

Ohjelmistokehitys

Ketteryys / Scrum

Ohjelmistokehitys

Ketteryys

- ☐ Vuoteen 2000 mennessä julkaistu kymmenkunta iteratiivista ohjelmistokehitysmenetelmää, eniten huomiota saanut Kent Beckin kehittämä XP (eXtreme Programming)
- ☐ Vuonna 2001 joukko kevyiden menetelmien kehittäjiä, kannattajia ja konsultteja kokontui hiihtokeskukseen Utahissa keskustelemaan käyttämistään menetelmistä
- ☐ Varsinkin Yhdysvalloissa sovellataan yleisesti suunnitelmaohjattuja(plan-driven), raskaita kehitysprosesseja
- ☐ Vastapainoksi kehittämiään kevyitä menetelmiä päättivät kokouksen osallistuja kutsua ketteriksi (agile) menetelmiksi
- ☐ Kahden päivän kokouksen aikana sovittiin ketteristä menetelmiä yhdistävästä periaatteista ja perustettiin ketterien menetelmien edistämistä ajava Agile Alliance- järjestö
- ☐ ”Peruskijana” julkaistiin Agile Manifesto; Manifestin ydin on sen arvoissa
- ☐ Voidaan tulkita niin, että prosessi, työkalut, dokumentaatio jne. ovat turhia; kuitenkin manifesto toisaalta tunnustaa nämä ohjelmistokehitystyön kannalta arvokkaiksi
- ☐ Todetaan kuitenkin että kaikkein tärkeintä ovat tyytyväinen asiakas ja toimiva ohjelma

Ohjelmistokehitys

Scrum

- ☐ Yleisemmin käyttöönotettu ketterä menetelmä viime vuosina
- ☐ Jopa niin että Scrum sanasta on tullut lähes sanan ketterä synonyymi
- ☐ Scrum on alkujaan peräisin Japanista
- ☐ Ensimmäinen kuvaus tehtiin Harvard Business Review –lehdessä jo 1986; esitettiin nykyisen Scrum-menetelmän keskeisimmät periaatteet
- ☐ 1990-luvun alkupuolella Jeff Sutherland yhdessä Ken Schwaberin kanssa kehittivät näitä periaatteita noudattavaa ohjelmistokehitysmenetelmää Sutherlandin ohjelmistokehitystiimille
- ☐ Scrum syntyi käytännössä 10 vuotta ennen kuin ketteryydestä oli tullut muotitermi

Ohjelmistokehitys

Scrum (Taustaa)

- ☐ Uusien projektinhallintamallien lanseeraus ei yleensä ole innostanut ohjelmistosuunnittelijoita
- ☐ Scrum on selkeä poikkeus, sillä on pääsääntöisesti saanut lämpimän vastaanoton
- ☐ Vaikka Scrumia yleensä pidetään ketteränä menetelmänä, todellisuudessa ei välttämättä ole kovin ketterä ja voi olla jopa ketteryyden ”jäykistäjä”
- ☐ Scrumin suosiota kuvaa hyvin, että muutaman päivän kursseja järjestetään sadoittain eri puolilla maailmaa
- ☐ Scrumin merkittävä etu on yksinkertaisuus: perusperiaatteet on helppo selittää nopeasti ja Scrum tarjoaa ratkaisuja hyvinkin moniin ohjelmistotuotannon ongelmiin
- ☐ Scrum ottaa kantaa vain osaan ohjelmiston elinkaareen liittyvistä tehtävistä, eikä ota kantaa kehitysmenetelmiin ja työkaluihin
- ☐ Kärjistäen voidaan sanoa että Scrum ei ole projektinhallintamenelmä vaan lähinnä projektin toteutusvaiheeseen tarkoitettu tapa organisoida projektin iteraatiot.
- ☐ Esim. vaatimusten hallinta on delegoitu projektin omistajan tehtäväksi
- ☐ Scrum ei yksinään riitä vaan on yhdistettävä kaikkeen muuhun projektinhallinnassa tarvittavaan välineistöön

Ohjelmistokehitys

Scrum-prosessi

- ☐ Scrummissa on vain kolme roolia: tuotteen omistaja (product owner), Scrum-mestari (Scrum Master) ja tiimi
- ☐ Perinteisillä nimityksillä tuotteen omistajaa vastaa tuotepäällikkö (eli tuotteen omistajaa), Scrum-mestari projektipäällikkö ja tiimi projektiryhmää
- ☐ Tuotteen omistaja (product owner) vastaa projektin taloudellisesta tuloksesta
- ☐ Toimii projektin rajapintana kaikkiin sidosryhmiin, joilta kerää järjestelmän vaatimukset työlistaksi ja ylläpitää tuotteen työlistaa (product backlog) prioritetijärjestyksessä
- ☐ Kuhunkin työlista-alkioon liittyy alustava aika-arvio sekä arvio sen liiketoiminta-arvosta
- ☐ Työlistan alkiot (item) voivat olla mitä tahansa esim:
 - tuotteen ominaisuuksia
 - käyttötapauksia (use case)
 - käyttäjätarinoita (user story)
 - vaatimuksia (requirements)
 - virheraportteja
 - dokumentaation kehittämistä

Ohjelmistokehitys

Scrum-mestari

- ❑ Scrum-mestarin (Scrum master) voidaan sanoa olevan ”projektipäällikkö ilman valtaa”
- ❑ Vastaa että Scrum-prosessia noudatetaan ja toimii tiimin sekä tuotteenomistajan valmentajana
- ❑ Vastaa pyrähdysten tuloksesta ja varmistaa, että tehtävää ei merkitä valmiiksi ennen kuin kaikki valmistumiseen määritellyt ehdot (definition-of-done) on täytetty
 - esim. koodi kirjoitettu, testitapaukset olemassa sekä dokumentaatio päivitetty
- ❑ Scrum-mestarin tehtävänä on myös vastata tiimin hyvinvoinnista ja tiimin työtä haittaavien esteiden(impediment) poistamisesta
- ❑ Tärkein valtuutensa on sopimattoman jäsenen poistaminen tiimistä

Ohjelmistokehitys

Scrum (Tiimi)

- ☐ Tiimin optimaalinen koko on noin 7 kokopäiväistä henkilöä, mielellään taustoiltaan erilaisia esim. käyttöliittymäspecialisteja, testaajia, ohjelmoijia ns. cross functional team
- ☐ Tiimi on itseorganisoituva eli varsinaista projektipäällikkö ei ole
- ☐ Scrum-mestari usein saa projektipäällikön roolin
- ☐ Tiimi saa päättää itse omista työskentelykäytännöistä
- ☐ Tiimi vastaa pyrähdyksen työlistan paloittelusta tehtäviksi ja jakaa ne keskenään
- ☐ Tiimin olisi hyvä olla samassa työtilassa johon voidaan sijoittaa tehtävätaulu
- ☐ Tehtävät ja niiden tilat kaikkien nähtävissä (ei aloitettu, kesken, valmis ja hyväksytty)
- ☐ Tiimi ylläpitää taulua yhdessä Scrum-mestarin kanssa
- ☐ Päivittäiset Scrum-palaveri (daily scrum), voidaan pitää seisaallaan taulun ääressä

Ohjelmistokehitys

Scrum Eteneminen

- ☐ Scrummissa projekti etenee pyrähdysinä (sprint), pituus on 30 kalenteripäivää
- ☐ Jokainen pyrähdys alkaa suunnittelukokouksella (sprint planning meeting), joka kestää koko päivän
- ☐ Kokoukseen osallistuvat tuotteen omistaja, Scrum-mestari ja tiimi
- ☐ Kokouksen aamupäivä käytetään siihen että tuotteen omistaja esittelee tuotteen työlistan (product backlog)
- ☐ Iltapäivällä käydään keskustelut mitkä tuotteen tehtävälistan alkiot(product backlog item) otetaan mukaan seuraavaan pyrähdykseen
- ☐ Tiimi päättää mihin se voi sitoutua
- ☐ Tehtävälistan alkiot pilkotaan tehtäviin (task) joiden kesto on tyypillisesti 4-6 tuntia
- ☐ Yksi tapa on käyttää niin sanottua suunnittelupokeria (plannin poker) jossa jokainen tiimin jäsen antaa aika-arvion
- ☐ Aikayksiköllä ei ole väliä; muutaman pyrähdyksen jälkeen tiimi tietää kuinka monta yksikköä voidaan ahtaa yhteen pyrähdykseen -> Tätä sanotaan tiimin nopeudeksi (Velocity)
- ☐ Pyrähdyksen aikana suunnittelukokouksessa sovitut asiat tehdään
- ☐ Tilannetta seurataan käyttäen pyrähdyksen edistämiskäyrää (sprint burndown chart)

Ohjelmistokehitys

Scrum Eteneminen

- ☐ Pyrähdyn aikana pidetään päivittäinen 15minuutin Scrum- kokous (daily scrum)
- ☐ Jokainen tiimin jäsen vastaa kolmeen kysymykseen:
 - Mitä olet tehnyt edellisen kokouksen jälkeen?
 - Mitä aiot tehdä seuraavaksi?
 - Onko tiedossasi esteitä, jotka hidastavat työtäsi?
- ☐ Kokoukseen osallistuvat Scrum-mestari ja tiimi
- ☐ Myös tuotteen omistaja voi osallistua, mutta hänellä ei ole puheoikeutta
- ☐ Pyrähdyn päätyttyä pidetään pyrähdyn katselmointikokous, johon osallistuvat tiimi, Scrum-mestari, tuotteen omistaja ja mahdollisesti sidosryhmän edustajia (asiakkaita)
- ☐ Tiimi demostroii pyrähdyn tulokset ja kerää palautteen sidosryhmien edustajilta
- ☐ Jos käytetään lyhyitä, esim. 2viikon pyrähdysiä voidaan jatkaa suoraan seuraavan pyrähdyn suunnittelukokoukseen
- ☐ Pyrähdyn arviointipalaveriin (retrospective) osallistuvat tiimi, Scrum-mestari ja tuotteen omistaja
- ☐ Arvioidaan pyrähdyn onnistumista ja mietitään toimintatapojen kehittämistä

Ohjelmistokehitys

Scrum-prosessi

