

Ohjelmistoprojekti 2

Tervetuloa Ohjelmistoprojekti 2 kurssille!

Kurssista

- Kurssilla toteutetaan ryhmätyönä ohjelmistoprojekti ryhmäläisten muodostamasta aiheesta
- Projekti etenee kolmen viikon iteraatioissa kahden opetusperiodin ajan
- Jokaisen iteraation aluksi ryhmäläiset suunnittelevat iteraation aikana ohjelmistoon toteutettavat toiminallisuudet
- Iteraation aikana ryhmäläiset toteuttavat suunniteltuja toiminallisuuksia itsenäisesti sekä opettajan ohjauksessa
- Iteraation päätteeksi ryhmäläiset esittelevät opettajalle projektin edistymistä ja saavat palautetta
- Opetusta järjestetään viikottain. Lisätietoa opetustilaisuuksista löytyy kurssisivulta

Arviointi

- Kurssin arviointi perustuu seuraaviin tekijöihin:
 - Ryhmätyöskentelyn sujuvuuteen
 - Ryhmän lopullisen tuloksen tarkoituksenmukaiseen toimivuuteen
 - Kurssin lopuksi yksilötyönä tehtävään esseeseen, jossa arvioidaan omaa ja ryhmän toimintaa
- Arvioinnin tukena käytetään kurssin aikana tehtävää itseis- ja vertaisarviointia

Ketterä ohjelmistokehitys ja Scrum

Ketterä ohjelmistokehitys

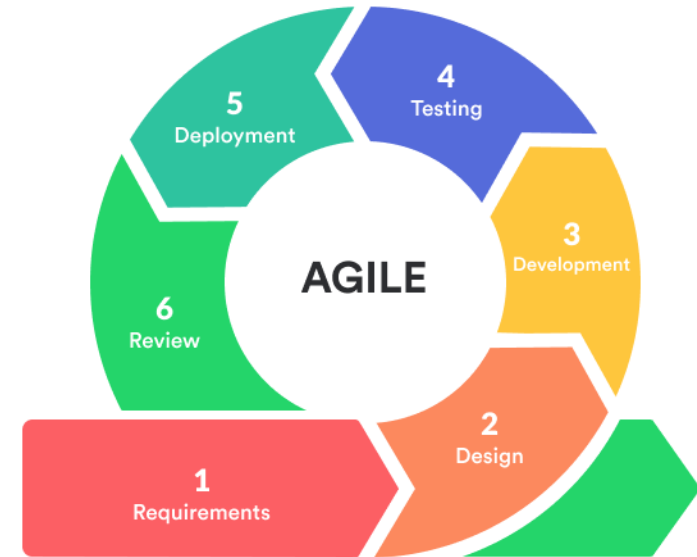
- Ohjelmistoprojekteissa toteutaan ohjelmistoja, jotka perustuvat johonkin tarpeeseen, jonka määrittelee ohjelmistoprojektin asiakas
- Ohjelmistokehittäjien tehtävä on muodostaa asiakkaan tarpeista toteuttamiskelpoisia teknisiä vaatimuksia ja toteuttaa ne sopivilla teknologioilla
- Tarpeiden selvittäminen vaatii jatkuvaa suoraa viestintää asiakkaan kanssa. Tarpeilla on myös tapana muuttua ja tarkentua ohjelmistoprojektin edetessä
- Ohjelmistotuotannossa sovelletaan nykyisin laajalti nk. *ketteriä menetelmiä*, jotka korostavat mm. suoraa viestintää sidosryhmien kanssa ja nopeaa muutokseen reagointia
- Ketterät menetelmät ovat vaihtoehto perinteisille vaiheellisille ohjelmistotuotantoprosesseille, kuten vesiputousmalli

Ketterät menetelmät

"Kevyesti, joustavasti, notkeasti liikkuva, sukkela, vikkelä."

– Sanakirjan määritelmä sanalle *"ketterä"*

- Ketterien menetelmille tyypillistä on, että ohjelmistoa kehitetään lyhyissä iteraatioissa
- Jokaisen iteraation aikana suoritetaan kaikki ohjelmiston elinkaaren vaiheet vaatimusmäärittelystä tuotantoonvientiin



Scrum

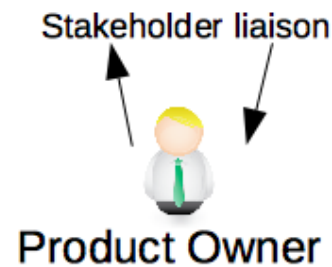
- Kurssin ohjelmistoprojektin projektinhallinnassa noudatetaan ketteriä menetelmiä soveltavaa *Scrum*-viitekehystä
- Scrum määrittelee ohjelmistokehitykselle iteratiivisen prosessin, joka etenee tyypillisesti 1-4 viikon iteraatioissa, joita kutsutaan *sprinteiksi*
- Sprintin aikana ohjelmistoon tuotetaan inkrementaalisesti uutta julkaisukelpoista toiminnallisuutta
- Kurssilla sprintin pituus on kolme viikkoa

Scrumin roolit

- Ohjelmistoprojektista vastaa *Scrum-tiimi*, jossa on kolme erilaista roolia
- Ohjelmiston toteutuksesta vastaa *kehittäjätiimi*, joka koostuu 3-9 ohjelmistokehittäjästä
 - Sprintin aikana kehittäjätiimi toteuttaa itseorganisoidusti sprinttiin valitut ohjelmiston toiminnallisuudet
- *Scrum master* on usein kehittäjätiimin jäsen, joka toimii sen apuna ohjaten mm. prosessin noudattamisessa ja parantamisessa
- *Tuoteomistaja* (product owner) määrittelee ja priorisoi kehittäjätiimin työtä
 - Hallinnoi projektin *product backlogia*, joka sisältää priorisoidussa järjestyksessä projektissa toteutettavalle ohjelmistolle asetetut vaatimukset

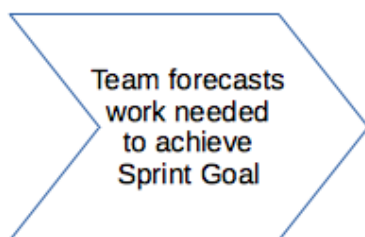
Scrumin tapahtumat

- Scrumissa käytetään ennaltasovittuja tapahtumia, jotka toistuvat jokaisessa sprintissä
- Sprintti alkaa *sprintin suunnittelulla* (sprint planning), jonka aikana päätetään mitä toiminallisuutta sprintin aikana toteutetaan
- Sprintin aikana järjestetään lyhyitä kehittäjätiimin sisäisiä *päiväpalavereja* (daily scrum), joissa jokainen tiimin jäsen kertoo vuorallaan, miten kehitystyö etenee
- Sprintti päättyy *sprinttikatselmukseen* (sprint review), jonka aikana kehittäjätiimi esittelee sprintin aikana toteutetut toiminallisuudet kaikille kehitettävästä tuotteesta kiinnostuneille sidosryhmille
- Sprintin päätteeksi järjestetään *retrospektiivi*, jossa kehittäjätiimi tarkastelee Scrum masterin johdolla omaa työskentelyprosessiaan ja pyrkivät kehittämään sitä



1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	

Product Backlog

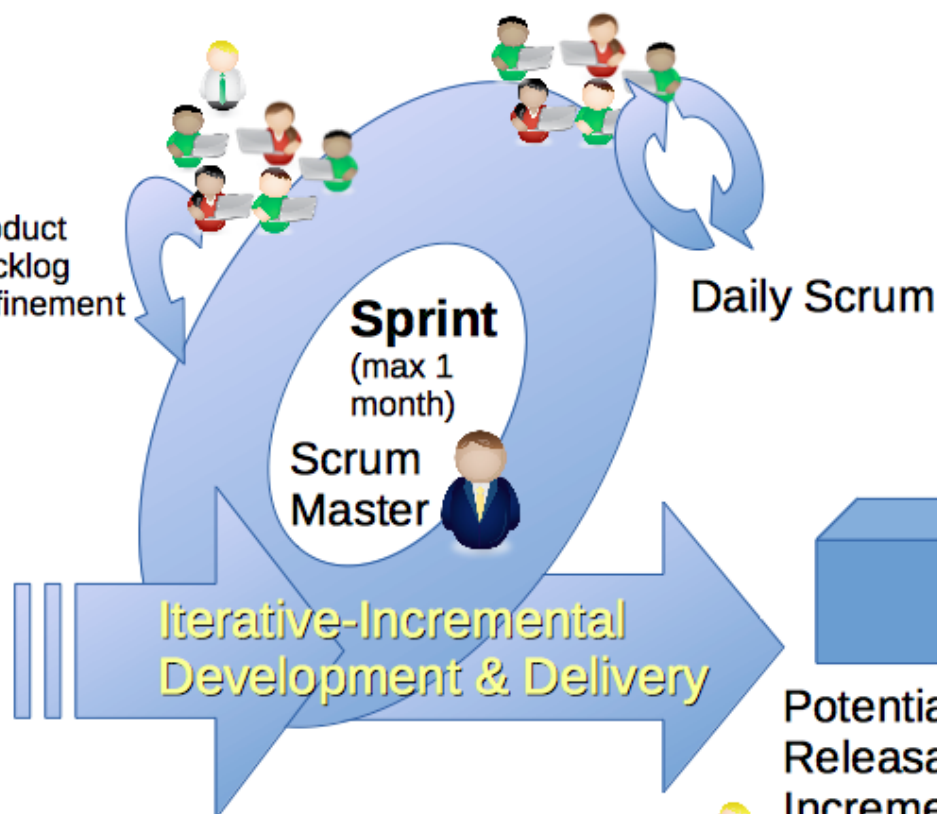


Sprint Planning

Topic 1: forecast PBI's
Topic 2: plan work (e.g. tasks)



Product Backlog Refinement



Potentially Releasable Increment

Sprint Review

Sprint Retrospective

Ryhmäytyminen ja aiheen ideointi

Aloitetaan työskentely ryhmissä

1. Jakaudutaan noin viiden hengen ryhmiin
2. Tehkää ryhmän sisällä lyhyt esittelykierros. Jokainen voi kertoa vuorollaan esimerkiksi:
 - Minkälainen tausta on opinnoissa tai mahdollisesti työelämässä
 - Mitkä ovat omat kiinnostuksen kohteet ohjelmistokehityksessä?
 - Mitkä ovat omat vahvuusalueet ohjelmointikielissä, tai muissa toteutusteknologioissa?
 - Mitä odotuksia on kurssin suhteen?
3. Alkakaavaa ideoimaan yhdessä ohjelmistoprojektin aihetta