Kehitysmenetelmät

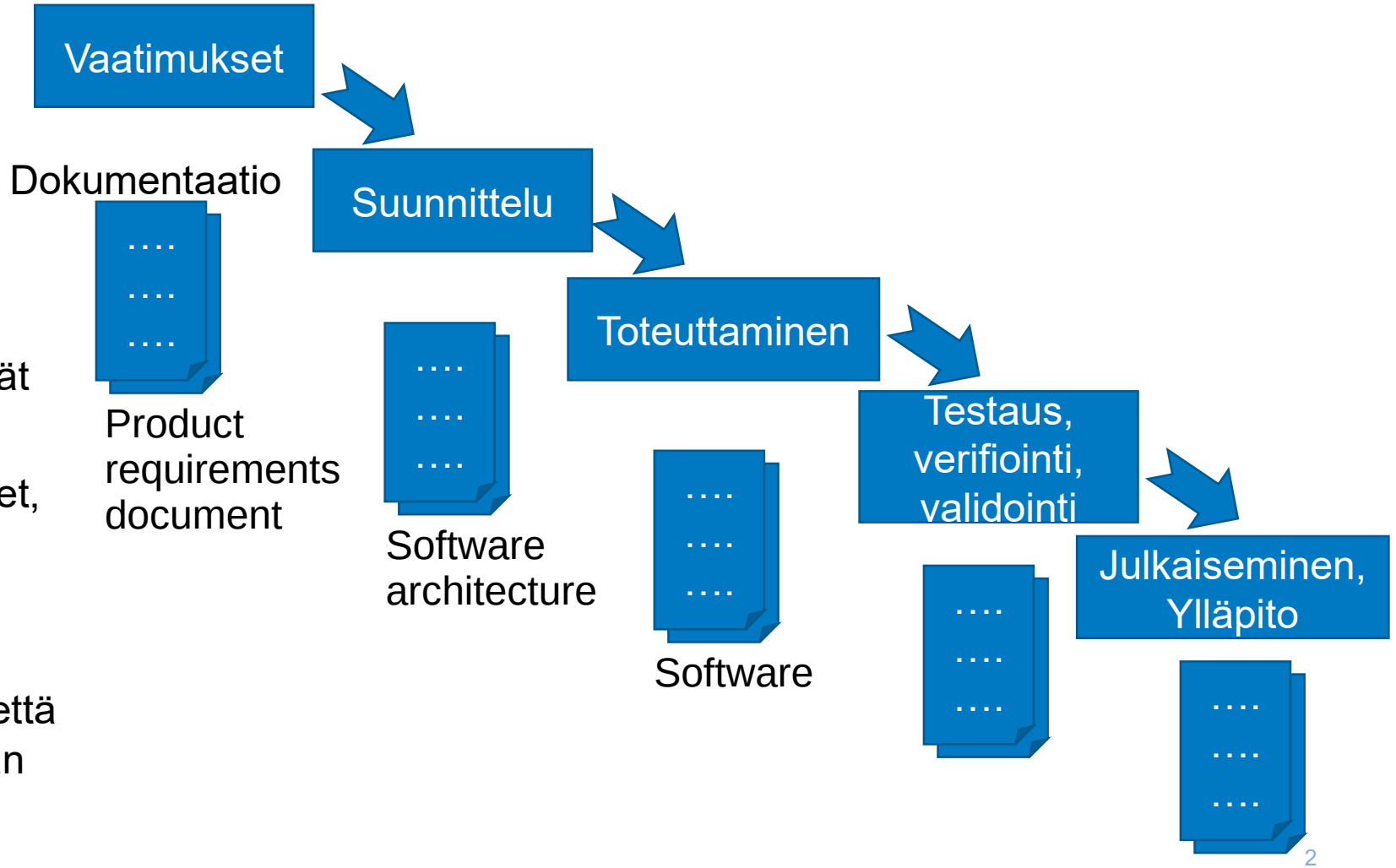
Software development methods

20.04.2021

Sauli Isonikkilä

Vesiputousmalli, (Software Waterfall development)

- Requirements analysis
- Software design
- Software implementation
- Verification
- Maintenance
- Soveltuu asiakkaille, jotka jo tietävät mitä haluavat.
- Kuten sairaalat, koulut, palolaitokset, armeija, poliisi yms.
- Tällaisille asiakkaille ohjelmiston dokumentaatio on tärkeää. Sitä käytetään koulutukseen ja siihen, että voidaan todeta ohjelmiston toimivan miinkuin pitää.



- Dokumenttaation tekeminen on siis keskeisessä roolissa, jotta voidaan todeta ohjelmiston olevan turvallinen, lakien ja asetusten mukainen tms.
- Vesiputousmalli ei puolestaan sovellu sellaisiin tilanteisiin, joissa asiakas ei tiedä vielä mitä haluaa tai tehdään täysin uutta ohjelmistoa jollaista ei ole koskaan ennen tehty.

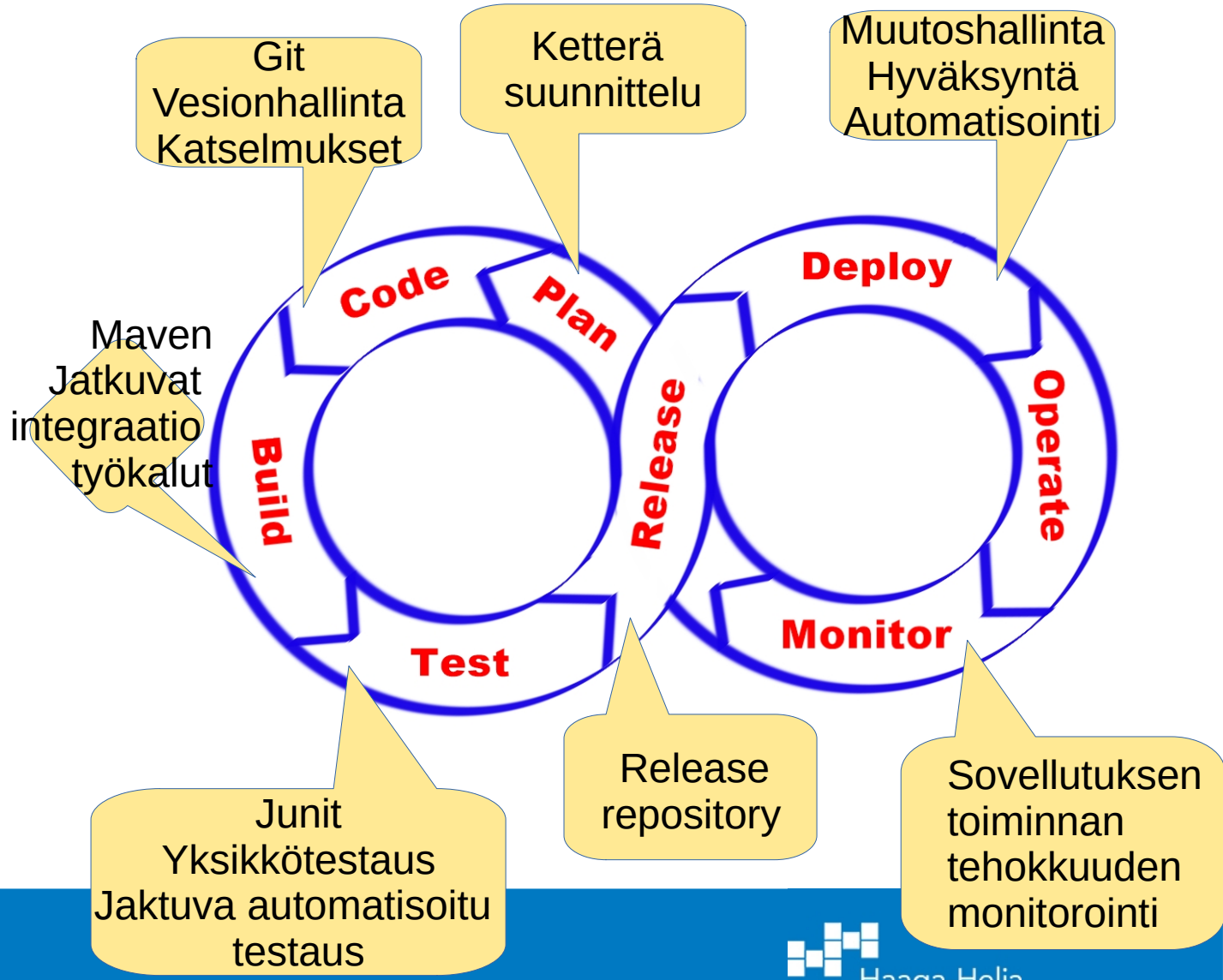
Ketterät menetelmät (Software agile methods)

- For example:
  - DevOps
  - Extreme Programming (XP)
  - Feature-driven development (FDD)
  - Kanban
  - Lean software development
  - Scrum
  - Scrumban
  - Scaled Agile Framework (SAFe)
  - Jne...

- Agile software development principles:
  - Customer satisfaction
  - Welcome changing requirement
  - Deliver working software frequently
  - Close and daily cooperation
  - Motivated individuals
  - Face-to-face communication
  - Working software is the measure of progress
  - Sustainable development
  - Simplicity
  - Self-organizing teams
  - Teams reflect and become more effective

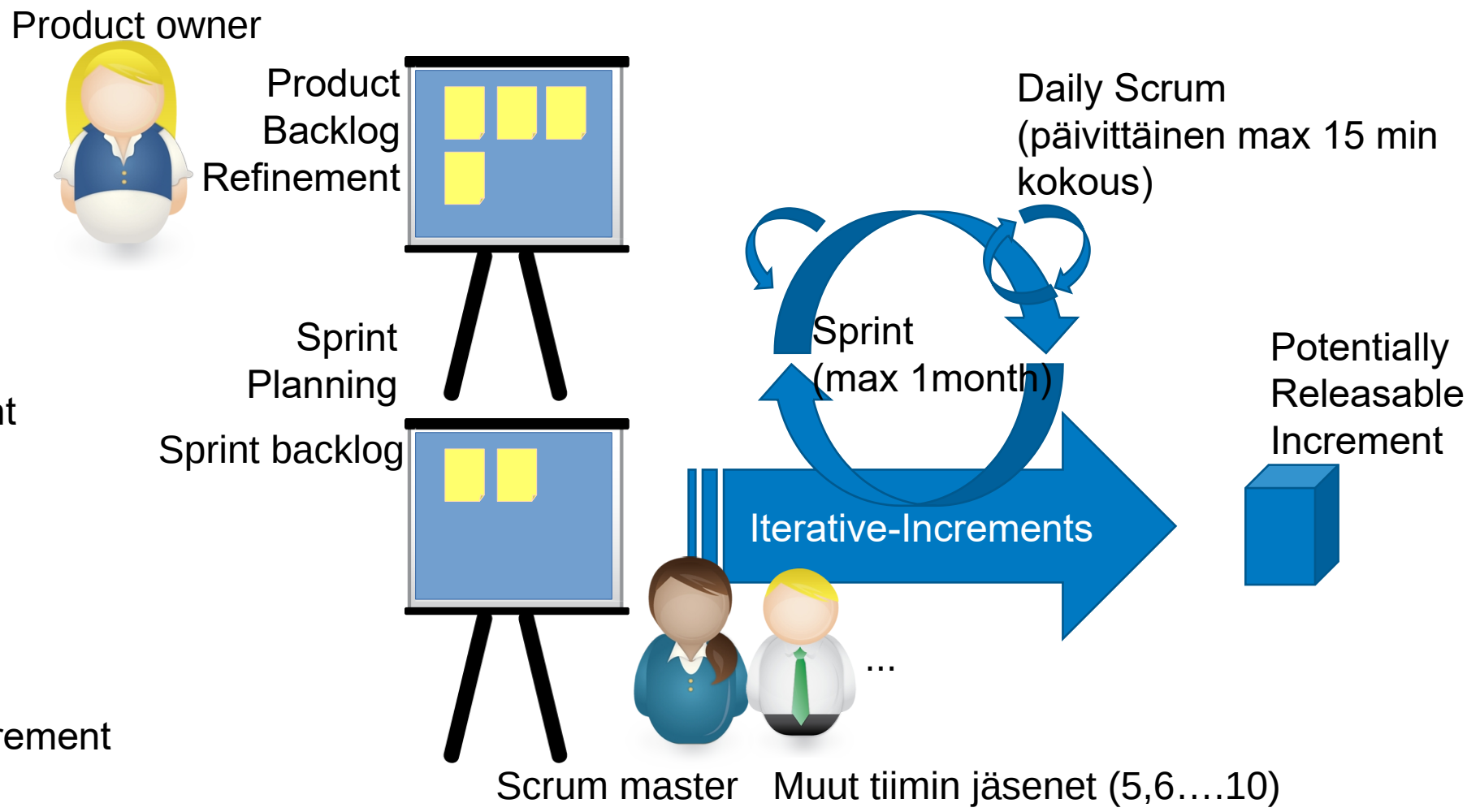
DevOps (Development-Operations)

- DevOps:n useat osa-alueet ovat ketteristä menetelmistä peräisin.
- DevOps:sa yhdistetään ohjelmisto kehityksen (software development) kanssa IT-operaatiot. Tavoitteena on lyhentää ohjelmistokehitystä ja tehdä julkaiseminen jatkuvana toimituksena (continious delivery).
- DevOps:ssa on tavoitteena myös korkea laatu.
- DevOps:ssa käytetään työn tukena sitä automatisoivia työkaluja. Niiden avulla saadaan työvaiheista jatkuvia (jatkuva toimitus, jatkuva testaus, jatkuva muutos hallinta jne.)
- DevOps:sta on useita eri versioita käytössä eri yrityksissä.



Scrum (Software Scrum)

- Product Owner
- Product backlog
- Development team
- Sprint planning
- Sprint backlog
- Sprint
- Product backlog refinement
- Scrum Master
- Daily Scrum
- Sprint Review
- Sprint Retrospective
- Potentially Releasable Increment



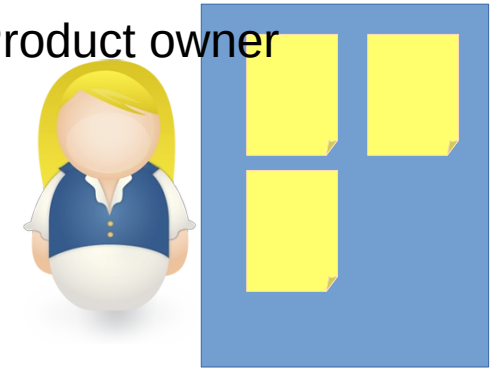
Scrum master

Muut tiimin jäsenet (5,6….10)

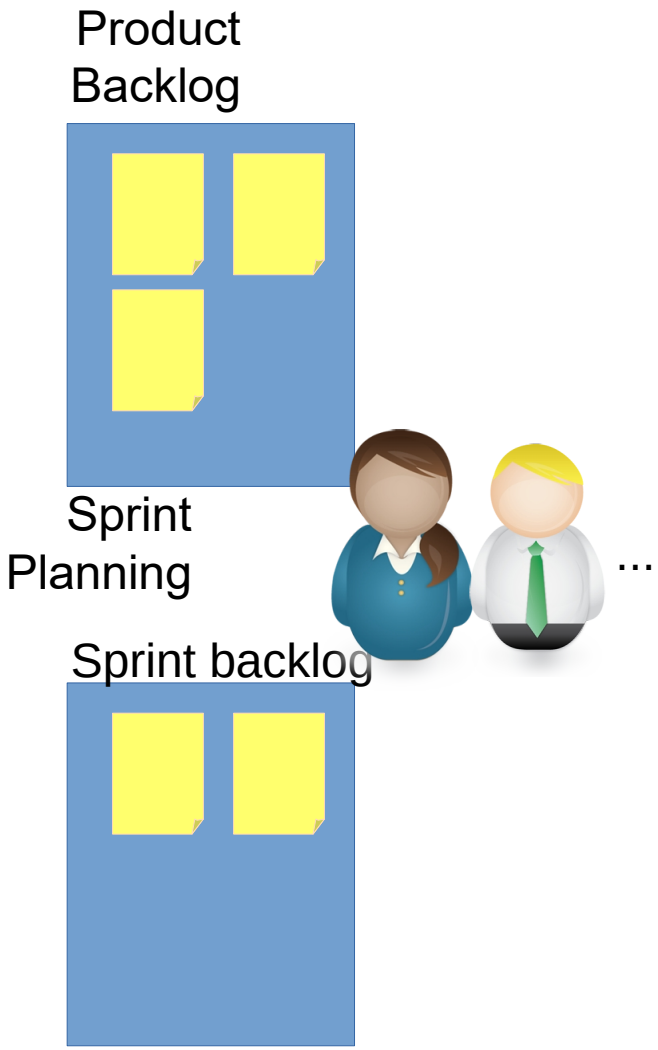
- Scrum on ketterä menetelmä, joka antaa ohjeita siitä miten kehitystiimi toimii keskenään.
- Soveltuu siis ketteryytensä vuoksi silloin, kun asiakas ei tiedä mitä haluaa. Tai ollaan tekemässä jotain sellaista, mitä koskaan ennen ei ole tehty. Silloin voidaan kehitys kierros kehitys kierrokselta (sprint) tehdä lisää kehitettävää sovellutusta ja näyttää sitä asiakkaalle.
- Scrum tiimiin kuuluu korkeintaan 10 kehittäjää. Heillä kullakin on roolinsa tiimissä (esim. dokumentoija, vaatimusmäärittelijä, ohjelmoija, testaaja jne).
- Yksi kehitystiimin (scrum tiimi) jäsen on Scrum master. Hän on kehitystiimin mahdollistaja (ei siis ole projektipäällikkö). Raivaa tiimin edestä kaikki kehitystyön esteet (järjestää koulutuksen, jos sitä ei ole riittävästi, hankki tarvittavat ohjelmistot ja laitteet jos niitä puuttuu tms.). Scrum master hoitaa myös tiimin ja ulkopuolisten välisen kommunikaation sprintin eli kehitysjakson aikana.
- Product owner puolestaan valvoo asiakkaan etua, jos asiakas itse ei voi kehityspalavereihin (sprint review) osallistua. Product owner pitää yllä Product Backlog listaa.

Product Backlog Refinement

Product owner



- Kun Product owner (tuoteomistaja) on saanut selville ne ominaisuudet, jotka kehitettävässä sovelluksessa on.
- Valitsee kehitystiimi Sprint planning kokouksessa ne ominaisuuden, jotka kukin tiimin jäsen haluaa toteuttaa sprintin (2-4 viikkoinen kehitysjakso). Ominaisuudet valitaan tietysti asiakkaan mukaan tärkeimpien ominaisuuksien joukosta.
- Daily Scrum on puolestaan päivittäin järjestettävä max. 15 min kokous, jossa kukin tiimin jäsen kertoo mitä on siihen mennessä ehtinyt saada aikaan ja mitä aikoo tehdä päivän aikana seuraavaksi.
- Sprint eli kehitysjakso voi Scrum:issa olla 2-4 viikkoa pitkä yrityksestä riippuen. Sprint alkaa Sprintin suunnittelulla (Sprint planning). Sitä seuraa kehitysjakso joissa kunkin päivän alussa on lyhyt Daily Scrum kokous.
- Kehitysjakson eli Sprintin lopussa on Sprint review ja Sprint retrospective kokoukset.
- Seuraava Sprint alkaa jälleen Sprint planning kokouksella.
- Sprint Review kokouksessa kehitystiimi esittää Sprintin aikana tehty sovellus intressiryhmille.

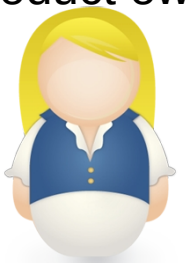


- Sprint Review kokouksessa kehitystiimi esittää Sprintin aikana tehty sovellus intressiryhmille.
- Sprint retrospective kokous seuraa heti edellisen jälkeen. Siinä keskustellaan siitä miten Sprint meni ja miten sitä voisi kehittää. Onnistuiko Sprint planning kokouksen arviot ominaisuuksien vaatineesta työajasta. Ja jos ei niin miten sitä voisi tarkentaa paremmaksi. Mitä tehtäisiin erilaille seuraavassa Sprint:ssä?
- Burndown Chart kaaviota käytetään usein Scrum:ssa, näyttämään kuinka paljon vielä on tekemätöntä työtä Sprint:ssä.

Kehitystiimi

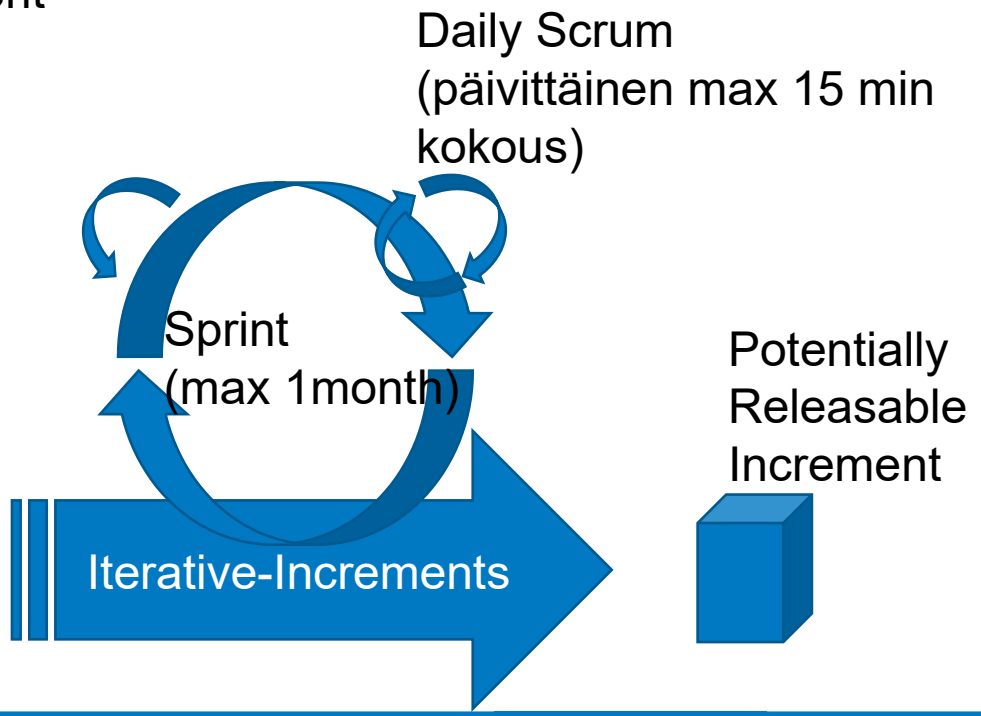
...

Product owner



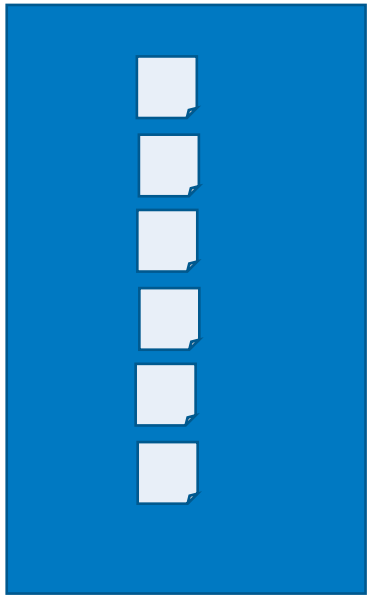
Intressiryhmät

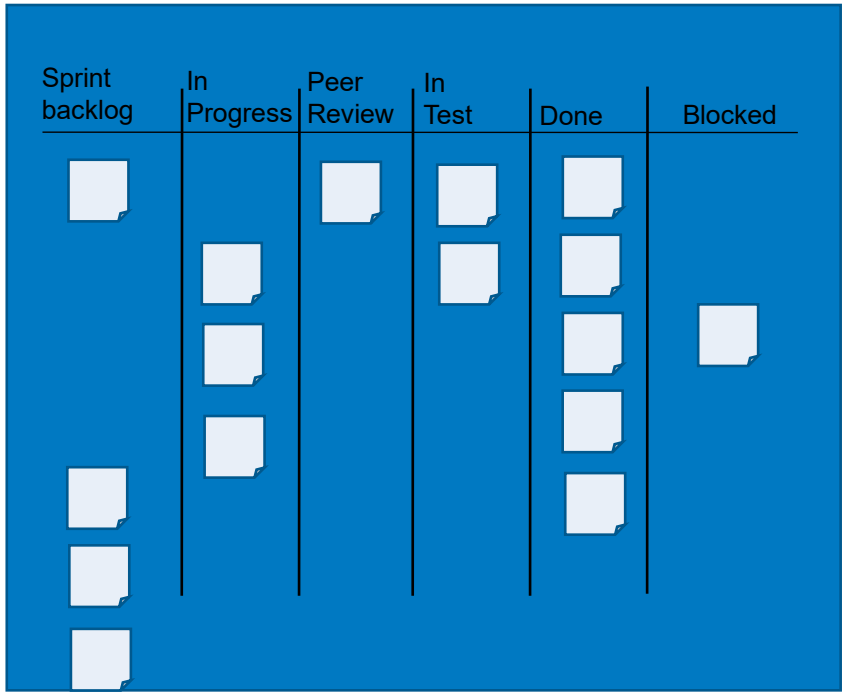
- Definition od done (DoD) määrittele milloin tehtävä (tai ominaisuus) on valmis.
- Definition of Ready (DoR) määrittelee milloin specifikaatio tai määrittely on riittävän hyvin tehty, jotta voidaan alkaa tehtävää (tai ominaisuus) tekemään.
- Increment (kehityskierros) jatkuu edellisestä Sprint:in aikana valmistuneesta, joka on Sprintin tavoitteen mukainen ja täyttää Definition of Done kriteerit kehitystiimissä.



Kanban
(Software Kanban Development)

1. To make the software development visible
2. Using Kanban board





Lean
(Lean Software Development)

Lean principles:

1. Eliminate waste
2. Amplify learning
3. Decide as late as possible
4. Deliver as fast as possible
5. Empower the team
6. Build integrity in
7. Optimize the Whole

Kiitos