

TE Object

“Perinteinen” oliomenetelmä ketteräksi



Heini Holopainen ja Eija Hamina - Mäki
TietoEnator Oyj/ Public & Healthcare/dGov

TietoEnator^{TE}
Building the Information Society

Copyright © TietoEnator 2003

Agenda

- Agile Manifesti
- Ketterän menetelmän päätavoitteet
- Ketterät menetelmät ja niiden yleisiä ja yhteisiä piirteitä
- Ketterä menetelmä ↔ Perinteinen oliomenetelmä **TE Object**
- Milloin menetelmä on ”agile”
- Hajoitustehtävä

TietoEnator^{TE}
Building the Information Society

Copyright © TietoEnator 2003

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

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

Ketteryys

- Yleisesti ketteryydellä tarkoitetaan ohjelmistotuotannonossa :
 - dynaamisuutta
 - joustavuutta
 - suunnan vaihdon nopeutta/ vikkelyyttä
 - tehokkuutta
 - vuorovaikutteisuutta



Ketterän menetelmän päätavoitteet



- Pyritään vastaamaan liiketoiminnan vaatimuksiin kevyistä, nopeista ja nokkelista ohjelmistotuotanto prosesseista
- Päätavoitteena tuottaa asiakkaalle jatkuvasti ja aikaisessa vaiheessa konkreettista tulosta (sykli/ inkrementit)
- Pyritään hallitsemaan muuttuvat vaatimukset, jopa pitkälle edenneessä ohjelmistotuotantovaiheessa.

Ketterän menetelmän päätavoitteet

- Pyritään suoraan vuorovaikutukseen asiakkaan kanssa. Asiakas on osa projekti- / ohjelmistokehitystiimiä.
 - Motivoivaa henkilöstöhallintaa- roolit oikein – itseorganisoituva/ tiimi- motivoitunut tiimi.
- Huomiota kiinnitettävä:

- taitoihin, osaamiseen
- viestintään
- yhteisöllisyyteen



Ketterän menetelmän päätavoitteet

- Edistää kestäväää kehitystä
- Yksinkertaisuus on hallintaa - kuvataan vain tarvittavat asiat ja vain tarvittavalla tasolla, tuotetaan vain toiminnallisuus/ joustavuus, josta on hyötyä (tällä hetkellä). Keskitytään hallitsemaan epätäydellisyyttä (milloin valmis?)

Ketterän menetelmän päätavoitteet

- Arvioidaan jatkuvasti etenemistä ja toimintaa, jotta pystytään tulemaan "laadukkaasti" tehokkaammiksi.
- Korostaa prosessin lisäksi voimaa, yhteisöä, ylpeyttä ja oppimista



Ketteriä menetelmiä

- **XP- Extreme programming** (Beck 1999), ”yksinkertaisesti pyrkimys saada työ tehtyä”. /commonsense principles and practise to extreme level. Kurinalaisuus lisääntyy

<http://www.extremeprogramming.org/>

<http://www.xprogramming.org/>

- **SCRUM** (Schwaber 1995; Schwaber and Beedle 2002). Lähtökohtana on hallita ohjelmistotuotantoprosessia ja hallita/tukea muuttuvaa ympäristöä/ vaatimuksia.

<http://www.controlchaos.com/>

Ketteriä menetelmiä

- **Crystal family of methodologies** (Cockburn 2002). Sisältää useita eri menetelmiä. Useita erilaisia lähestymistapoja erilaisille projekteille, erilaisia toimintamalleja, joita voi käyttää.

<http://www.crystallmethodologies.org/>

- **FDD – Feature Driven Development** (Palmer and Felsing 2002). Prosessiorioitunut ketterä tapa. Keskittyy enemmän suunnittelu ja rakentamisvaiheisiin, ei kata koko ohjelmistotuotantoprosessia.

<http://www.nebulon.com/fdd/index.html>

Ketteriä menetelmiä

- **RUP - The Rational Unified Process** (Kruchten 1996, Kruchten 2000). Oliopohjaisten järjestelmien iteratiivinen ohjelmistotuotantomenetelmä.

<http://www.rational.com/products/rup/index.jsp>

- **DSDM - Dynamic Systems Development Method** (Stapelton 1997). Vapaasti hyödynnettävä sovelluskehys RAD ohjelmistotuotantoon. Kiinnittää ajan ja resurssit ja sitten siinä puitteissa säädetään toiminnallisuus (ei kaikkea kerralla)

<http://www.dsdm.org/>

Ketteriä menetelmiä

- **ASD - Adaptive Software Development** (Highsmith 2000). Keskittyy pääasiassa suurten, monimutkaisten järjestelmien ongelmiin. Painottaa inkrementaalista ja iteratiivista ohjelmistotuotantoa ja jatkuvaa protoilua.

<http://www.adaptivesd.com/>

- **AM – Agile modelling** (Amber 2002). Lisäys/ lisäarvo olemassa olevaan menetelmään – se ei ole täydellinen menetelmä. Sisältää joukon käytännön ohjeita ja periaatteita.

<http://www.agilemodeling.com/>

Ketteriä menetelmiä

- **PP – Pragmatic Programming** (Hunt & Thomas 2000). Joukko mielenkiintoisia ohjelmointi käytäntöjä/ vinkkejä, ei varsinainen menetelmä (ei prosessia, vaiheita, tai rooleja)

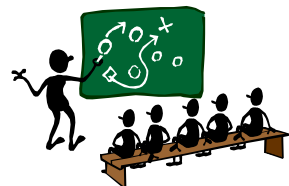
<http://www.pragmaticprogrammer.com/>

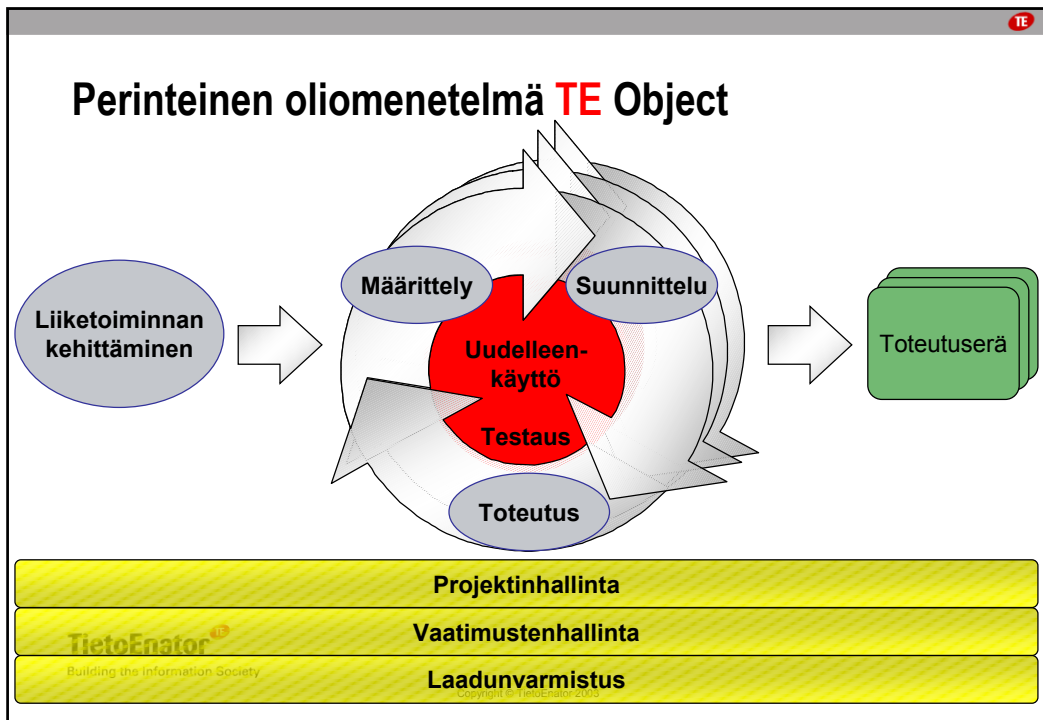
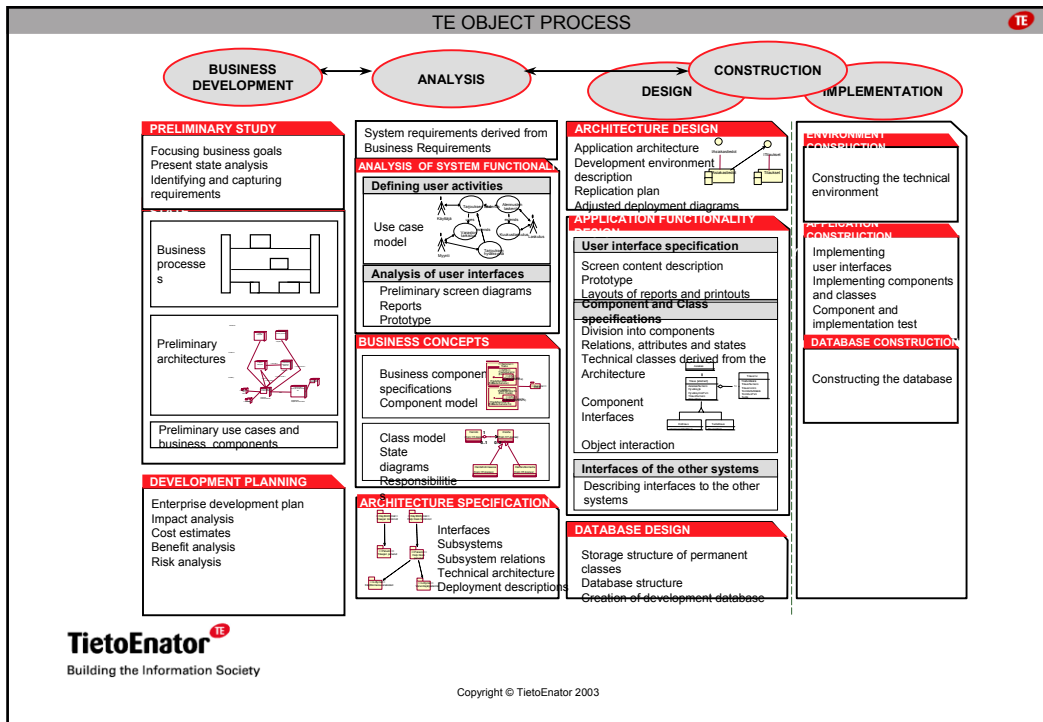
- **OSS - Open Source Software development** (O'Reilly 1999). Yhteneväisyyksiä agile - menetelmien kanssa

<http://www.opensource.org/>

Mitä yhteisiä piirteitä on ketterissä menetelmissä

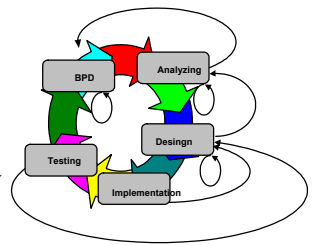
- iteratiivisuus (tiheät syklit)
- inkrementaalisuus (jatkuvasti arvioitavaa tulosta)
- testauksen automatisointi (test first)
- pyritään sallimaan ja hallitsemaan muutokset
- kommunikoinnin ja yhteistyön ja yleensä vuorovaikutuksen korostaminen.
- Suoraviivaisuuden korostaminen



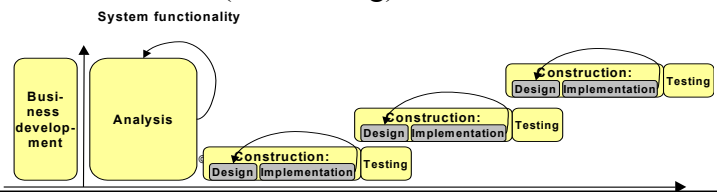


TE Object: Projektihallinta

- Työnjako ja organisointi
 - perustana perinteinen projektihallinta
 - projektipäällikön tunnettava oliolähestymistapa
 - uusia rooleja
- Työtavat
 - kasvattava (inkrementaalinen)
 - vähitellen täsmäntävä (iteroiva)
 - aikaviipaletekniikkaa soveltava (time boxing)
 - protoileva
 - pilotoiva



TietoEnator
Building the Information Society



TE Object: Organisointi

- Osaprosessikohtaiset roolit ja vastuut
 - Toimittaja
 - Asiakas => aktiivisesti mukana
- Yhdellä henkilöllä useampia rooleja
- Laadunvarmistuksen vastuut ja roolit
 - esimerkkirooleja:
 - uudelleenkäytön edistäjä
 - sovellusarkkitehti
 - oliokonsultti

Analysis

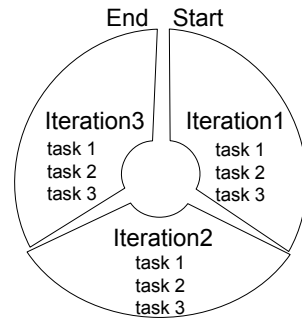
	Vendor				Client	
	Object consultant/analysis consultant	System architect	Technical architect	Quality facilitator	Project manager	Operational expert/technical user
Analysis of system functionality						
Defining user activities	M	C			W	W
Describing the user interfaces	M	C			W	W
Use case model review	W			M	W	
Business concepts specification						W
Describing business classes	M	C				W
Describing business components	M	C	W		W	W
Business class review	W	W		M	W	
Architecture specification						
Describing system interfaces	W	M	C			
Describing technical architecture	W	W	M			
Analysis and results review	W	W	W	M	W	

• Aloittelija : asiantuntija = 2 : 1

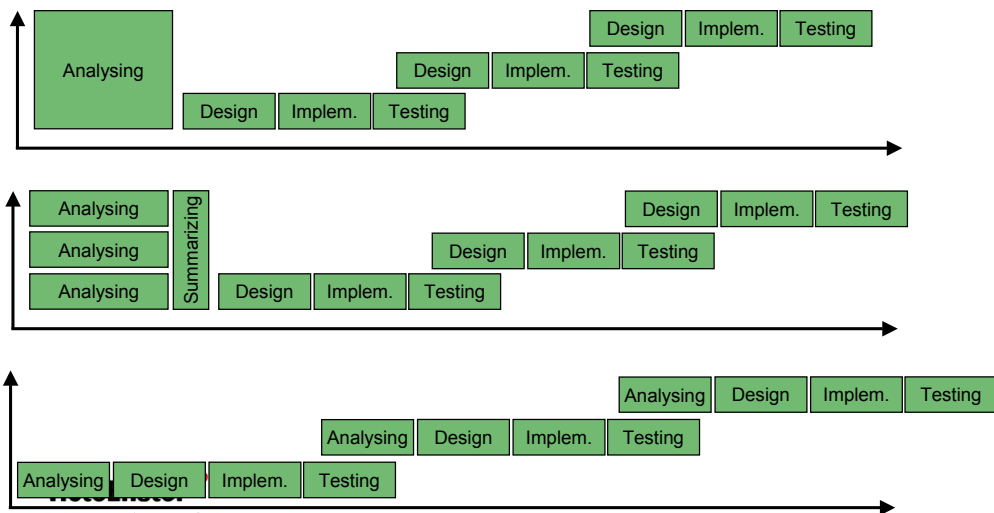
TietoEnator
Building the Information Society

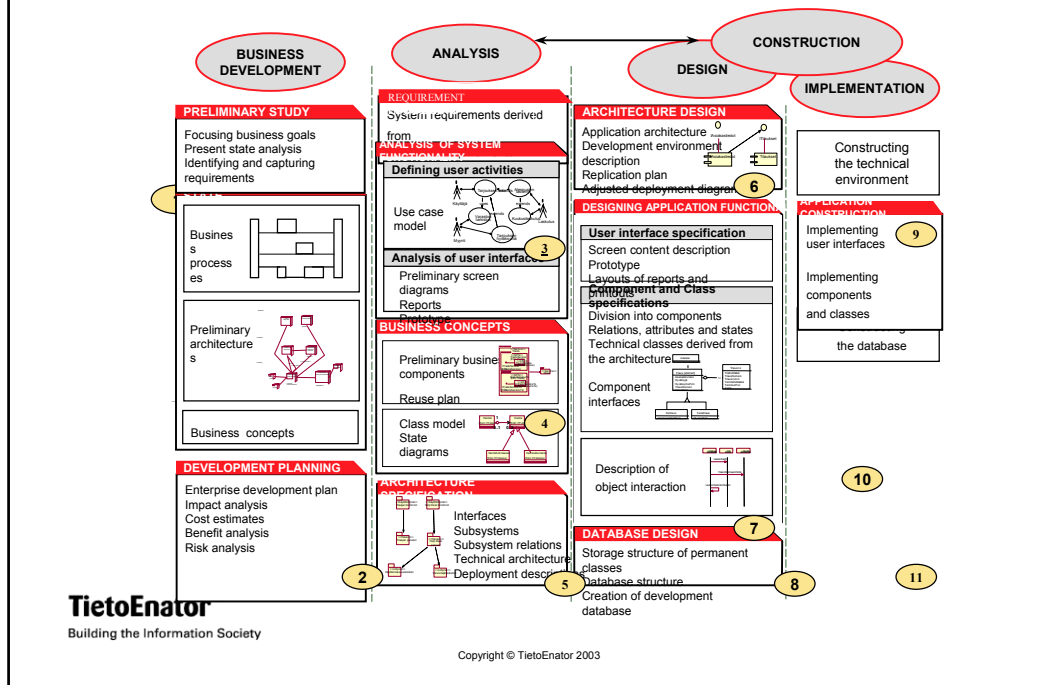
TE Object: Iteratiivinen työskentely

- Varmista tulosten laadullisuus
 - Vertaile vaatimukseen
 - Hyväksytä tulokset
- Sovi ja kirjaa projektisuunnitelmaan
 - Iterointikertojen lukumäärän
 - Jokaisen iterointikierroksen lopputulokset ja tarkkuustaso
 - Iteroinnin työskentelytapa
 - Iteroitavat työt/tehtävät
- Iterointikierrosten välillä
 - Varmista tulosten laatu
 - Sovi seuraavan kierroksen tavoitteet
 - Tarkista vaikutukset projektiin (muutoshallinta)



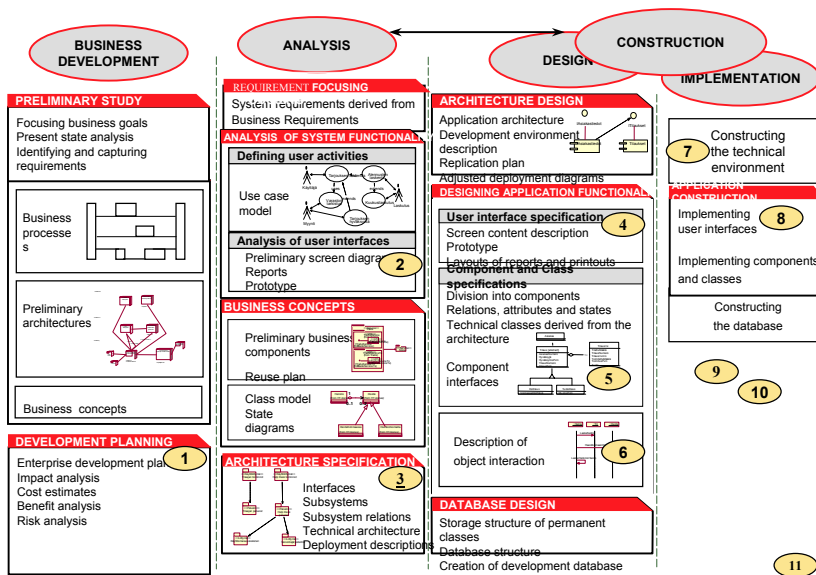
TE Object: Inkrementaalinen työskentely





TE Object: Laadun varmistus

- Tavoitteena varmistaa tulosten oikeellisuus ja oikea-aikaisuus
- Seuranta auttaa laatupoikkeamien havaitsemiseen ajoissa -> korjaavat toimenpiteet ajoissa
- Katselmoinnit
 - jokaisen osaprosessin lopussa
 - tarvittaessa osaprosessin aikana
- Projektitoiminnan tueksi erillinen toimintaohjeisto -> projektikatselmuksset



Milloin menetelmä "on agile"

- Ketteryys vähenee järjestelmän koon kasvaessa
- Ketteryyteen voidaan pyrkiä osittamalla järjestelmän kehitys "itsenäisiin" palasiin
- Tukee inkrementaalista ja iteratiivinen työskentelytapaa
- Menetelmä on riittävän tarkka ja riittävän yksityiskohtainen
 - Menetelmä täyttää tarkoituksensa, on joustava, on sovellettavissa
 - Periaatteet helposti opittavia suoraviivaisia
 - Täydellisyyttä ei voi saavuttaa

Milloin menetelmä ”on agile”

- Tuottaa projektille positiivista lisäarvoa
 - Auttaa projektia pääsemään päämääräänsä
 - Vähentää kustannuksia
- Menetelmä on ymmärrettävä
 - Vaatimukset määritellään ensin liiketoiminnan kielellä
 - Jos käytetään UML kuvaustapaa, tulee käyttäjät kouluttaa käyttämään kuvaustekniikkaa
- Tukee testauksen automatisointia
 - Mieti ensin miten testaat ajattelu

Milloin menetelmä ”on agile”

- Edellyttää asiakkaan osallistumista projektiin.
 - projektitiimin roolit ja vastuut kuvattu myös asiakkaan osalta
- **”Sallii maalaisjärjen käytön”**

Milloin menetelmä ”on agile”

- ”agility in business” EI merkitse
 - tehokkuuden parantamista
 - kustannusten vähentämistä**laadun kustannuksella**

Linkkejä

- Agile alliance: <http://www.agilealliance.com/>
- Martin Fowler's Agile Software Development pages: <http://www.martinfowler.com/>
- VTT Agile Software Technologies <http://mango2.vtt.fi:84/virtual/agile/index.html>

Harjoitustehtävä

- Mieti mitä haasteita tuo ketterä lähestymistapa käytännön ohjelmistotuotantoon.(3 - 5 keskeisintä)

Haasteet

- Vaatimusten hallinta
 - Jäljitettävyys
 - Muutoksen arviointi kustannukset, aikataulut
 - Muutoksen toteutus Milloin?
- Testauksen automatisointi
 - Test first, Mieti kuinka testaat ensin, kirjoita testitapaukset ja koodaa sitten, hahmottuu toiminnallisuuden arvo
 - Testitapausten automatisoitu tuottaminen (activity kaavioiden perusteella ?) Kuka , Miten

Haasteet

- Asiakkaan sitoutuminen 100 %
 - Asiakkaalla on omat hommat, delegointi
 - Asiakkaan johdon sitoutuminen
- Tiimityöskentely, tiimijattelu projekteissa
 - Eri organisaatioiden henkilöiden välillä
 - Henkilöiden osaaminen huomioitava, koulutustarve (kaikki eivät voi olla asiantuntijoita, aloittelijat mukaan
 - Ei ole ”omia ” lapsia”- kuka tahansa saa koskea toisen tekemään tulokseen (dokumentti, koodi jne.)

Haasteet

- Palaute
 - Palautteen antaminen ja palautteen vastaanottaminen
 - Oltava rakentavaa, ei henkilökohtaista
 - Hyväksyttävä, ettei ole ”täysin oppinut” – nöyryys
- Yksinkertaisuus ja riittävyys
 - Vain ne piirteet, jotka tuovat lisäarvoa, mutta ei mitään turhaa.
 - Kaiken ei tarvitse olla kuvattuna 100% tarkasti

TietoEnator^{TE}

Building the Information Society

More time for Users

Heini Holopainen ja Eija Hamina-Mäki

Public & Healthcare

dGov

heini.holopainen@tietoenator.com

eija.hamina-maki@tietoenator.com

TietoEnator^{TE}
Building the Information Society

Copyright © TietoEnator 2003