

Ryhmädynamiikka ja ketterät menetelmät

Ohjelmistoprojektien johtaminen ja ryhmädynamiikka

13.2.2018

Fabian Fagerholm



Scrum-tiimin päiväkokous (Flickr, Creative Commons)

Johdanto

Fabian Fagerholm
fabian.fagerholm@helsinki.fi

- Tutkijatohtori, Empirical Software Engineering –tutkimusryhmä, tietojenkäsittelytieteen laitos (HY)
- Yliopistonlehtori, ohjelmistotuotannon laitos, Blekinge tekniska högskola (Ruotsi)
- Kehittäjäkokemus (*developer experience*)
 - Kuinka ohjelmistokehittäjät kokevat suorituskykyä virtaviivaisessa ja ketterässä (Lean-Agile) ohjelmistokehityksessä
 - Virtaviivaisen ja ketterän ohjelmistokehityksen arvot ohjelmistokehittäjien keskuudessa
 - Kehittäjäkokemukseen vaikuttaminen Open Source projekteissa mentoroinnin kautta
 - Ohjelmistokehittäjien onnellisuus
- Muita tutkimusaiheita
 - Ohjelmistotiimit, tiimien suorituskyky, ei-tekninen osaaminen
 - Jatkuva eksperimentointi (continuous experimentation) ohjelmistointensiivisessä tuote- ja palvelukehityksessä
 - Ohjelmistotuotannon opetus
 - (Ohjelmistoprojektien/tuotteiden/prosessien mittaaminen)

Fagerholm, F., Ikonen, M., Kettunen, P., Münch, J., Roto, V., Abrahamsson, P. (2015). Performance Alignment Work: How software developers experience the continuous adaptation of team performance in Lean and Agile environments. *Information and Software Technology*, vol. 64, pp. 132-147.

Fagerholm, F., Pagels, M. (2014). Examining the Structure of Lean and Agile Values Among Software Developers. *XP 2014. LNBP*, vol. 179, pp. 218-233.

Fagerholm, F., Sanchez Guinea, A., Borenstein, J., Münch, J. (2014). Onboarding in Open Source Software Projects. *IEEE Software*, vol. 31(6), pp. 54-61.

Fagerholm, F., Sanchez Guinea, A., Mäenpää, H., Münch, J. (2017). The RIGHT Model for Continuous Experimentation. *The Journal of Systems and Software*, vol. 123, pp. 295-305.

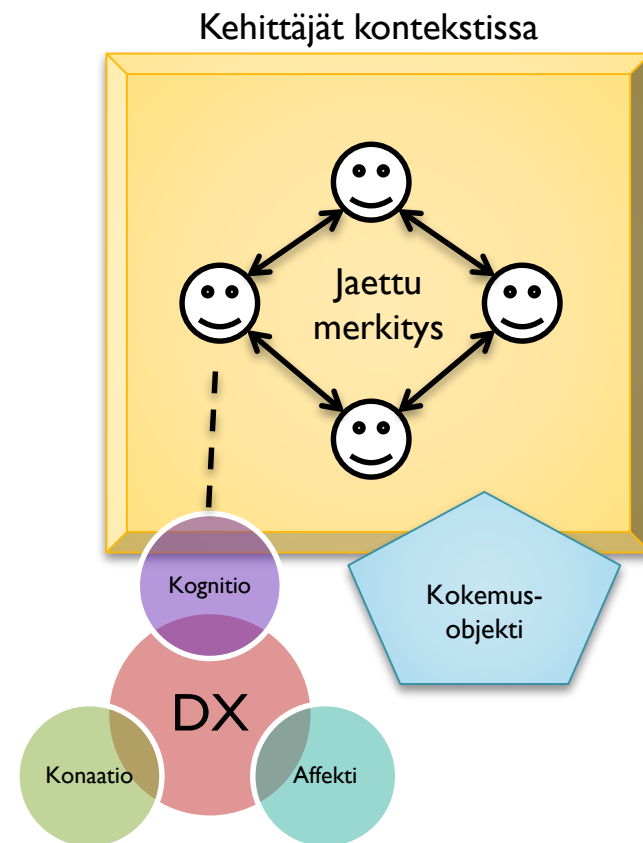
Graziotin, D., Fagerholm, F., Wang, X., Abrahamsson, P. (2017). On the Unhappiness of Software Developers. *Proc. 21st International Conference on Evaluation and Assessment in Software Engineering*. ACM pp. 324-333.

Lisää: <http://orcid.org/0000-0002-7298-3021>

Kehittäjäkokemus

(developer experience, DX)

- Tarkastelee kehittäjäksilöä sosioteknisessä ympäristössä
 - Vaikutteita käyttäjäkokemuksesta mutta tarkastelee *kehittämisen kontekstia* käytön kontekstin sijaan
 - Nojaa vahvasti sosiaalipsykologiaan: yksilö ryhmässä
- Kysyy, kuinka jotain valittua osaa ohjelmistokehityksen työympäristöstä (“kokemusobjekti”) voidaan ymmärtää ja/tai parantaa ohjelmistokehittäjien näkökulmasta
 - Esim. Organisaatio, kehitysmenetelmä, prosessi, alusta, ympäristö, työkalu, ...
 - Laajempi ymmärrys tai parannus voi tavoitella sekä suorituskykyä että henkilökohtaista hyvinvointia
 - Tutkimusmenetelmät riippuvat tavoitteista; (kontrolloiduista) kokeista kvalitatiivisiin tutkimuksiin
 - Paljon avoimia kysymyksiä, myös soveltuvista menetelmistä
- Tässä: yksilön ja ryhmän tarkastelua ketterässä ohjelmistokehityksessä



Kehittäjäkokemuksen avainkäsitteitä.
Muokattu: Fagerholm (2012, 2015).

Ketterä ohjelmistokehitys

- Syntyi 2001 17 ohjelmistoammattilaisen tapaamisessa
- Vaihtoehto dokumentaatiovetoisille, raskaille ohjelmistotuotantoprosesseille
- Keskeinen perustelu oli nopeampi reaktiokyky muuttuvassa markkina-tilanteessa (jossa asiakkaan vaatimukset voivat muuttua projektin aikana)
- Asiakas tiiviisti mukana, aktiivinen kommunikointi ohjelmistokehittäjien kanssa
- Ohjelmistokehittäjän rooli korostuu



Ketterä manifesti:

Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä.

Kokemuksemme perusteella arvostamme:

Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja

Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota

Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja

Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa

Jälkimmäisilläkin asioilla on arvoa, mutta arvostamme ensiksi mainittuja enemmän.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Ihmiskeskeisyys ketterässä ja virtaviivaisessa ohjelmistokehityksessä

Ketterä manifesti:

Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä.

Kokemuksemme perusteella arvostamme:

Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja

Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota

Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja

Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa

Jälkimmäisilläkin asioilla on arvoa, mutta arvostamme ensiksi mainittuja enemmän.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

© 2001, the above authors. This declaration may be freely copied in any form, but only in its entirety through this notice. Suomenkielinen käännös Lasse Koskela.

Virtaviivaisen ohjelmistokehityksen filosofia:

Kehitysprosessia on jatkuvasti parannettava

Eliminoi jätteen

Vahvista oppimista

Päätä mahdollisimman myöhään

Toimita mahdollisimman nopeasti

Valtuuta tiimi

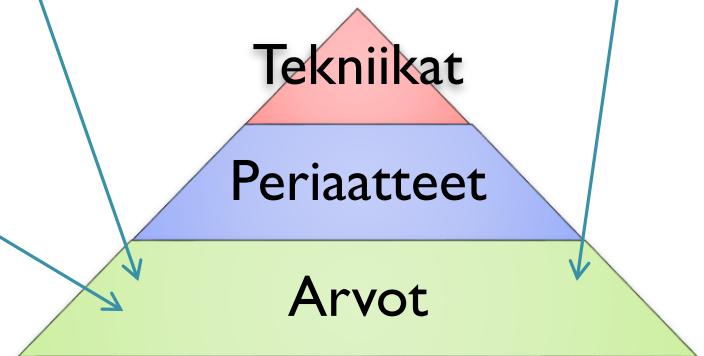
Rakenna laatu sisään

Näe kokonaisuus

Tekniikat

Periaatteet

Arvot



Ihmiskeskeisyys ketterässä ja virtaviivaisessa ohjelmistokehityksessä

- Ketterä ohjelmistokehitys keskittyy lähinnä pienryhmään
 - Itseohjautuva ryhmä, joka itsetarkkailun kautta jatkuvasti parantaa toimintaansa
 - Nojaa erityisesti hyvin toimivaan ryhmädynamiikkaan – ei vahvasti ohjaavaan prosessiin
- Viimeaikaisen tutkimuksen perusteella ketterän ohjelmistokehityksen skaalaaminen ison organisaation tasolle on haasteellista
 - Voidaanko arvopohjaista menetelmää skaalata vai onko pakko lisätä ohjaavia rakenteita?



Nk. Deming laatuympyrä tai PDCA-sykli.
(Keskellä olevat hahmot eivät kuulu alkuperäiseen. ☺)

Pohdinta: Mitä vahvuuksia ja heikkouksia seuraa ketterän ohjelmistokehityksen ihmiskeskeisyydestä? Miksi sitä on vaikea skaalata ison organisaation tasolle?
Huom: Pyrkikää käyttämään kurssilla opittuja teorioita ja käsitteitä!

Ihmiskeskeisyys ketterässä ja virtaviivaisessa ohjelmistokehityksessä

Ketterä manifesti:

Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä.

Kokemuksemme perusteella arvostamme:

Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja

Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota

Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja

Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa

Jälkimmäisilläkin asioilla on arvoa, mutta arvostamme ensiksi mainittuja enemmän.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

© 2001, the above authors. This declaration may be freely copied in any form, but only in its entirety through this notice. Suomenkielinen käännös Lasse Koskela.

Virtaviivaisen ohjelmistokehityksen filosofia:

Kehitysprosessia on jatkuvasti parannettava

Eliminoidi jätteen

Vahvista oppimista

Päätä mahdollisimman myöhään

Toimita mahdollisimman nopeasti

Valtuuta tiimi

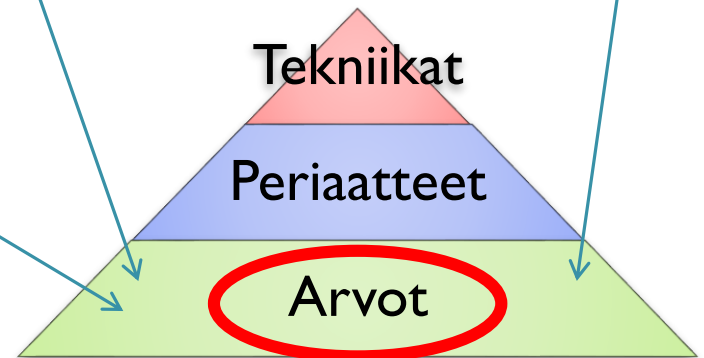
Rakenna laatu sisään

Näe kokonaisuus

Tekniikat

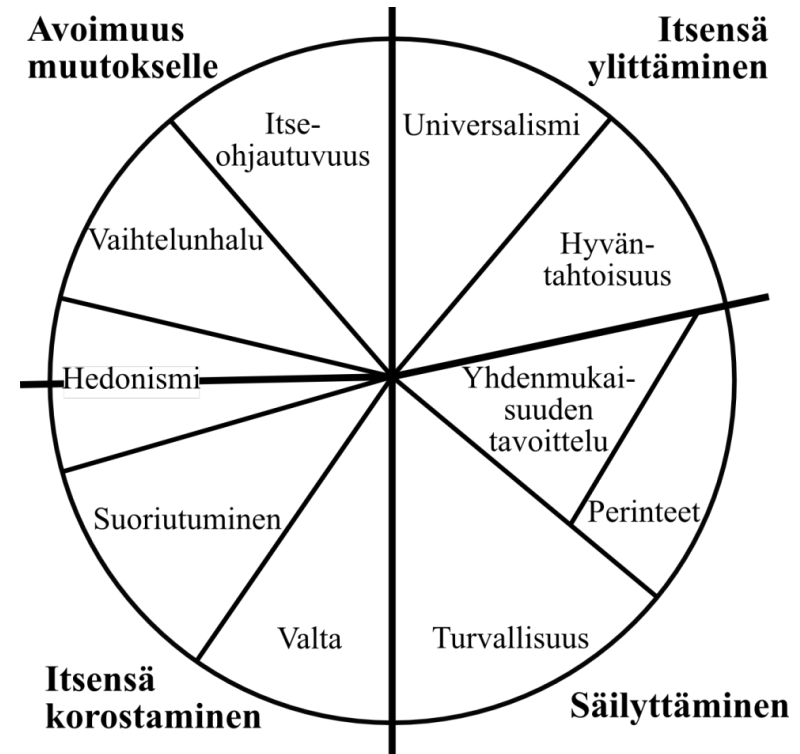
Periaatteet

Arvot



Arvot

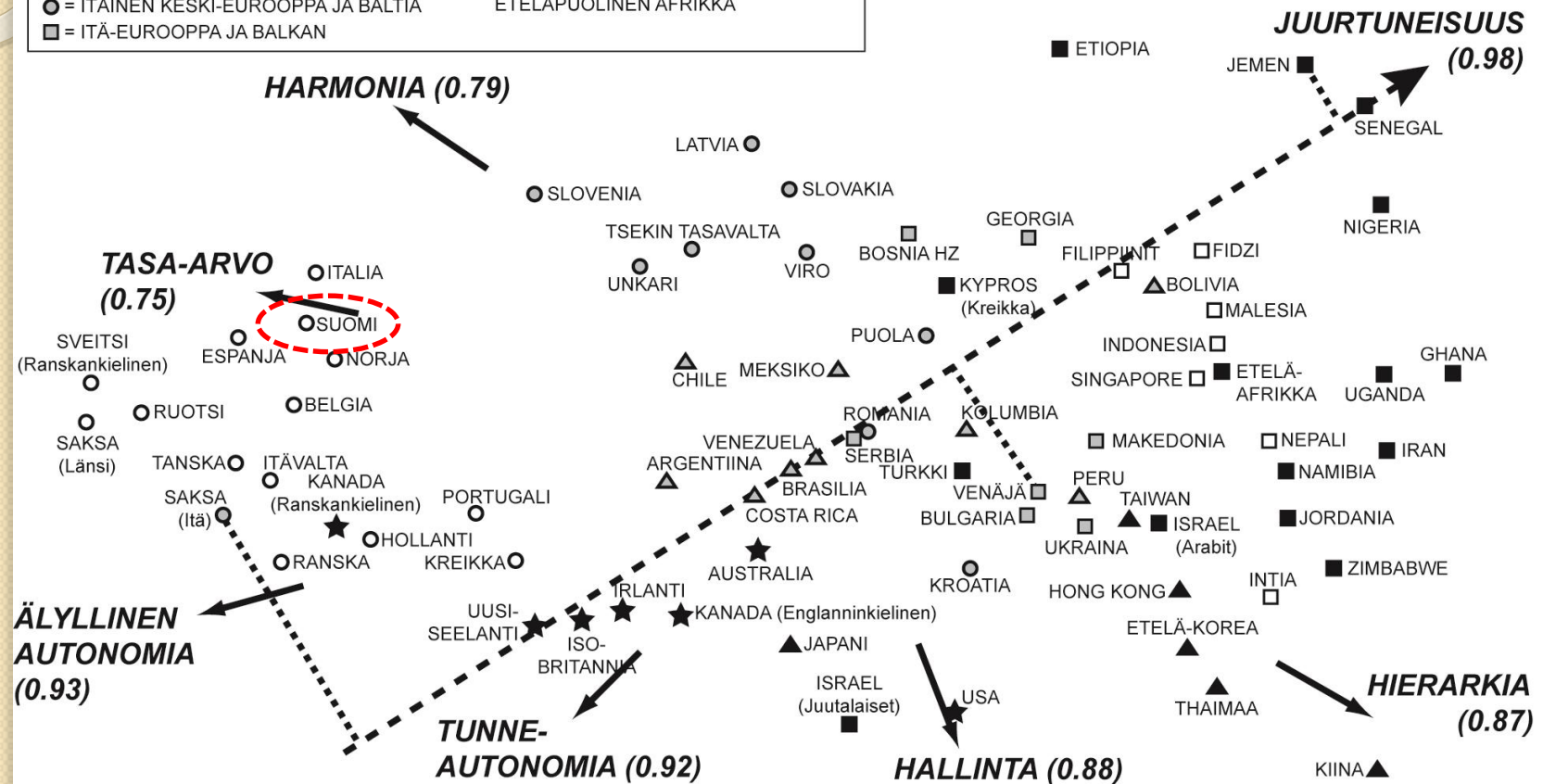
- Arvot ovat yksilölle tai ryhmälle ominaisia käsityksiä toivottavasta
- Arvoja opitaan ympäristöstä, ne ovat yleisiä ja verraten pysyviä, tavoitteita koskevia valintataipumuksia



Schwartzin arvoteoria
(mm. Bilsky & Schwartz 2011).

Arvot: Schwartz

- | | |
|--|---|
| ○ = LÄNSI-EUROOPPA | □ = ETELÄ- JA KAAKKOIS-AASIA |
| ★ = ENGLANTIA PUHUVA ALUE | ▲ = KONFUSETLAISUUDEN VAIKUTUSALUE |
| △ = LATINALAINEN AMERIKKA | ■ = ISLAMILAINEN LÄHI-ITÄ JA SAHARAN ETELÄPUOLINEN AAFRIKKA |
| ○ (pieni) = ITÄINEN KESKI-EUROOPPA JA BALTIA | |
| □ (pieni) = ITÄ-EUROOPPA JA BALKAN | |



Esim. Schwartz, S. H. and Bilsky, W. (1990). Toward a theory of the universal content and structure of values: Extensions and cross cultural replications. *Journal of Personality and Social Psychology*, 58, 878-891. Kuva: Puuhiniemi (2012).

Ketterän ohjelmistokehityksen arvojärjestelmä

1. Valuing a Narrow Work Focus
2. Valuing Flexibility in Task Execution and Leadership
3. Valuing Planning and Preparation
4. Valuing Adherence to the Process
5. Valuing Discipline
6. Valuing Reliance on People
7. Valuing the Freedom to Organise
8. Valuing a Sense of Purpose
9. Valuing Predictability and Justification
10. Valuing Collaboration
11. Valuing Broad Stakeholder Involvement

Fagerholm, F., Pagels, M. (2014). Examining the structure of Lean and Agile values among software developers. XP 2014. Agile processes in Software Engineering and Extreme Programming.

Normit

- Normit ovat arvoja spesifisempiä ja konkreettisempia
 - Arvot ~ mitä on yleisesti toivottavaa (hyvää/huonoa)
 - Normit ~ käyttäytymiseen kohdistuvia odotuksia tietyissä tilanteissa
- Pohdinta: Miten ketterä ohjelmistokehitys voisi vaikuttaa ryhmän arvoihin ja normeihin?
- Vinkki:
 - Ketterälle kehittäjälle on tärkeää ... (arvo)
 - Ketterän kehittäjän (ei) pitäisi ... kun ... (normi)

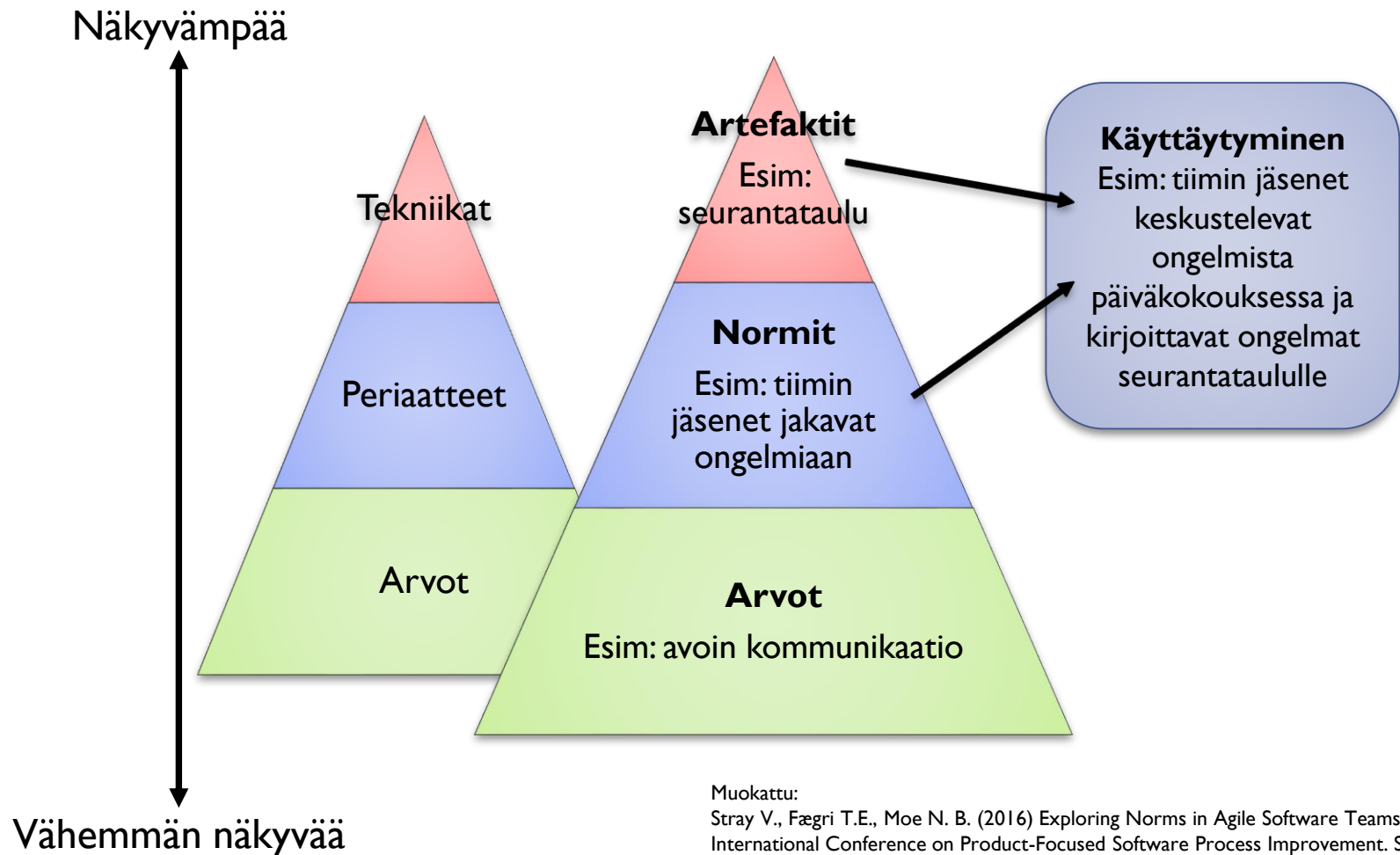
Normit

- Injuktiiviset normit
 - Konkreettiset odotukset tai kiellot
 - ”Tuoteomistaja (product owner) ei saa osallistua retrospektiiveihin.”
 - ”Jokaisen pitää tulla ajoissa kokouksiin.”
- Deskriptiiviset normit
 - Kuvauksia siitä, miten suurin osa toimii; implisiittisiä
 - Tavoite: Jokainen päivittää edistymiskäyrää (burndown chart) itse.
 - Havainto: Tiimin jäsenet päivittävät edistymiskäyrää harvoin.
 - Deskriptiivinen normi: ”Edistymiskäyrä päivitetään harvoin / silloin tällöin.”
 - Tavoite: Suunnittelupokerin avulla saadaan hyvin estimoidut työtehtävät.
 - Havainto: Tiimin jäsen, joka esittää suurimman tai pienimmän estimaatin, joutuu selittämään ehdotuksensa.
 - Deskriptiivinen normi: ”Pyri estimoimaan keskimmäistä lukua (jotta et joudu selittämään valintaasi muille).”
- Molempia voi esiintyä yhtä aikaa

Normit

- Psykologinen turvallisuus
 - Tunne siitä, ettei ryhmä rankaise, hylkää tai nolaa jäsentään jos hän nostaa (vaikeita) asioita esiin
 - Tunnistettu tuottavien ohjelmistotiimien ominaisuudeksi
 - Esimerkki:
 - Testaaja A: ”Minun pitää miettiä tarkkaan miten esitän bugin kehittäjille koska he pitävät sitä kritiikkinä ja loukkaantuvat.” (Normi: kehittäjät reagoivat negatiivisesti bugiraportteihin)
 - Testaaja B: ”Kehittäjät ovat iloisia kun löydän bugin.” (Normi: kehittäjät reagoivat positiivisesti bugiraportteihin)

Arvojen ja normien suhde käyttäytymiseen ketterissä menetelmissä



Muokattu:

Stray V., Fægri T.E., Moe N. B. (2016) Exploring Norms in Agile Software Teams. International Conference on Product-Focused Software Process Improvement. Springer.
Kirch, L. J., Ko, D.-G., Haney, M. H. (2010) Investigating the antecedents of team-based clan control: adding social capital as a predictor. Organ. Sci. 21, 269-489.

Yhteinen pohdinta



Kysymykset, gradu-aloitteet, ym.: fabian.fagerholm@helsinki.fi