

Ohjelmistotuotanto

Matti Luukkainen ja ohjaajat Jami Kousa, Tero Tapio, Mauri
Karlin

syksy 2019

Luento 2

29.10.2019

Vesiputousmallin suurimmat ongelmat

- ▶ Vaatimukset on mahdotonta määritellä tyhjentävästi projektin alkuvaiheessa
 - ▶ asiakas ei ymmärrä vielä alussa mitä haluaa
 - ▶ Bisnesympäristö muuttuu projektin kuluessa

Vesiputousmallin suurimmat ongelmat

- ▶ Vaatimukset on mahdotonta määritellä tyhjentävästi projektin alkuvaiheessa
 - ▶ asiakas ei ymmärrä vielä alussa mitä haluaa
 - ▶ Bisnesympäristö muuttuu projektin kuluessa
- ▶ Suunnittelu sillä tasolla, että ohjelmointi on suoraviivainen “rakennusvaihe” on mahdotonta
 - ▶ ohjelmointi on osa suunnitteluprosessia, ohjelmakoodi on tuotteen lopullinen suunnitelma
 - ▶ suunnittelu taas on teknisesti haastavaa, riskejä sisältävää toimintaa

Vesiputousmallin suurimmat ongelmat

- ▶ Vaatimukset on mahdotonta määritellä tyhjentävästi projektin alkuvaiheessa
 - ▶ asiakas ei ymmärrä vielä alussa mitä haluaa
 - ▶ Bisnesympäristö muuttuu projektin kuluessa
- ▶ Suunnittelu sillä tasolla, että ohjelmointi on suoraviivainen “rakennusvaihe” on mahdotonta
 - ▶ ohjelmointi on osa suunnitteluprosessia, ohjelmakoodi on tuotteen lopullinen suunnitelma
 - ▶ suunnittelu taas on teknisesti haastavaa, riskejä sisältävää toimintaa
- ▶ Vasta lopuksi tehtävä testaus paljastaa ongelmat liian myöhään
 - ▶ ongelmien korjaaminen voi edellyttää kalliita muutoksia

Ohjelmiston tuottaminen ei ole kontrolloitu prosessi

- ▶ 90-luvun iteratiiviset prosessimallit korjaavat monia edellisen kalvon epäkohdista
- ▶ ne kuitenkin olivat tarkkoihin etukäteissuunnitelmiin perustuvia
 - ▶ Tarkka projektisuunnitelma ja sen noudattaminen
 - ▶ Selkeä roolijako: projektipäälliköt, analyytikot, arkkitehdit, ohjelmoijat, testaajat

Ohjelmiston tuottaminen ei ole kontrolloitu prosessi

- ▶ 90-luvun iteratiiviset prosessimallit korjaavat monia edellisen kalvon epäkohdista
- ▶ ne kuitenkin olivat tarkkoihin etukäteissuunnitelmiin perustuvia
 - ▶ Tarkka projektisuunnitelma ja sen noudattaminen
 - ▶ Selkeä roolijako: projektipäälliköt, analyytikot, arkkitehdit, ohjelmoijat, testaajat
- ▶ eli ne olettivat että ohjelmistotuotanto on jossain määrin *kontrolloitavissa oleva prosessi*

Ketterien menetelmien perusolettamuksia

- ▶ Useimmat ohjelmistoprojektit ovat laadultaan uniikkeja
 - ▶ Vaatimukset erilaiset kuin millään jo tehdyllä ohjelmistolla
 - ▶ Uusi tekijätiimi, omanlaisilla kompetensseilla ja persoonallisuuksilla varustettu
 - ▶ Toteutusteknologiat kehittyvät, joten tehdään todennäköisesti tavalla, joka ei ole kaikille tuttu

Ketterien menetelmien perusolettamuksia

- ▶ Useimmat ohjelmistoprojektit ovat laadultaan uniikkeja
 - ▶ Vaatimukset erilaiset kuin millään jo tehdyllä ohjelmistolla
 - ▶ Uusi tekijätiimi, omanlaisilla kompetensseilla ja persoonallisuuksilla varustettu
 - ▶ Toteutusteknologiat kehittyvät, joten tehdään todennäköisesti tavalla, joka ei ole kaikille tuttu
- ▶ Järkevää lähteä oletuksesta että kyseessä ei ole *kontrolloitu prosessi*, joka voidaan tarkkaan etukäteen aikatauluttaa ja suunnitella
- ▶ Parempi ajatella tuotekehitysprojektina, näiden kontrollointiin sopii paremmin ns. *empiirinen prosessi*
 - ▶ Toiminnan periaatteina *transparency, inspection, adaption*

Ketterien menetelmien perusolettamuksia

- Tekijät yksilöitä, oletetaan että yksilöt toimivat paremmin kun heihin luotetaan ja annetaan tiimille vapaus organisoida itse toimintansa

Ketterien menetelmien perusolettamuksia

- ▶ Tekijät yksilöitä, oletetaan että yksilöt toimivat paremmin kun heihin luotetaan ja annetaan tiimille vapaus organisoida itse toimintansa
- ▶ “The whole team”-periaate: tiimi kollektiivina vastuussa aikaansaannoksesta
- ▶ Oletuksena että perinteinen command-and-control ja jako eri vastuu- alueisiin (suunnittelija, ohjelmoija, testaaja) ei tuota optimaalista tulosta

Eilen käsitelty *ketterän manifesti* heijastelee näitä olettamuksia

- ▶ Tutustumme kurssillat Scrumiin, joka on tällä hetkellä selvästi suosituin ketterä menetelmä/prosessimalli

Scrum

- ▶ Tutustumme kurssillat Scrumiin, joka on tällä hetkellä selvästi suosituin ketterä menetelmä/prosessimalli
- ▶ Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value

Scrum

- ▶ Tutustumme kurssillat Scrumiin, joka on tällä hetkellä selvästi suosituin ketterä menetelmä/prosessimalli
- ▶ Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value
- ▶ Scrum is:
 - ▶ Lightweight
 - ▶ Simple to understand
 - ▶ Extremely difficult to master

Scrum

- ▶ Tutustumme kurssillat Scrumiin, joka on tällä hetkellä selvästi suosituin ketterä menetelmä/prosessimalli
- ▶ Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value
- ▶ Scrum is:
 - ▶ Lightweight
 - ▶ Simple to understand
 - ▶ Extremely difficult to master
- ▶ Scrum is not a process or a technique for building products; it is a framework within which you can employ various processes and techniques
- ▶ Scrum makes clear the relative efficacy of your management and development practices so that you can improve

Scrum lyhyesti

- ▶ Iteratiivinen ja inkrementaalinen menetelmä (tai kehittäjiensä mukaan framework eli menetelmäkehys)
- ▶ Kehitys tapahtuu 1-4 viikon iteraatioissa, joita Scrumissa kutsutaan *sprinteiksi*

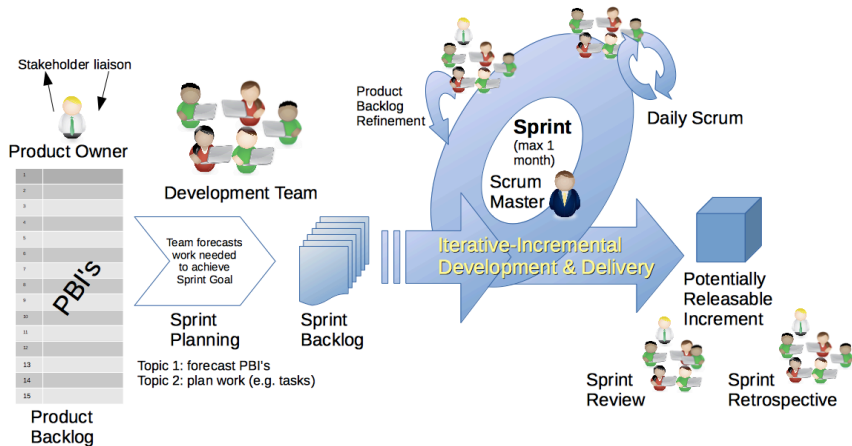
Scrum lyhyesti

- ▶ Iteratiivinen ja inkrementaalinen menetelmä (tai kehittäjiensä mukaan framework eli menetelmäkehys)
- ▶ Kehitys tapahtuu 1-4 viikon iteraatioissa, joita Scrumissa kutsutaan *sprinteiksi*
- ▶ *Scrum-tiimi* koostuu 3-9:stä kehittäjästä
- ▶ *Scrum master* toimii tiimin apuna ohjaten mm. prosessin noudattamisessa sekä toimien rajapintana yrityksen hallintoon
- ▶ *Product owner* eli tuotteen omistaja hallinnoi projektin backlogia
 - ▶ *backlog* sisältää priorisoidussa järjestyksessä projektissa toteutettavan ohjelmiston ominaisuudet/vaatimukset/toiminnot

Scrum lyhyesti

- ▶ Iteratiivinen ja inkrementaalinen menetelmä (tai kehittäjiensä mukaan framework eli menetelmäkehys)
- ▶ Kehitys tapahtuu 1-4 viikon iteraatioissa, joita Scrumissa kutsutaan *sprinteiksi*
- ▶ *Scrum-tiimi* koostuu 3-9:stä kehittäjästä
- ▶ *Scrum master* toimii tiimin apuna ohjaten mm. prosessin noudattamisessa sekä toimien rajapintana yrityksen hallintoon
- ▶ *Product owner* eli tuotteen omistaja hallinnoi projektin backlogia
 - ▶ *backlog* sisältää priorisoidussa järjestyksessä projektissa toteutettavan ohjelmiston ominaisuudet/vaatimukset/toiminnot
- ▶ Jokaisen sprintin alussa tiimi valitsee projektin backlogista sprintin aikana toteutettavat vaatimukset
- ▶ Sprintin aikana scrum-tiimi toteuttaa itseorganisoidusti sprintiin valitut vaatimukset lopputuloksena vaatimusten osalta toimiva ohjelmisto

Scrum kuvana



Scrum: roles, artifacts and events

Terminologiaa

- ▶ Scrum määrittelee 3 erilaista roolia:
 - ▶ Kehittäjä
 - ▶ Scrum master
 - ▶ Product owner

Scrum: roles, artifacts and events

Terminologiaa

- ▶ Scrum määrittelee 3 erilaista roolia:
 - ▶ Kehittäjä
 - ▶ Scrum master
 - ▶ Product owner
- ▶ Scrumiin kuuluvat artefaktit eli "konkreettiset asiat" ovat
 - ▶ Product backlog eli projektin kehitysjono
 - ▶ Sprint backlog eli sprintin tehtävälista
 - ▶ Työn alla olevan ohjelmiston uudet versiot (product increment)

Scrum: roles, artifacts and events

Terminologiaa

- ▶ Scrum määrittelee 3 erilaista roolia:
 - ▶ Kehittäjä
 - ▶ Scrum master
 - ▶ Product owner
- ▶ Scrumiin kuuluvat artefaktit eli "konkreettiset asiat" ovat
 - ▶ Product backlog eli projektin kehitysjono
 - ▶ Sprint backlog eli sprintin tehtävälista
 - ▶ Työn alla olevan ohjelmiston uudet versiot (product increment)
- ▶ Scrumissa tekeminen rytmittyy sprintteihin eli 1-4 viikon mittaisiin iteraatioihin

Scrum: roles, artifacts and events

Terminologiaa

- ▶ Scrum määrittelee 3 erilaista roolia:
 - ▶ Kehittäjä
 - ▶ Scrum master
 - ▶ Product owner
- ▶ Scrumiin kuuluvat artefaktit eli "konkreettiset asiat" ovat
 - ▶ Product backlog eli projektin kehitysjono
 - ▶ Sprint backlog eli sprintin tehtävälista
 - ▶ Työn alla olevan ohjelmiston uudet versiot (product increment)
- ▶ Scrumissa tekeminen rytmittyy sprintteihin eli 1-4 viikon mittaisiin iteraatioihin
- ▶ Sprintteihin kuuluu muutamia standardipalavereja (events):
 - ▶ Sprintin suunnittelupalaveri
 - ▶ Daily scrum -palaverit
 - ▶ Sprintin katselmointi
 - ▶ Retrospektiivi

Product backlog

- ▶ Priorisoitu lista asiakkaan tuotteelle asettamista *vaatimuksista*
 - ▶ asiakkaan tasolla olevia mielekkäitä toiminnallisuuksia, kirjattuna asiakkaan ymmärtämällä kielellä

Product backlog

- ▶ Priorisoitu lista asiakkaan tuotteelle asettamista *vaatimuksista*
 - ▶ asiakkaan tasolla olevia mielekkäitä toiminnallisuuksia, kirjattuna asiakkaan ymmärtämällä kielellä
- ▶ Korkeimman prioriteetin vaatimukset valitaan toteutettavaksi seuraavan sprintin aikana
 - ▶ ne ovat kirjattu tarkemmin kuin backlogin häntäpään vaatimukset

Product backlog

- ▶ Priorisoitu lista asiakkaan tuotteelle asettamista *vaatimuksista*
 - ▶ asiakkaan tasolla olevia mielekkäitä toiminnallisuuksia, kirjattuna asiakkaan ymmärtämällä kielellä
- ▶ Korkeimman prioriteetin vaatimukset valitaan toteutettavaksi seuraavan sprintin aikana
 - ▶ ne ovat kirjattu tarkemmin kuin backlogin häntäpään vaatimukset
- ▶ Vaatimukset ovat usein *estimoituja* eli niiden toteutuksen vaatima työmäärä on arvioitu
 - ▶ Työmääräarviot tekee kehittäjätiimi

Product backlog

- ▶ Priorisoitu lista asiakkaan tuotteelle asettamista *vaatimuksista*
 - ▶ asiakkaan tasolla olevia mielekkäitä toiminnallisuuksia, kirjattuna asiakkaan ymmärtämällä kielellä
- ▶ Korkeimman prioriteetin vaatimukset valitaan toteutettavaksi seuraavan sprintin aikana
 - ▶ ne ovat kirjattu tarkemmin kuin backlogin häntäpään vaatimukset
- ▶ Vaatimukset ovat usein *estimoituja* eli niiden toteutuksen vaatima työmäärä on arvioitu
 - ▶ Työmääräarviot tekee kehittäjätiimi
- ▶ Scrum ei määrittele missä muodossa backlog ja siinä olevat vaatimukset esitetään
 - ▶ nykyään käytetään usein *user story* -formaattia

Product owner

- ▶ Scrumin mukaan kuka vaan voi milloin tahansa lisätä backlogiin vaatimuksia
- ▶ Backlogia priorisoi ainoastaan *product owner* eli tuotteen omistaja

Product owner

- ▶ Scrumin mukaan kuka vaan voi milloin tahansa lisätä backlogiin vaatimuksia
- ▶ Backlogia priorisoi ainoastaan *product owner* eli tuotteen omistaja
- ▶ Product owner on yksittäinen henkilö
 - ▶ Priorisointiin voi toki olla vaikuttamassa useampikin henkilö
 - ▶ Product owner tekee lopulliset päätökset prioriteettien suhteen

Product owner

- ▶ Scrumin mukaan kuka vaan voi milloin tahansa lisätä backlogiin vaatimuksia
- ▶ Backlogia priorisoi ainoastaan *product owner* eli tuotteen omistaja
- ▶ Product owner on yksittäinen henkilö
 - ▶ Priorisointiin voi toki olla vaikuttamassa useampikin henkilö
 - ▶ Product owner tekee lopulliset päätökset prioriteettien suhteen
- ▶ Product owner on vastuussa backlogista
 - ▶ Priorisoi vaatimukset maksimoiden asiakkaan tuotteesta saaman hyödyn
 - ▶ Varmistaa että kehittäjätiimi ymmärtää toteutettavaksi valitut vaatimukset

Scrum master

- ▶ Tiimeillä on *scrum master*, eli henkilö joka huolehtii siitä että ohjelmistokehitys etenee sujuvasti

Scrum master

- ▶ Tiimeillä on *scrum master*, eli henkilö joka huolehtii siitä että ohjelmistokehitys etenee sujuvasti
- ▶ Ei perinteinen projektipäällikkö vaan *servant-leader*
 - ▶ huolehtii että scrumia noudatetaan järkevällä tavalla
 - ▶ rohkaisee ja auttaa tiimiä itseorganisoitumisessa
 - ▶ opastaa hyvien käytänteiden noudattamisessa
 - ▶ järjestää Scrumiin liittyvät palaverit

Scrum master

- ▶ Tiimeillä on *scrum master*, eli henkilö joka huolehtii siitä että ohjelmistokehitys etenee sujuvasti
- ▶ Ei perinteinen projektipäällikkö vaan *servant-leader*
 - ▶ huolehtii että scrumia noudatetaan järkevällä tavalla
 - ▶ rohkaisee ja auttaa tiimiä itseorganisoitumisessa
 - ▶ opastaa hyvien käytänteiden noudattamisessa
 - ▶ järjestää Scrumiin liittyvät palaverit
- ▶ Pyrkii poistamaan kehitystyön esteitä
 - ▶ Esteenä voi olla jokin tiimistä riippumaton asia, jonka poistamiseksi Scrum master joutuu neuvottelemaan yrityksen hallinnon kanssa
 - ▶ Este voi myös liittyä ryhmän työtapoihin, tällöin Scrum master opastaa ryhmää toimimaan siten, että este poistuu

Scrum master

- ▶ Tiimeillä on *scrum master*, eli henkilö joka huolehtii siitä että ohjelmistokehitys etenee sujuvasti
- ▶ Ei perinteinen projektipäällikkö vaan *servant-leader*
 - ▶ huolehtii että scrumia noudatetaan järkevällä tavalla
 - ▶ rohkaisee ja auttaa tiimiä itseorganisoitumisessa
 - ▶ opastaa hyvien käytänteiden noudattamisessa
 - ▶ järjestää Scrumiin liittyvät palaverit
- ▶ Pyrkii poistamaan kehitystyön esteitä
 - ▶ Esteenä voi olla jokin tiimistä riippumaton asia, jonka poistamiseksi Scrum master joutuu neuvottelemaan yrityksen hallinnon kanssa
 - ▶ Este voi myös liittyä ryhmän työtapoihin, tällöin Scrum master opastaa ryhmää toimimaan siten, että este poistuu
- ▶ Scrum master tekee kaikkensa, jotta tiimillä olisi optimaaliset olosuhteen kehittää tuotetta

Kehittäjätiimi

- ▶ Kehittäjätiimi koostuu noin 3-10:stä henkilöstä, kaikista käytetään nimikettä developer
 - ▶ vaikka kaikilla nimike developer, voivat jotkut tiimin jäsenistä ovat erikoistuneet omaan osa-alueeseensa (esim. testaaminen)
 - ▶ loko tiimi kuitenkin kantaa aina yhteisen vastuun kehitystyöstä

Kehittäjätiimi

- ▶ Kehittäjätiimi koostuu noin 3-10:stä henkilöstä, kaikista käytetään nimikettä developer
 - ▶ vaikka kaikilla nimike developer, voivat jotkut tiimin jäsenistä ovat erikoistuneet omaan osa-alueeseensa (esim. testaaminen)
 - ▶ loko tiimi kuitenkin kantaa aina yhteisen vastuun kehitystyöstä
- ▶ Oletuksena on että tiimin jäsenet työskentelevät tiimissä 100%:lla työajalla
- ▶ Koko tiimin tulee oletusarvoisesti työskennellä samassa paikassa, mieluiten yhteisessä tiimille varatussa avokonttorissa

- ▶ tiimi on *cross-functional*, eli sen jäsenten tulisi sisältää kaikki tarvittava osaaminen järjestelmän suunnitteluun, toteuttamiseen ja testaamiseen

Kehittäjätiimi

- ▶ tiimi on *cross-functional*, eli sen jäsenten tulisi sisältää kaikki tarvittava osaaminen järjestelmän suunnitteluun, toteuttamiseen ja testaamiseen
- ▶ pääperiaatteena on että kehitystiimiä ei johdeta ulkopuolelta
 - ▶ päättää mihin tavoitteisiin se kussakin sprintissä sitoutuu, eli miten paljon vaatimuksia backlogista valitaan sprintissä toteutettavaksi
 - ▶ päättää myös (tiettyjen reunaehtojen puitteissa) itse miten se sprintin tavoitteen toteuttaa
- ▶ tiimi on siis *itseorganisoituva* (self organizing)

Sprintti

- ▶ Scrumissa kehitystyö siis jakautuu 1-4 viikon mittaisiin iteraatioihin eli sprintteihin
 - ▶ nykyään suosituin sprintin pituus lienee 2 viikkoa
- ▶ Sprintti on *time-boxed*, sitä ei missään olosuhteissa pidennetä

Sprintti

- ▶ Scrumissa kehitystyö siis jakautuu 1-4 viikon mittaisiin iteraatioihin eli sprintteihin
 - ▶ nykyään suosituin sprintin pituus lieenee 2 viikkoa
- ▶ Sprintti on *time-boxed*, sitä ei missään olosuhteissa pidennetä
- ▶ Jokaisen sprintin alussa tiimi valitsee projektin backlogista sprintin aikana toteutettavat vaatimukset
 - ▶ Backlog on priorisoitu ja vaatimukset valitaan aina priorisoidun listan kärjestä

Sprintti

- ▶ Scrumissa kehitystyö siis jakautuu 1-4 viikon mittaisiin iteraatioihin eli sprintteihin
 - ▶ nykyään suosituin sprintin pituus lienee 2 viikkoa
- ▶ Sprintti on *time-boxed*, sitä ei missään olosuhteissa pidennetä
- ▶ Jokaisen sprintin alussa tiimi valitsee projektin backlogista sprintin aikana toteutettavat vaatimukset
 - ▶ Backlog on priorisoitu ja vaatimukset valitaan aina priorisoidun listan kärjestä
- ▶ Tiimi valitsee sprinttiin ainoastaan sen verran toteutettavaa minkä valmistumiseen se uskoo kykenevänsä sitoutumaan

Sprintti

- ▶ Scrumissa kehitystyö siis jakautuu 1-4 viikon mittaisiin iteraatioihin eli sprintteihin
 - ▶ nykyään suosituin sprintin pituus lieenee 2 viikkoa
- ▶ Sprintti on *time-boxed*, sitä ei missään olosuhteissa pidennetä
- ▶ Jokaisen sprintin alussa tiimi valitsee projektin backlogista sprintin aikana toteutettavat vaatimukset
 - ▶ Backlog on priorisoitu ja vaatimukset valitaan aina priorisoidun listan kärjestä
- ▶ Tiimi valitsee sprinttiin ainoastaan sen verran toteutettavaa minkä valmistumiseen se uskoo kykenevänsä sitoutumaan
- ▶ Sprintin aikana scrum-tiimi toteuttaa *itseorganisoidusti* sprinttiin valitut ohjelmiston ominaisuudet
- ▶ Sprintin aikana tiimille ei esitetä uusia vaatimuksia

Sprintti

- ▶ Scrumissa kehitystyö siis jakautuu 1-4 viikon mittaisiin iteraatioihin eli sprintteihin
 - ▶ nykyään suosituin sprintin pituus lieenee 2 viikkoa
- ▶ Sprintti on *time-boxed*, sitä ei missään olosuhteissa pidennetä
- ▶ Jokaisen sprintin alussa tiimi valitsee projektin backlogista sprintin aikana toteutettavat vaatimukset
 - ▶ Backlog on priorisoitu ja vaatimukset valitaan aina priorisoidun listan kärjestä
- ▶ Tiimi valitsee sprinttiin ainoastaan sen verran toteutettavaa minkä valmistumiseen se uskoo kykenevänsä sitoutumaan
- ▶ Sprintin aikana scrum-tiimi toteuttaa *itseorganisoidusti* sprinttiin valitut ohjelmiston ominaisuudet
- ▶ Sprintin aikana tiimille ei esitetä uusia vaatimuksia
- ▶ Sprintin lopuksi tuotteesta on oltava olemassa *toimiva versio* (potentially shippable product increment)

definition of done

- ▶ Jokaisessa sprintissä lopputuloksena toimiva, valmiiksi tehty osa ohjelmistoa
- ▶ Scrumissa on määriteltävä projektitasolla *definition of done*: mitä tarkoittaa, että jokin vaatimus on toteutettu valmiiksi

definition of done

- ▶ Jokaisessa sprintissä lopputuloksena toimiva, valmiiksi tehty osa ohjelmistoa
- ▶ Scrumissa on määriteltävä projektitasolla *definition of done*: mitä tarkoittaa, että jokin vaatimus on toteutettu valmiiksi
- ▶ Valmiiksi tehty määritellään yleensä tarkoittamaan sitä, että vaatimus on
 - ▶ *analysoitu, suunniteltu, ohjelmoitu, testattu, testaus automatisoitu, dokumentoitu, integroitu muuhun ohjelmistoon ja viety tuotantoympäristöön*

definition of done

- ▶ Jokaisessa sprintissä lopputuloksena toimiva, valmiiksi tehty osa ohjelmistoa
- ▶ Scrumissa on määriteltävä projektitasolla *definition of done*: mitä tarkoittaa, että jokin vaatimus on toteutettu valmiiksi
- ▶ Valmiiksi tehty määritellään yleensä tarkoittamaan sitä, että vaatimus on
 - ▶ *analysoitu, suunniteltu, ohjelmoitu, testattu, testaus automatisoitu, dokumentoitu, integroitu muuhun ohjelmistoon ja viety tuotantoympäristöön*
- ▶ Jos Sprintissä on toteutettu joitain vaatimuksia puutteellisesti DoD:in kannalta, niitä ei tule raportoida valmiina, Scrum master siirtää ne toteutettavaksi seuraavaan sprinttiin
- ▶ Jos sprintin aikana osoittautuu että tiimi ei ehdi toteuttamaan kaikkea joihin se sitoutui, ei ole hyväksyttävää tinkiä laadusta, vaan osa vaatimuksista jätetään seuraavaan sprinttiin

Sprint planning

- ▶ Ennen jokaista sprinttiä järjestetään sprintin *suunnittelukokous* eli sprint planning
 - ▶ kouksella kaksi tavoitetta ta Scrumin sanoin *aihetta* (engl. topic)

Sprint planning

- ▶ Ennen jokaista sprinttiä järjestetään sprintin *suunnittelukokous* eli sprint planning
 - ▶ kouksella kaksi tavoitetta ta Scrumin sanoin *aihetta* (engl. topic)
- ▶ Ensimmäisen aihe on selvittää *mitä* sprintin aikana tehdään
 - ▶ Product owner esittelee product backlogin kärjessä olevat vaatimukset
 - ▶ Tiimin on tarkoitus olla riittävällä tasolla selvillä siitä, mitä vaatimuksilla tarkoitetaan
 - ▶ Tiimi arvioi kuinka monta backlogin eli tehtävälistan vaatimuksista se kykenee sprintin aikana toteuttamaan (Definition of donen määrittelemällä laadulla)

Sprint planning

- ▶ Ennen jokaista sprinttiä järjestetään sprintin *suunnittelukokous* eli sprint planning
 - ▶ kouksella kaksi tavoitetta ta Scrumin sanoin *aiheetta* (engl. topic)
- ▶ Ensimmäisen aihe on selvittää *mitä* sprintin aikana tehdään
 - ▶ Product owner esittelee product backlogin kärjessä olevat vaatimukset
 - ▶ Tiimin on tarkoitus olla riittävällä tasolla selvillä siitä, mitä vaatimuksilla tarkoitetaan
 - ▶ Tiimi arvioi kuinka monta backlogin eli tehtävälistan vaatimuksista se kykenee sprintin aikana toteuttamaan (Definition of donen määrittelemällä laadulla)
- ▶ Sprintin aikana toteutettavien vaatimusten lisäksi asetetaan *sprintin tavoite* (sprint goal)
 - ▶ yksittäisiä vaatimuksia geneerisempi ilmaus siitä mitä tulevassa sprintissä on tarkoitus tehdä

Sprint planning

- ▶ Suunnittelukokouksen toisena aiheena on selvittää *miten* sprintin tavoitteet saavutetaan

Sprint planning

- ▶ Suunnittelukokouksen toisena aiheena on selvittää *miten* sprintin tavoitteet saavutetaan
- ▶ Tämä yleensä edellyttää että tiimi suunnittelee toteutettavaksi valitut vaatimukset tarvittavalla tasolla
- ▶ Aikaansaannoksena on usein lista *tehtävistä* (task), jotka sprintin aikana on toteutettava, jotta sprinttiin valitut vaatimukset saadaan toteutettua

Sprint planning

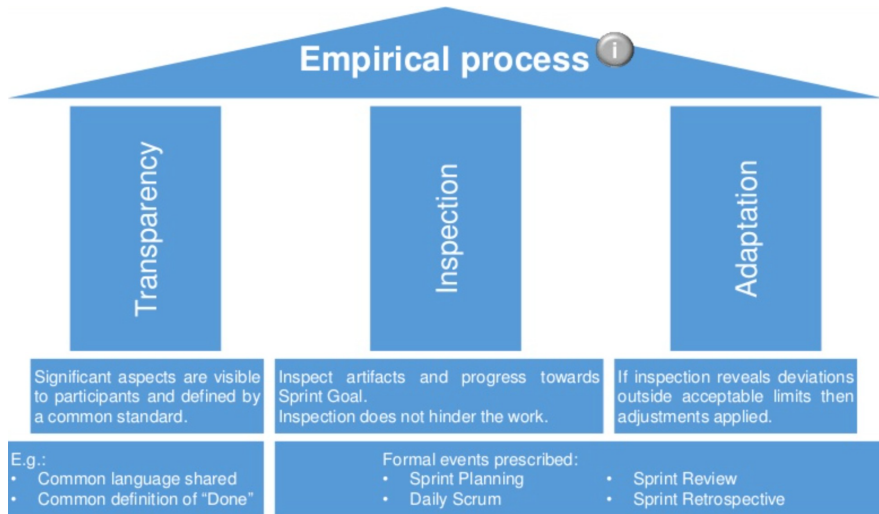
- ▶ Suunnittelukokouksen toisena aiheena on selvittää *miten* sprintin tavoitteet saavutetaan
- ▶ Tämä yleensä edellyttää että tiimi suunnittelee toteutettavaksi valitut vaatimukset tarvittavalla tasolla
- ▶ Aikaansaannoksena on usein lista *tehtävistä* (task), jotka sprintin aikana on toteutettava, jotta sprinttiin valitut vaatimukset saadaan toteutettua
- ▶ Suunnittelun aikana identifioidut tehtävät kirjataan *sprintin backlogiin* eli sprintin tehtävälistaan

Sprint planning

- ▶ Suunnittelukokouksen toisena aiheena on selvittää *miten* sprintin tavoitteet saavutetaan
- ▶ Tämä yleensä edellyttää että tiimi suunnittelee toteutettavaksi valitut vaatimukset tarvittavalla tasolla
- ▶ Aikaansaannoksena on usein lista *tehtävistä* (task), jotka sprintin aikana on toteutettava, jotta sprinttiin valitut vaatimukset saadaan toteutettua
- ▶ Suunnittelun aikana identifioidut tehtävät kirjataan *sprintin backlogiin* eli sprintin tehtävälistaan

Palaamme sprintin suunnitteluun tarkemmin ja konkreettisten esimerkkien kanssa ensi viikolla

Transparency - inspect -adapt



Fails...