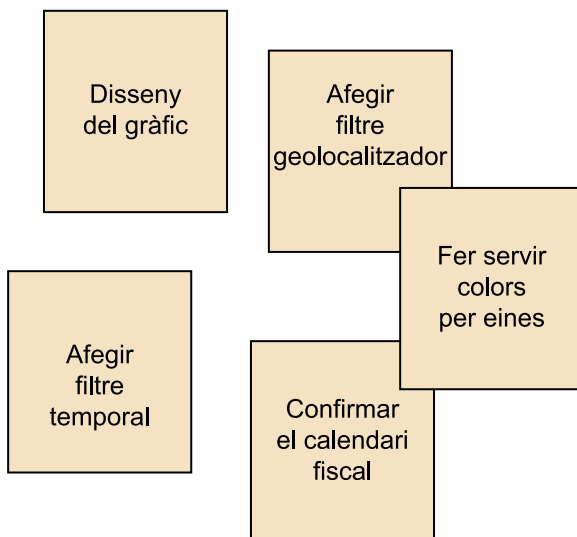
- Filtre per geolocalització actiu per al gràfic
- Filtre temporal (any/trimestre/mes/setmana/dia) actiu per al gràfic
- ...

Esforç: **L**

5.3. Tasques de l'*sprint*

Els PBI especifiquen quines funcionalitats i característiques s'inclouran en l'increment, però el com es realitza està especificat a les tasques de l'*sprint*. Cada PBI pot contenir una o diverses tasques en què es defineix detalladament el temps necessari per a desenvolupar la feina, quin esforç queda encara pendent per ser completada la tasca, i una persona responsable, entre altres detalls que poden ser a criteri de l'equip o de l'empresa.

Figura 16. Exemple de tasques possibles per al PBI «Gràfic d'eines: ús enfront de cost»



Taula 3. Exemple de tasques possibles per al PBI «Gràfic d'eines: ús enfront de cost»

Story	Tasca	Mida	Creador de la tasca	Responsable	Estat	Hores restants
Gràfic d'eines: ús enfront de cost	Disseny del gràfic	S	John	Manuel	Completada	0 h

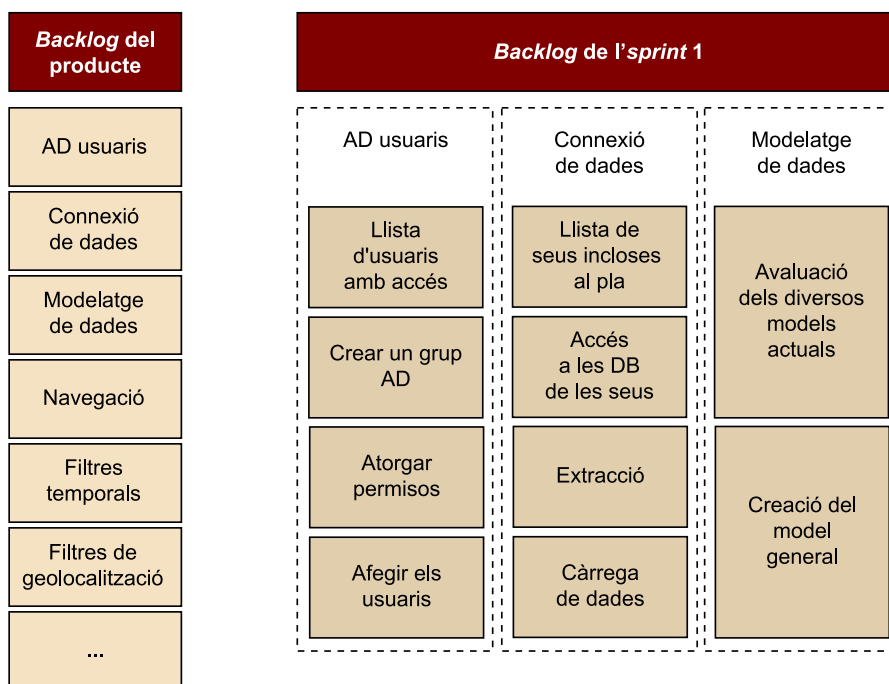
Story	Tasca	Mida	Creador de la tasca	Responsible	Estat	Hores restants
Gràfic d'eines: ús enfront de cost	Afegir filtre temporal	S	John	Zenzie	Sense començar	3 h
Gràfic d'eines: ús enfront de cost	Afegir filtre localitzador	M	John	John	Sense començar	10 h
Gràfic d'eines: ús enfront de cost	Confirmar el calendari fiscal	S	Zenzie	Zenzie	Bloquejada	2 h
Gràfic d'eines: ús enfront de cost	Utilitza colors per eines	S	John	Manuel	Completada	0 h

5.4. Backlog de l'sprint

El *backlog* de l'*sprint* consisteix en una col·lecció de PBI que han de ser implementats al llarg de la iteració fins a arribar a la DoD. Cada PBI és inclòs a l'*sprint* a criteri de l'equip de desenvolupament després d'haver estat negociats amb el *product owner* a la reunió de planificació de l'*sprint*.

L'equip identifica per a cada PBI quines tasques són necessàries per a realitzar-lo.

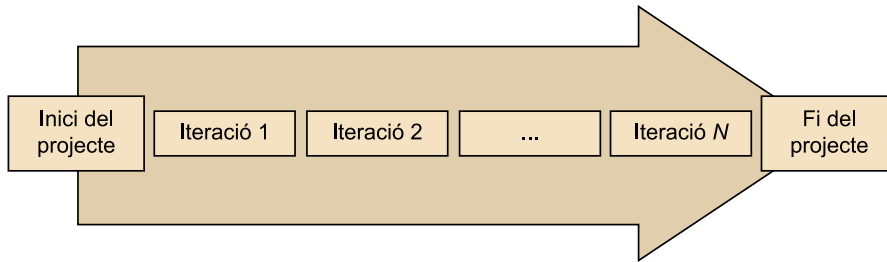
Figura 17. Backlog del producte a l'esquerra, PBI inclosos a l'sprint i desglossats



5.5. Increment

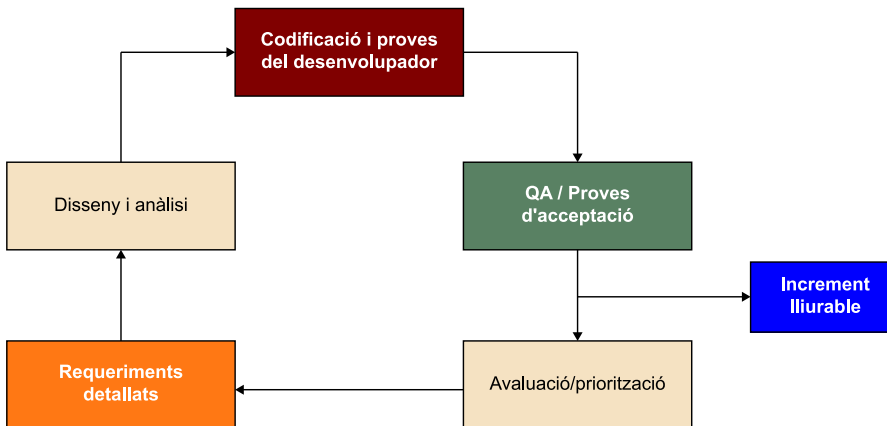
Se l'anomena increment a la totalitat de PBI que són completats al final de cada *sprint*. L'increment ha de ser potencialment lliurable i només els membres de l'equip de desenvolupament poden crear-lo.

Figura 18. Desenvolupament d'un projecte per iteracions



Per poder completar amb èxit cada increment es realitzen tasques similars repetidament:

Figura 19. Fases dins d'una iteració per a completar l'increment



5.6. *Sprint burn down chart*

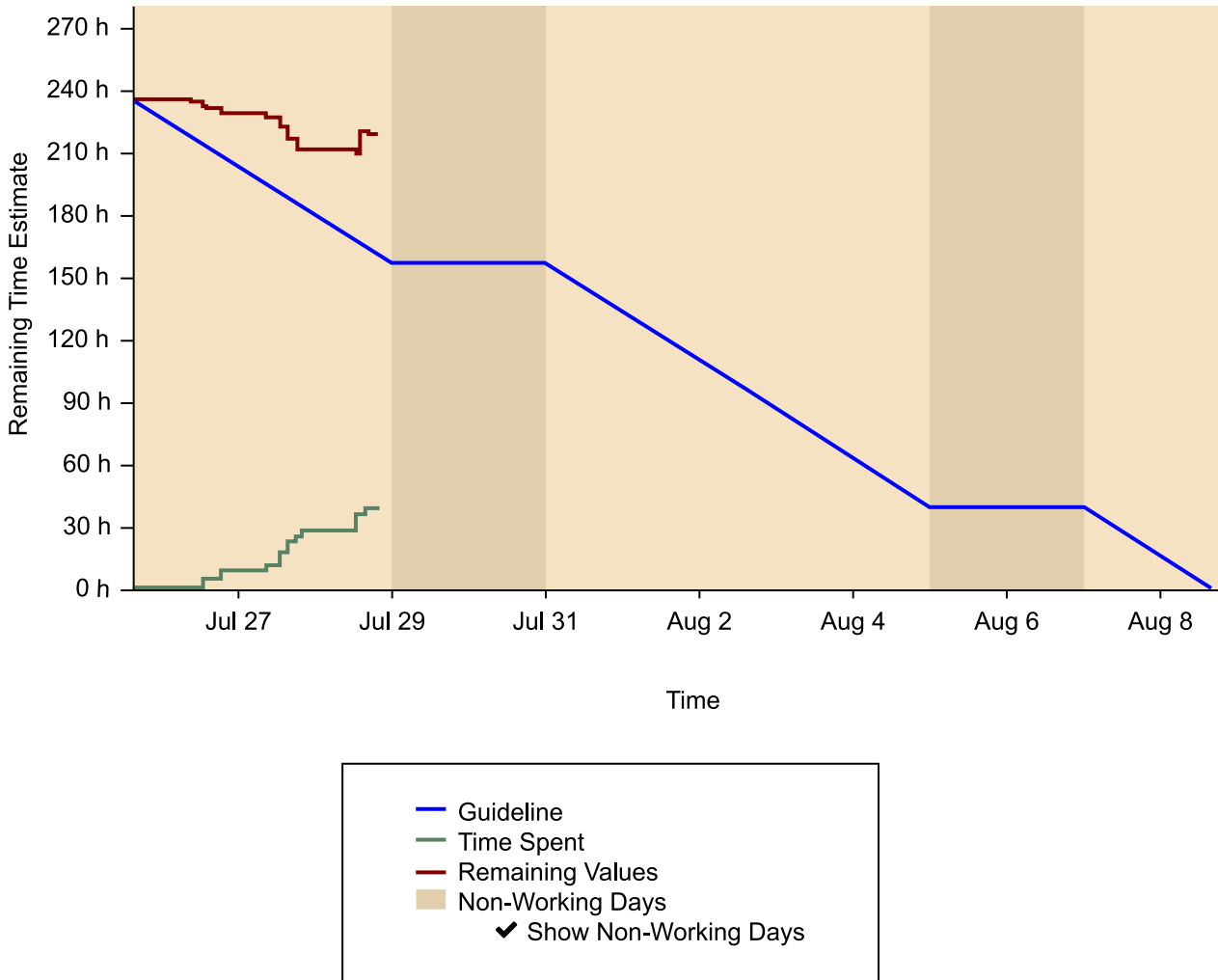
L'*sprint burn down chart* és un gràfic que facilita informació sobre l'execució de cada *sprint*, mostrant el nombre d'hores restant dins de cada iteració. Gràcies a això es pot saber quina tendència s'està seguint, és a dir, si s'està endarrerit respecte a la planificació o si, per contra, l'equip està avançat. Aquesta és una eina interna de l'equip que facilita l'autogestió i en cap cas s'ha d'utilitzar com a informe de gestió.

Els gràfics *burndown* poden donar molta informació valuosa sobre com està desenvolupant la feina l'equip de desenvolupament:

- tendència real enfront de tendència ideal
- hores de desenvolupament superiors a les previstes
- tasques finalitzades abans d'hora
- tasques massa temps en procés
- tasques innecessàries
- més tasques necessàries segons el volum de feina
- etc.

Quan el projecte ha finalitzat també es realitzen gràfiques *burndown* que preveuen tots els *sprints* executats. En aquest cas, es donarà informació més genèrica sobre l'evolució de l'equip al llarg de tot el procés.

Figura 20. Exemple de gràfica *burndown* de Diffus per al seu primer *sprint*



6. Més enllà d'àgil

6.1. Squads - Cas pràctic: Spotify

Spotify, la coneguda empresa de reproducció musical nascuda el 2008, també segueix una cultura àgil usant Scrum. Després de la seva gran expansió, els equips de desenvolupament de programari es van anar incrementant i hi va haver un moment en què era complicat mantenir uns estàndards comuns entre tants conjunts d'ED, SM i PO. Així que van decidir fer part d'Scrum opcional, ja que no sempre van trobar necessari seguir el marc de treball al peu de la lletra.

Spotify

El web oficial d'Spotify és <https://www.spotify.com>.

Principis àgils per sobre de pràctiques d'Scrum.

A la companyia es van produir diversos canvis. En primer lloc, van canviar la nomenclatura:

- Scrum *master* > *agile coach*
- Equip Scrum > *squad*

Un *squad* consisteix en un equip de menys de vuit persones, habitualment, multifuncionals i autònoms. Tot i que cada *squad* té els seus propis objectius, entre tots tenen objectius comuns relacionats amb l'estratègia de la companyia, el producte que desenvolupa i pel que fa a la resta d'*squads*.

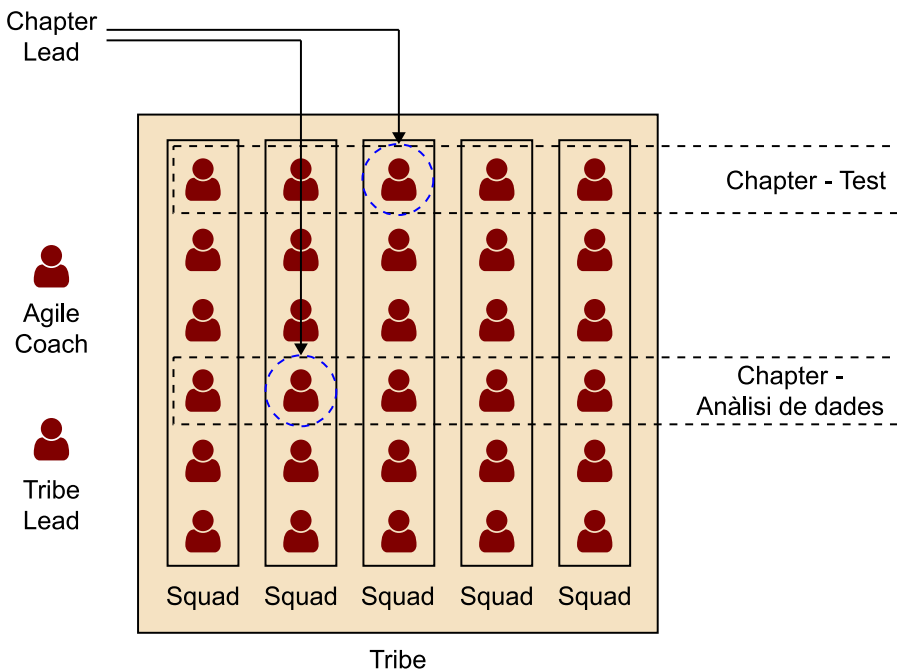
Taula 4. Autonomia enfront d'alineació en equips de desenvolupament

		Autonomia	
		Baixa	Alta
Alineació	Baixa	<ul style="list-style-type: none"> • Poca direcció o direcció nul·la. • Els equips fan independentment i internament dins el grup el que consideren. 	<ul style="list-style-type: none"> • La direcció no és clara. • L'equip s'autogestiona però no té un objectiu comú.
	Alta	<ul style="list-style-type: none"> • La direcció expressa què necessita. • Els equips segueixen les normes estrictament, no hi ha llibertat per decidir què i com fer-ho. 	<ul style="list-style-type: none"> • La direcció expressa què necessita. • Els equips s'autogestionen en què fer i com fer-ho.

L'ideal és estar en una **alta alineació** de l'empresa on la direcció comunica clarament quina és la necessitat o el problema que cal resoldre i una **alta autonomia** dels equips en què aquests estudien, decideixen i col·laboren en com i qui donarà lloc a la millor solució.

A continuació, es mostra un gràfic de com s'estructuren els *squads* i altres figures importants dins d'aquest nou paradigma:

Figura 21. Exemple de *tribe* i *squads*



a) **Esquadró**, de l'anglès *squad*: compost per diferents perfils com poden ser desenvolupadors, administradors de base de dades, *testers*, etc. Cada *squad* se centra en el lliurament de productes i en la seva qualitat.

b) **Capítol**, de l'anglès *chapter*: grup de gent que treballa en la mateixa àrea de competència. L'objectiu d'un *chapter* és intercanviar idees, innovar tecnologies i ajudar-se mútuament amb els reptes que se'ls presenten dia a dia.

c) **Líder del capítol**, de l'anglès *chapter lead*: és l'últim responsable del *chapter* al qual pertany, té la responsabilitat dins el *chapter* de desenvolupament, *coaching* i cicles de treball.

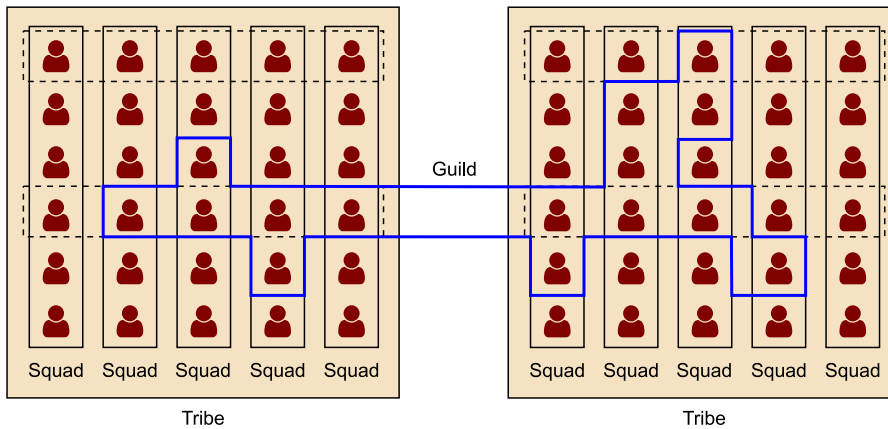
d) **Tribu**, de l'anglès *tribe*: grup d'*squads* amb missions interconnectades, necessari per a millorar la coordinació dins dels diferents equips. Normalment es compon de menys de cent cinquanta persones.

i) **Líder de tribu**, de l'anglès *tribe lead*: figura que assegura la intercomunicació entre tots els membres, garanteix que es comparteixi el coneixement i estableix prioritats, entre altres gestions. És també la interfície amb altres tribus.

f) **Agile coach**: persona encarregada d'assessorar altres membres o directament *squads*, facilitant el seu progrés.

També hi ha **gremis**, de l'anglès *guild*: és un grup de persones de diferents *squads* que tenen els mateixos interessos i comparteixen coneixement sobre una àrea en concret. No tots tenen per què pertànyer al mateix *chapter*; poden ser, per exemple, part de l'automatització, el desenvolupament web, etc.

Figura 22. Exemple de *guild*



Dins d'aquest nou marc presentat, hi ha una forta doctrina de lliurament continu. És ben sabut que passar de desenvolupament a producció és una tasca difícil que comporta alta responsabilitat; és per això que normalment es passa a producció poques vegades. En aquest cas, s'evita tenir grans lliuraments i es facilita el procés de lliurament perquè sigui molt freqüent, de manera que passar a producció es converteix en una tasca rutinària més. Això comporta prestar especial atenció a l'automatització de test i a la infraestructura per al lliurament continu.

6.2. Més pràctiques àgils: Kanban, eXtreme Programming, DevOps

Scrum és un dels marcs de treball més utilitzats per a implementar metodologies àgils, però n'hi ha molts més que també són comuns a dia d'avui. A continuació, es descriuen algunes de les metodologies més importants.















1) kanban, Kanban o sistema kanban

La paraula *kanban* deu el seu origen a dues paraules japoneses que en conjunt signifiquen alguna cosa així com 'targeta visual' o 'targeta de visualització' (*kan* significa 'visual' i *ban* significa 'targeta').

El concepte kanban va ser creat en els sistemes de producció de Toyota (TPS, de l'anglès *Toyota production system*). Es basa en un sistema de planificació que controla l'avanç de la producció dins de la cadena de producció. Deu els seus orígens a Lean i està basat en els principis següents:

a) **Visualització:** el sistema kanban fa servir taulers kanban per a visualitzar el treball i el progrés.

Figura 23. Exemple de tauler Kanban

Per fer	En procés	En espera	Finalitzat
 			
 	 		
 			
			

b) **Límit de treball en curs:** es fomenta que es faci poca feina a la vegada, sense confondre això amb fer poca feina. Limitar el treball en curs ajuda a completar ràpidament més feina en total.

c) **Control de flux:** eliminar residus, tasques que no donen valor.

d) **Millora col·laborativa.**

e) **Rols principals:** *service request manager*, entén les necessitats del negoci i prioritza les necessitats, i *service delivery manager*, responsable del flux de treball, facilitador de cerimònies i planificacions.

2) eXtreme Programming (XP)

Creada per Kent Beck, Ward Cunningham i Ron Jeffries a finals dels noranta. Alguns la defineixen com un moviment «social» dels analistes del programari cap als homes i les dones de negocis, del que hauria de ser el desenvolupament de solucions en contraposició dels legalismes dels contractes de desenvolupament.

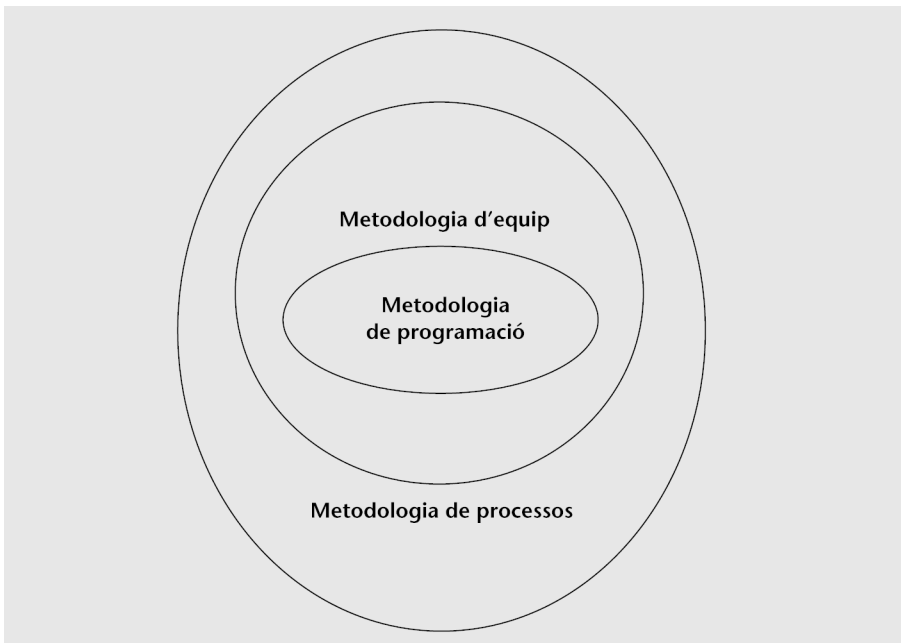
Per assolir l'objectiu de programari com a solució àgil, la metodologia XP s'estructura en tres capes que agrupen les dotze pràctiques bàsiques d'XP:

a) **Metodologia de programació:** disseny senzill, test, refactorització i codificació amb estàndards.

b) **Metodologia d'equip:** propietat col·lectiva del codi, programació en parelles, integració contínua, només quaranta hores setmanals de treball i metàfora del negoci.

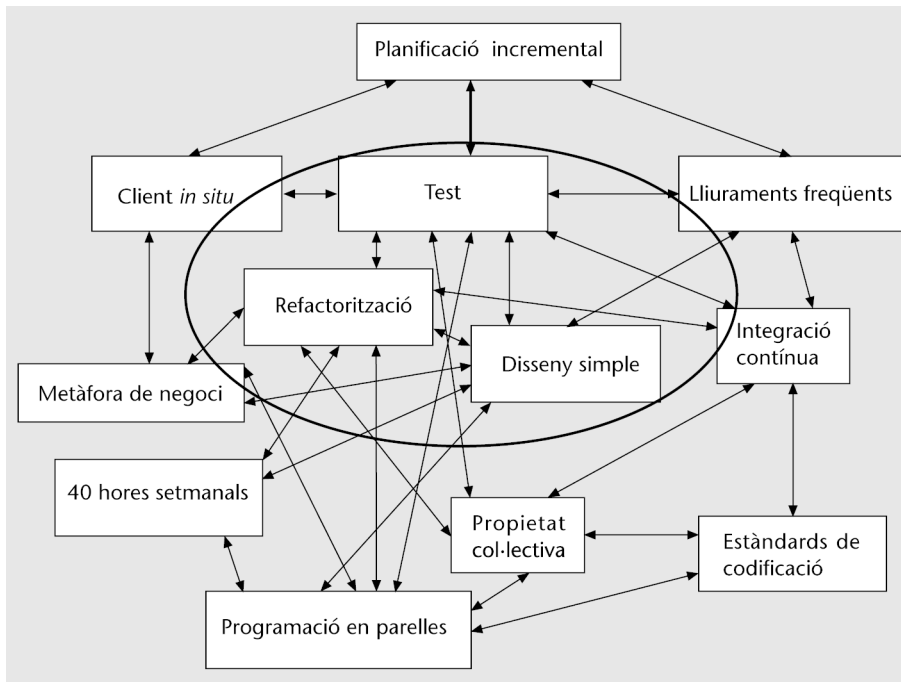
c) **Metodologia de processos:** client *in situ*, lliuraments freqüents i joc de la planificació.

Figura 24. Estructura en tres capes de la metodologia XP



Kent Beck, Ward Cunningham i Ron Jeffries tenien molt clar les pràctiques que els havien donat millors resultats en els seus projectes. Així que van intentar aplicar-les totes juntes, amb la qual cosa es va fer un pas més enllà. Aquest va ser l'embrió de la metodologia XP. Van crear les dotze pràctiques que es reforcen entre elles per obtenir els millors resultats. Les relacions entre elles les podem veure al gràfic següent, que hem adaptat del que va proposar Kent Beck.

Figura 25. Relacions entre les dotze pràctiques de la metodologia XP



Al centre se situen les pràctiques que més resultats poden donar en adaptar-les; no són altres que el disseny simple, el test i la refactorització. Fins i tot si no es vol prendre la totalitat de les pràctiques d'XP adoptant aquestes tres a la nostra metodologia habitual, es pot obtenir una millora substancial en els resultats.

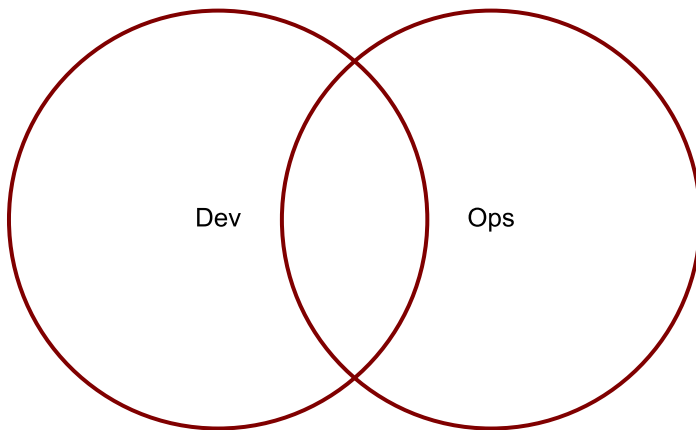
Cicle de vida d'un projecte XP:

- fase d'exploració,
- fase de planificació,
- fase d'iteracions,
- fase de producció,
- fase de manteniment,
- fase de mort del projecte.

3) DevOps

DevOps, de l'anglès *development and operations*, és considerada una extensió de les metodologies àgils. Se centra en el desenvolupament de productes mantenint una comunicació i una col·laboració intensa entre desenvolupadors de programari i operadors de serveis.

Figura 26. Diagrama mostrant DevOps com la intersecció de desenvolupament i operacions



A continuació, es mostren algunes de les característiques més importants que defineixen aquesta metodologia:

- a) **Compartir coneixement, responsabilitat i objectiu comú:** tots dos rols mantenen l'objectiu d'unir forces per millorar el rendiment i la qualitat del producte reduint el temps de desenvolupament. La col·laboració és la clau.
- b) **Automatització, lliurament i desenvolupament, integració contínua.**
- c) **Mètriques:** per millorar contínuament s'agafen mètriques contínuament amb l'objectiu de comparar el procés. Per exemple: clients guanyats/perduts cada mes?, incidències/*bugs* oberts aquesta setmana?
- d) **Acceptar el fracàs, perquè és inevitable:** els equips aprenen a detectar i a reparar els errors ràpidament.

Resum

En aquest mòdul s'ha introduït a l'estudiant en què consisteixen les metodologies àgils i quin és el manifest i els principis en què es basen totes elles:

- Més valoració dels individus i les seves interaccions que no pas dels processos i eines.
- Programari que funciona per sobre de documentació extensa.
- Col·laboració amb el client abans que negociació contractual.
- Resposta davant els canvis per sobre del seguiment d'un pla.

S'ha vist amb detall una de les metodologies àgils més esteses, l'Scrum.

Dins d'Scrum s'han descrit els principis INVEST i els valors que se segueixen: transparència, revisió i adaptació.

També s'han descrit els diferents rols Scrum i les seves responsabilitats: *Scrum master*, que és el facilitador entre el negoci i els desenvolupadors i vigila com s'implementa el marc de treball Scrum; *product owner*, que representa el negoci i ajuda a aclarir què és necessari; i l'equip de desenvolupament, que implementa la solució en petites iteracions anomenades *sprints*.

A continuació, es pot veure un resum de totes les cerimònies que es realitzen al marc Scrum, quina durada tenen, qui són els participants, amb quina freqüència se celebren les cerimònies i quins resultats són obtinguts amb cadascuna d'elles:

Taula 5

Cerimònia	Durada	Assistents	Periodicitat	Inputs	Outputs
Refinament del <i>backlog</i>	Depenent de la durada de l' <i>sprint</i> , màx. 4 h	SM PO ED	A l'inici del projecte Inici d' <i>sprint</i> si cal	Ítems del <i>backlog</i> del producte	Ítems refinats del <i>backlog</i> del producte
Planificació de l' <i>sprint</i>	Depenent de la durada de l' <i>sprint</i> , màx. 8 h	SM PO ED	Inici de cada <i>sprint</i>	<i>Backlog</i> del producte	<i>Backlog</i> de l' <i>sprint</i> definit i limitat de la resta del <i>backlog</i> del producte
<i>Daily Scrum</i>	Màx. quinze minuts	SM PO (recomanat) ED	Diària	Impediments Tasques actuals	Solucions a impediments
Revisió de l' <i>sprint</i>	Depenent de la durada de l' <i>sprint</i> , màx. 4 h	SM PO ED <i>stakeholders</i>	Final de cada <i>sprint</i>	Demostració del producte desenvolupat	Retorn de l'acció

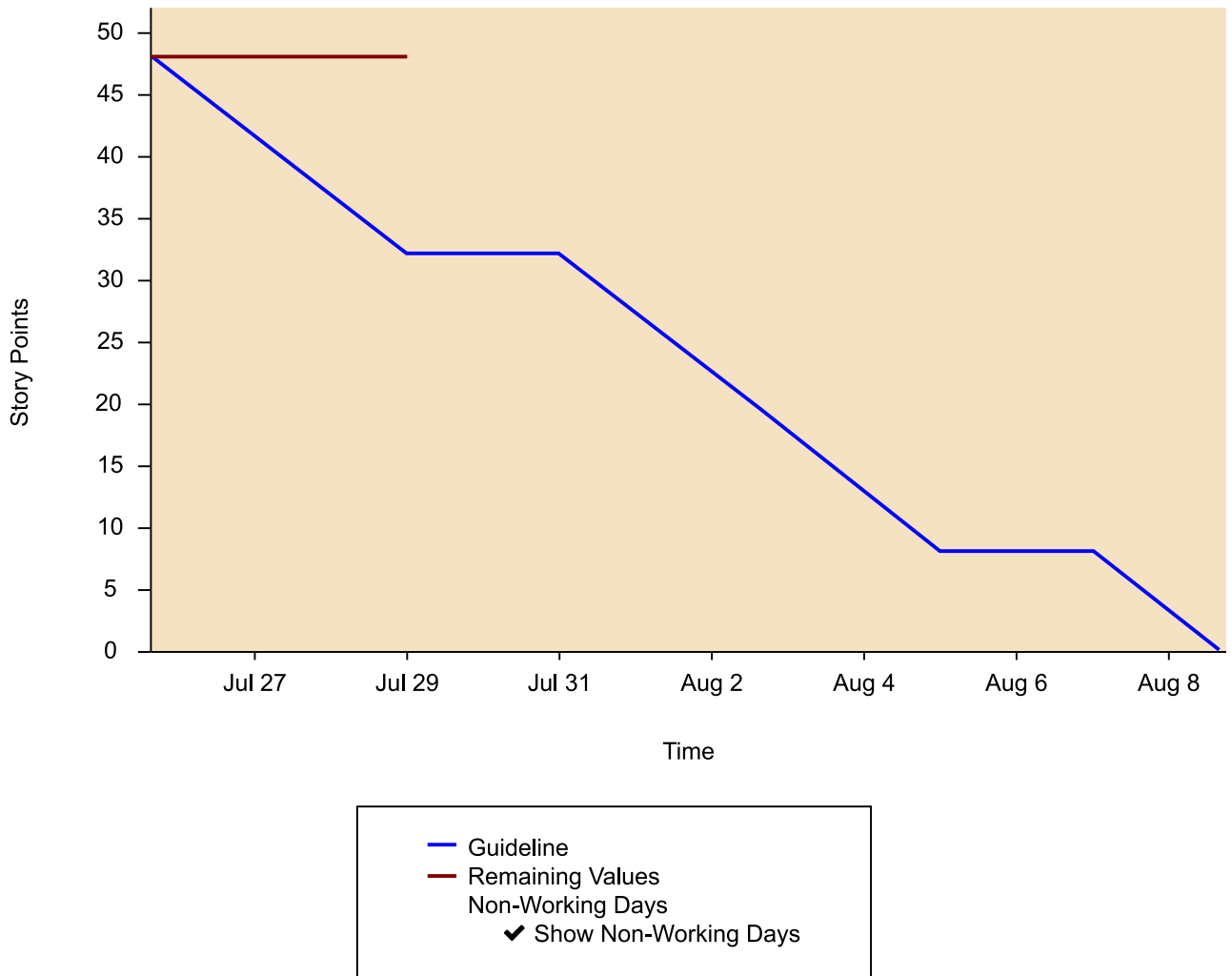
Cerimònia	Durada	Assistents	Periodicitat	Inputs	Outputs
Retrospectiva de l' <i>sprint</i>	Depenent de la durada de l' <i>sprint</i> , màx. 3 h	SM PO ED	Final de cada <i>sprint</i>		Identificades millores per a l' <i>sprint</i> següent

Per dur a terme Scrum s'han donat a conèixer els diferents artefactes necessaris per a facilitar la transparència en les tasques del dia a dia.

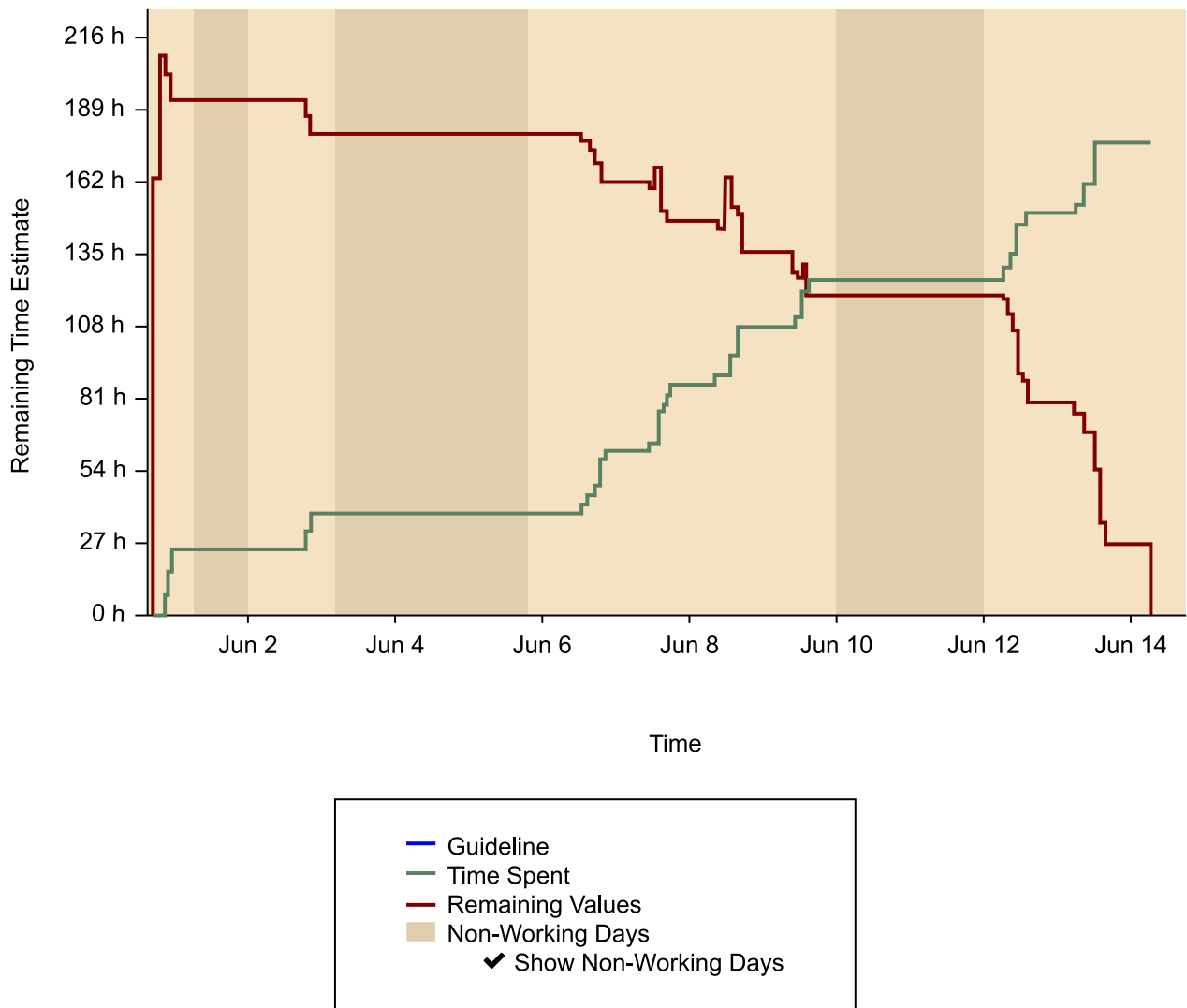
Finalment, s'han descrit altres metodologies àgils comunes a dia d'avui. Aquestes metodologies comparteixen alguns dels principis generals d'Scrum, però cadascuna defineix un procés i unes activitats diferenciades per dur a terme el desenvolupament.

Exercicis d'autoavaluació

1. Quines activitats solen pertànyer a la DoD en equips de desenvolupament de programari?
2. Imagineu que el *backlog* de l'*sprint* conté principalment *stories* de disseny. Podria absentar-se del refinament del *backlog* un membre de l'equip que únicament té experiència en l'àrea de test?
3. Què passa al gràfic *burndown* següent? Per què els *story points* no segueixen la línia blava?



4. Al gràfic següent es mostra un gràfic *burndown* on el temps estimat de les *stories* creix just al començament. Per què passa aquest efecte?



5. Quins són els rols d'un equip Scrum?

6. Per ordre d'execució, quines són les cerimònies portades a terme en Scrum?

7. En quines cerimònies tenen lloc les accions següents:

a) L'equip decideix afegir un nou ítem a la DoR.

b) El *product owner* clarifica dubtes sobre una *story*.

c) Tots estan dempeus.

d) Un desenvolupador explica què té previst fer durant el dia.

e) Els desenvolupadors negocien amb el *product owner* si una tasca pot ser escollida a l'*sprint* següent.

f) El *product owner* presenta la resta d'assistents.

8. De les dotze pràctiques d'XP, quins tenen relació amb una metodologia d'equip? Quines són les tres més importants?

Solucionari

Exercicis d'autoavaluació

1. Test unitaris passats, revisió de codi, criteris d'acceptació complets, test de regressió passats, test funcionals passats, el *product owner* accepta la *user story*.
2. No, no hauria d'absentar-se. L'equip de desenvolupament complet és responsable de l'increment realitzat i l'experiència de tots els membres és vital. En aquest cas, caldrà testejar la part visual i el seu coneixement de l'àrea pot ser decisiu per a identificar tant facilitats com dificultats.
3. L'*sprint* acaba de començar i encara no s'ha acabat cap *story* que tingui l'esforç estimat.
4. Es van afegir estimacions de temps un cop l'*sprint* ja havia començat.
5. *Product owner*, *Scrum master* i equip de desenvolupament. Els *stakeholders* no són considerats com a part de l'equip.
6. Refinament del *backlog*, planificació de l'*sprint*, *daily Scrum*, revisió de l'*sprint*, retrospectiva de l'*sprint*.
7. a) Retrospectiva de l'*sprint*.
b) Principalment durant el refinament del *backlog*. També es dona el cas d'aclarir detalls durant els *standups* diaris.
c) *Daily Scrum*.
d) *Daily Scrum*.
e) Planificació de l'*sprint*.
f) Revisió de l'*sprint*.
8. Les pràctiques que tenen relació amb la part metodològica de treball en equip són: propietat col·lectiva del codi, programació en parelles, integració contínua, quaranta hores setmanals i metàfora del negoci.

Les tres més importants de les dotze pràctiques són el disseny simple, el test i la refactorització. Fins i tot, si no volem adoptar la totalitat de les pràctiques d'XP, adoptant aquestes tres a la nostra metodologia habitual podem tenir una substancial millora en els resultats obtinguts.

Glossari

backlog *f* Tasques pendents que es poden dur a terme durant el projecte.

característica *f* Funcionalitat simple i poc costosa de desenvolupar que aporta valor al client del programari que s'ha d'utilitzar.

client *m* Membre de l'equip Scrum que representa als *stakeholders* (la visió de negoci) en el procés de desenvolupament. Entre altres funcions, s'encarrega de gestionar el *backlog* del producte (crear-lo, prioritzar les tasques, actualitzar-lo, etc.). *Product owner*, en anglès. sigla PO

definició d'acabat *f* Criteri usat per a decidir si un ítem del *backlog* ha estat completat de manera satisfactòria. sigla DoD

definició de preparat *f* Criteri usat per a decidir si és possible incloure un ítem del *backlog* a l'*sprint* següent. sigla DoR

definition of done *f* Vegeu definició d'acabat.

definition of ready *f* Vegeu definició de preparat.

DoD *f* Vegeu definició d'acabat.

DoR *f* Vegeu definició de preparat.

ED *m* Vegeu equip de desenvolupament.

equip de desenvolupament *m* Subconjunt de l'equip Scrum d'entre 3 i 8 persones dedicat a les activitats que permeten construir l'increment en cada iteració (disseny, desenvolupament, test, etc.). sigla ED

gràfic burndown *m* gràfic que representa la tendència de la feina feta al llarg d'un *sprint*, o bé al llarg de tot el desenvolupament. L'eix x mostra el transcurs del temps; l'eix y mostra el treball restant.

gremi *m* Grup mixt de persones diferents amb els mateixos interessos i que comparteixen coneixement. En anglès es diu *guild*.

història d'usuari *f* Descripció de la usabilitat i la funcionalitat que s'espera del sistema, redactada pel client final, en el seu llenguatge i sense tecnicismes. En anglès es diu *user story*.

increment *m* Desenvolupament realitzat durant un *sprint* que dona valor al negoci.

Independent negotiable valuable estimable small testable *m* Principis seguits per Scrum per assegurar la qualitat en la definició d'un ítem del *backlog*. sigla INVEST

INVEST *m* Vegeu *Independent negotiable valuable estimable small testable*.

ítem de la backlog del producte *m* Ítem que representa una *story* dins el *backlog* del producte. sigla PBI

iteració *f* Cicle amb durada definida, amb un màxim de trenta dies consecutius, dins d'Scrum que representa un *sprint*.

lean production *f* Vegeu producció ajustada.

metodologies àgils *f pl* Metodologies que compleixen el manifest àgil.

MVP *m* Vegeu producte mínim viable.

PBI *m* Vegeu ítem de la *backlog* del producte.

persona interessada *f* Persona que té un interès a obtenir el desenvolupament d'un producte, bé perquè desitja usar-lo o perquè serà afectat per ell.

PO *m* Vegeu **client**.

producció ajustada *m* Mètode originat en el sistema de producció de Toyota que es basa en la millora contínua i la reducció de residus. Forma part de la base de les metodologies àgils.

product backlog item *f* Vegeu ítem de la **backlog** del producte.

producte mínim viable *m* Producte que conté les funcionalitats imprescindibles per satisfer alguna necessitat identificada pel client.
sigla MVP

Scrum *m* Terme usat en rugbi que, traduït a l'espanyol, significa 'melé'. És un mètode que es fa servir per a restablir el joc de nou en aquest determinat esport.

Scrum master *m* Membre de l'equip Scrum que ajuda a que l'equip segueixi els valors i principis Scrum i actua com a facilitador (organitza les cerimònies Scrum, resol impediments, etc.).
sigla SM

SM *m* Vegeu **Scrum master**.

sprint *f* Iteració d'entre una i quatre setmanes durant la qual es desenvolupa un producte determinat.

squad *m* Equip multifuncional de menys de vuit persones.

stakeholder *m* Vegeu **persona interessada**.

Bibliografia

Hammarberg, Marcus; Sundén, Joakim (2014). *Kanban in Action*. Manning Publications.

Hundhausen, Richard (2012). *Professional Scrum Development with Microsoft Visual Studio®*. Microsoft Press.

Kantsev, Veselin (2017). *Implementing devops on AWS*. Packt Publishing.

Keith, Clinton (2010). *Agile Game Development with Scrum*. Addison-Wesley Professional.

Linz, Tell (2014). *Testing in Scrum*. Rocky Nook.

Matyas, Steve; Duvall, Paul M.; Glover, Andrew (2007). *Continuous Integration: Improving Programari Quality and Reducing Risk*. Addison-Wesley Professional.

Measey, Peter (2015). *Agile Foundations - Principles, practices and frameworks*. BCS Learning & Development Limited.

Rubin, Kenneth S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional.

Schwaber, Ken (2004). *Agile Project Management with Scrum*. Microsoft Press.

Schwaber, Ken; Derby, Esther; Larsen, Diana (2006). *Agile Retrospectives - Making Good Teams Great*. Pragmatic Bookshelf.

Takeuchi, Hirotaka; Nonaka, Ikujiro (1986, gener). «The New New Product Development Game» [en línia]. *Harvard Business Review*. <<https://hbr.org/1986/01/the-new-new-product-development-game>>

Willis, John; Debois, Patrick; Humble, Jez; Kim, Gene (2016). *The devops Handbook*. IT Revolution Press.

Webgrafia

<<http://agilemanifesto.org/>>

<<https://www.agilealliance.org/>>

<<http://scrumtrainingseries.com/>>

<<http://scrumreferencecard.com>>

<<https://www.atlassian.com/devops>>