

Agile

Manifesto of Agile Software Development (2001):

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

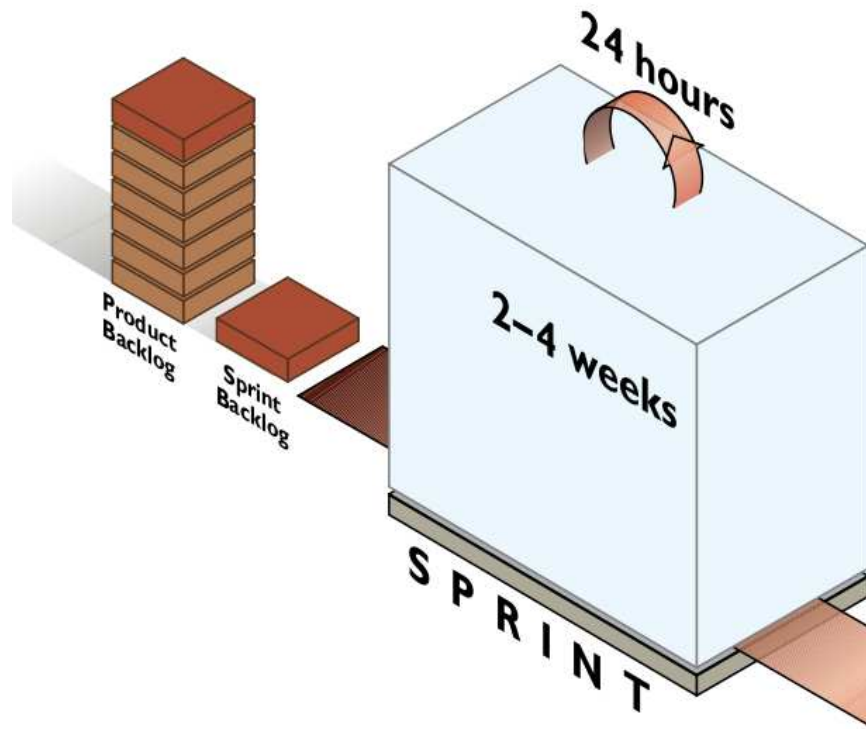
That is, while there is value in the items on the right, we value the items on the left more.

Lähde: <http://www.agilemanifesto.org>

**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, Ro-
bert C. Martin, Steve Mellor, Ken Schwaber, Jeff
Sutherland, Dave Thomas**

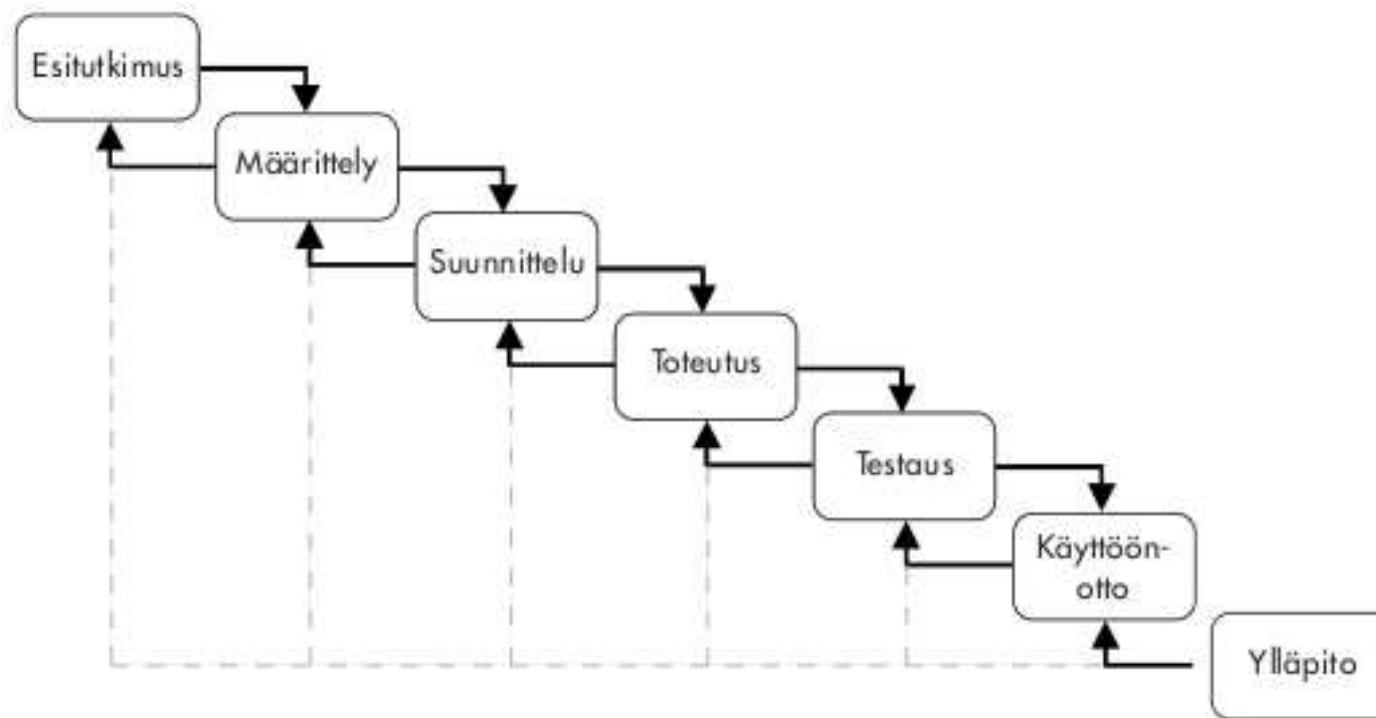
Lähde: <http://www.agilemanifesto.org>

Esimerkki Agile-prosessista (Scrum)



Lähde: http://www.softhouse.se/Uploades/Scrum_eng_webb.pdf

...vertaa tähän (vesiputousmalli)



Principles of Agile Software

- ...
- Working software is primary measure of progress
- ...

Lähde: <http://www.agilemanifesto.org/principles.html>

Agile-menetelmistä

- Väite: Agile-menetelmät ovat epämuodollisia, suunnittelemattomia "cowboy" -menetelmiä
- Höpö höpö. XP:n (Extreme Programming) käytänteet muodostavat (jopa liian?) kurinalaisen "common sense" -menetelmän
- Toisaalta Agile-menetelmien ympärillä edelleen HYPE

Lyhyet iteraatiot

- Asiakkaalle toimitetaan aikaisessa vaiheessa, ja sen jälkeen lyhyin aikavälein, toimiva sovellus
 - Inkrementaalinen ja iteratiivinen ohjelmistonkehitys
-
- **mahdollistaa muutoksen**
 - pakottaa suunnittelemaan ja seuraamaan projektia
 - pakottaa paljastamaan asiakkaalle työtahdin
 - puolin ja toisin realistiset odotukset
 - motivoi
 - mahdollistaa oppimisen (reflektointi) projektin aikana
 - konkretian esille tuonti tarkoittaa asiakasvaatimuksia

...iteraatioista

- Projektin alussa ”nollaiteraatio”
- Iteraation reunat: suunnittelu ja hyväksyntä
- Iteraation aikana työrauha (SCRUM korostaa), ts. vältettävä kaaosta, vrt. XP: on-site customer
- Päivittäiset lyhyet kehittäjäpalaverit (Stand-up meeting, Daily Scrum)

Kommunikointi

- Kasvotusten kommunikointi nähdään ideaalisena vuorovaikutustapana
- Uskaltaanko luottaa kasvotusten kommunikointiin?
- Kumpi on suurempi riski: kommunikointi erilaisia kaavioita katselmoiden vai ihmisten välinen keskustelu?
- Dokumentaatio vs. kehitysympäristö ja projektin piirteet (koko)?

Kehitystiimi

- Kommunikoi suoraan asiakkaan kanssa
- Itseorganisoituu, ei kiinteitä rooleja
- Koostuu (ideaalitapauksessa) eri alojen asiantuntijoista
- Coach, Scrum-master

...yhteenvedtoa...

- Väite: Agile-menetelmät soveltuvat kokeneille tiimeille?
- Entäpä arvot (manifesti)?

XP vs. Scrum

- XP määrittelee prosessin konkreettisten käytänteiden avulla, ts. ottaa kantaa miten ohjelmoidaan, testataan jne.
- Scrum puolestaan on tarkoitettu Agile-projektien hallintaan, ottamatta kantaa tiimin sisäisiin työskentelytapoihin
- Scrumia voidaan täydentää XP:n käytänteillä

Suunnittelu (Planning, XP)

- "Mitä tehdään seuraavaksi"
- Käyttäjätarinat (user stories): yksinkertainen dokumentointi esim. pahvikorteille
- Kommunikointi keskeisessä asemassa
- Suhteelliset arviot (esim. pisteet) tarinoiden toteutuksen vaativuudesta
→ vertailu aiemmin toteutettuihin toiminnallisuuksiin yksinkertaista
- Reflektointi avuksi, arviointia pystytyään parantamaan aikaisempien iteraatioiden tulosten perusteella
- Käyttäjätarinat suppeampia kuin use case
- Välitön arviointi (cost), vrt. "perinteisten" vaatimusten kirjoittaminen
- Cohn: A good story is independent, negotiable, valuable to users or customers, estimatable, small, and testable

Esimerkki (Cohn):

A company can pay for a job posting with a credit card.

Note: Will we accept Discover cards?

Note for UI: Don't have a field for card type (it can be derived from first two digits on the card).

Lähde: Cohn, M., Writing Stories (chapter 2), From "User Stories Applied", 2004

"XP-Spike"

- Tutkitaan ongelmallista toteutuksen osa-aluetta etukäteen
- Kestoltaan esim. yksi iltapäivä (kiinteä arvio)
- Perusteltua valita spike toteutettavaksi, ja story, jota spike tutkii vasta (mahdollisesti) seuraavaan iteraatioon
→ samaan iteraatioon sijoittaminen lisää epävarmuutta, koska spiken tulosta ei tunneta

Iteraatioon valittujen storyjen tarkennus (XP)

- Kehittäjät jakavat storyt ohjelmointitehtäviksi (tasks)
- Tarvittaessa tarkentavia kysymyksiä asiakkaalle
- Suhteelliset arviot taskeille
- Velocityn tarkistus → tarvittaessa uudelleen priorisointi asiakkaan kanssa

Implementoinnista (XP)

- Test Driven Development (TDD)
- Pariohjelmointi
- Refaktorointi
- Jatkuva integrointi
- Yhteinen koodin omistajuus
- Yhteiset koodauskäytännöt
- Pyrkimys yksinkertaisiin ratkaisuihin, YAGNI

Iteraation hyväksyntä

- Hyväksyntää ei saa ohittaa, muuten iteratiivisuus jää näennäiseksi
- Hyväksynnän kohdistuttava siihen, mitä on sovittu tehtäväksi
- Storyn hyväksyntätestit voidaan asiakkaan kanssa kirjoittaa esim. story-kortin kääntöpuolelle
- Pyrkimys hyväksyntätestien automatisointiin

Esimerkki (Cohn)

- Test with Visa, MasterCard and American Express (pass).
- Test with Diner's Club (fail).
- Test with good, bad and missing card ID numbers.
- Test with expired cards.
- Test with over \$100 and under \$100.

Lähde: Cohn, M., Writing Stories (chapter 2), From "User Stories Applied", 2004

Release Plan

- Karkea suunnitelma toiminnallisuuksista, joita julkaistaviin sovelluksen versioihin halutaan sisällyttää.
- Esimerkiksi lista storeja, jotka kiinnitetään tiettyyn iteraatioon ja julkistukseen (release)
- Saa muuttua

Edelleen ... kenelle Agile ei sovellu?

Esim. Beck arvelee XP:stä:

- Organisaatiohin, joiden todelliset arvot eivät vastaa XP-arvoja?
- Suuriin kehitystiimeihin (satoja ohjelmoijia)?
- Sovellusympäristöihin, joiden teknologia ei tue muutosta?
- Huom. soveltuvuutta arvioitava tapauskohtaisesti?

Entä käytänteet kuten "Distributed Pair Programming"

Practices of XP

- whole team
- metaphor
- the planning game
- simple design
- small releases
- customer tests
- pair programming

Practices of XP

- test-driven development
- design improvement
- collective code ownership
- continuous integration
- sustainable pace
- coding standards

Agile-kokemuksia ?