



Onnistunut ohjelmistoprojekti

2.12.2008

Hermann Hyytiälä
Reaktor Innovations Oy



Agenda

- Yritysesittely
- Keinoja onnistuneeseen ohjelmistoprojektiin
 - Ihmiset
 - Menetelmät
 - Käytännöt ja työkalut
- Tulevaisuuden trendejä



Reaktor Innovations Oy

- Perustettu vuonna 2000
- Olemme ohjelmistotuotannon erikoisosaaja
- Olemme asiantuntijaorganisaatio, jossa ihmiset viihtyvät
- Toimintamme perustuu huippuosaamiseen, asiakkaan kunnioittamiseen, rehellisiin ja avoimiin toimintatapoihin ja inhimillisiin arvoihin
- Yli 100 huippuammattilaista





Merkkejä onnistuneesta ohjelmistoprojektista

- Ohjelmisto tekee mitä pitää
- Tyytyväinen asiakas
- Teknisesti korkealaatuinen
- Asiakkaalle hyvä investointi
- Ohjelmistoa on mukava käyttää
- Tekijät ja muut sidosryhmät kokivat mielekkääksi



Liiketoiminnan näkökulma

- Fakta: markkinoilla kilpailu kovenee jatkuvasti
- Mikä oikeasti on tärkeää?
 - Kyky vastata muutokseen
 - Toimiva ohjelmisto, joka saadaan käyttöön nopeasti
 - Kilpailuedun saavuttaminen
- Tyytyväinen asiakas ja käyttäjä



Mitä aikaisemmin asiakas voi käyttää ohjelmistoa tai sen toimivaa osaa, sen enemmän asiakas ohjelmistosta hyötyy.



Miten tähän päästään?



Keino #1: ihmiset



Ihmiset

- Tärkein osa ohjelmistoprojektia
- Ihmisten käyttäytymistä vaikea ennustaa ja hallita
- Yksittäisen ihmisen, tiimin ja sidosryhmien tuottavuuden maksimointi olennaista ohjelmistoprojekteissa



Tuottavuuden tekijät

- Osaamisen taso
- Halu oppia
- Ihmisten välinen viestintä
 - Ymmärtäminen
 - Viestintälatenssit
 - Luottamus ihmisten välillä
 - Sosiaaliset taidot
- Tiimin toiminta, yhteistyö asiakkaan ja muiden sidosryhmien kanssa



Ohjelmistoprojekti on oppimisprosessi!



Oppimista ja viestintää tulee tehostaa!



Jos tekisit aiemmin tehdyn
projektin uudelleen,
tekisitkö sen nopeammin vai
hitaammin?



Erona I. kertaan: oppiminen!



Myös asiakas oppii omista
tarpeistaan projektin aikana
- tähän annettava
mahdollisuus!



Keino #2: menetelmät



Menetelmät

- Ohjelmistoprojektit luonteeltaan ovat sellaisia, joita ei ole tehty koskaan aikaisemmin
- Ennustettavat prosessit vaativat komponentteja, joiden käyttäytyminen voidaan ennustaa
- Ihmiset eivät ole ennalta arvattavia komponentteja!
- Mallintava vai oppiva?



Onko tärkeämpää, että
ohjelmisto tekee sen mitä
joskus on määritelty vai
sen, mitä asiakas tai käyttäjä
aidosti tarvitsee?



Osaammeko ennustaa tulevaisuutta?



Vaihtoehtoinen tapa:
pysähdy, opi ja reagoi



Oppiminen

- Ihminen oppii havainnoistaan ja virheistään
- Oppimisen iteraatio:
 - Pysähdy
 - Opi
 - Reagoi
 - Aloita alusta
- Tehokkuuden ja epävarmuuden kannalta iteraatioiden oltava säännöllisiä, jotta tiedämme missä olemme ja mihin olemme menossa



Ohjelmistoprojekti käytännössä

- Luodaan visio tulevasta ohjelmistosta
- Työtä tehdään läpinäkyvästi pienissä inkrementeissä iteraatio kerrallaan
- Helpompi hallita
- Säännölliset tapaamiset tekijöiden kesken ja tekijöiden ja asiakkaan välillä
- Jatkuvasti kehittyvä prosessi
- Mahdollisimman vähän viestinviejiä tekijän ja asiantuntijan välillä



Enter: Agile methods



Manifesto for Agile Software Development

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.

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.



Keino #3: käytännöt ja työkalut



- Testivetoinen ohjelmistokehitys
- Testauksen automatisointi
- Jatkuva integraatio
- Refaktorointi
- Pariohjelmointi
- Jatkuva katselmointi
- Versionhallinta
- Task boardit työn ohjaukseen ja seurantaan



Auttaa kehittäjiä, koska...



- Lyhyet vasta-ajat
 - Nopea reagointi
- Ohjelmiston integriteetti pysyy korkeana
- Tehdään asioita pienissä inkrementteissä
- Laatu rakennetaan sisään
- Tehokas viestintä: tietämyksen siirto



Miksi ketterät menetelmät toimivat?

- Antavat mahdollisuuden oppimiselle ja reagoimiselle
- Tehdään oikeasti oikeita asioita, ei tehdä turhia asioita
- Tehostavat ihmisten välistä viestintää
- Virheet nopeasti esille
- Hallitaan pienempiä kokonaisuuksia
- “Rakennetaan laatu ohjelmiston sisään”
- Vastataan asiakkaan muuttuviin tarpeisiin



Tulevaisuuden trendejä?



Menetelmät

- Olemme oppimassa parempia tapoja tehdä ohjelmistoja
 - Ala on vielä suhteellisen nuori (vrt. muut alat)
 - Opitaan mikä on tärkeää, ja mikä vähemmän tärkeää
- Uudet menetelmät ja periaatteet tulossa
 - Agile ja Lean
 - Menetelmät valittavat tilanteen mukaan
- Menetelmien adaptiivisuus; menetelmä(t) tilanteen mukaan



Teknologia

- Multi Core -prosessorit
 - Rinnakkaisuuden hallinta
 - Funktionaaliset ohjelmointikielet
- Monikielisyys
 - Platformi-vetoisuus (JVM, .NET)
 - Oikea ohjelmointikieli oikeaan tarpeeseen
- KISS
 - “Developer time is expensive, servers are cheap”
- Integroinnin tarve ja määrä kasvaa



Älkää unohtako seuraavia asioita!



Emme ole mukana korkean teknologian alalla!



Ohjelmistokehitys on joukkuepeli!



Olemme tietämyksen
tehokkaan siirron alalla -
teknologinen osaaminen on
vain työkalu työn
tekemiseen!



“We are mostly in the human
communication
business” (Peopleware, 1987)

Lopuksi...



Kuinka tulla huippuammattilaiseksi?

- Motivaatio ja asenne tärkeää
- Hyväksi koodaajaksi oppii vain koodaamalla
 - Omista virheistä oppii
 - Lukemalla muiden kirjoittamaa koodia
- Tehdä hyvää laatua
- Yhdessä tekemällä ja oppimalla



Kiitos!

hermanni.hyytiala@ri.fi

<http://www.ri.fi>