

Matti Vuori

Erilaisia tapoja tehdä ohjelmistokehitystä – eli miten luoda ketteryyttä ja ennakkosuunnittelua oikeisiin paikkoihin

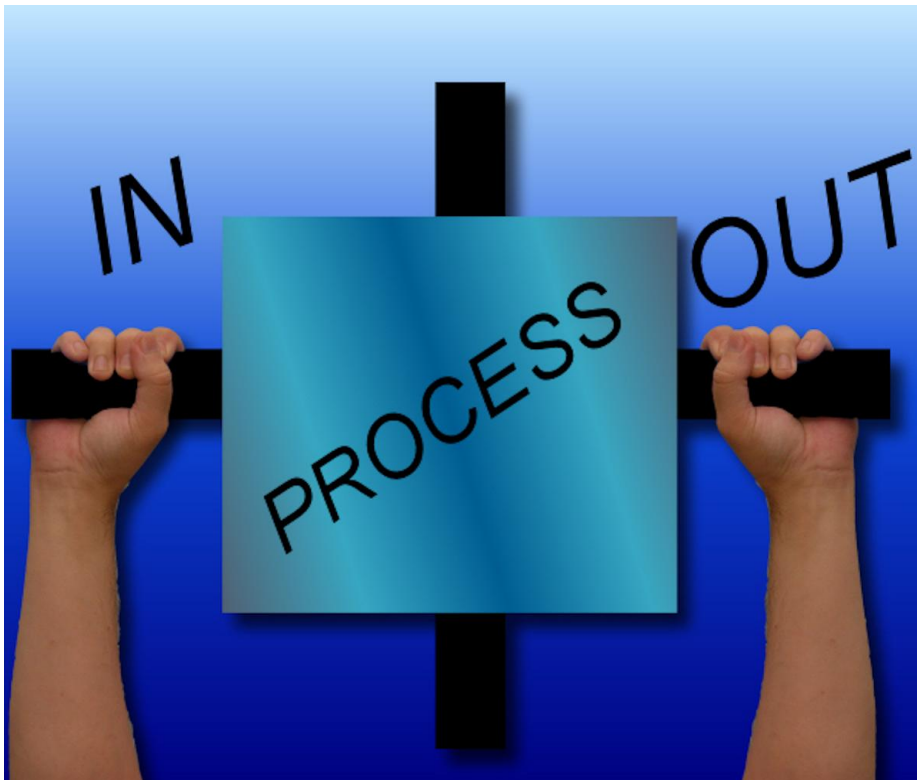
Johdanto

Kun esimerkiksi ohjelmistokehityksen projektimalleja kritisoidaan, unohtuu usein se, että on olemassa erilaisia tapoja tehdä asioita, joka erilaisuus riippuu tavoitteista. Sen sijaan ajatellaan, että on yksi paras tapa ja muut tavat ovat jotenkin vanhentuneita, epäkypsiä tai kehoja. **Usein väitetään, että juuri Suomessa mahtuu asiaan kuin asiaan kerrallaan vain yksi totuus.**

Joskus voi ollakin niin, että on yksi ja yleinen paras tapa, mutta yleensä voi kyse olla vain erilaisen kontekstin erilaisista vaatimuksista ja mahdollisuuksista. Tässä paperissa tarkastellaan lyhyestä erilaisia tuote- ja ohjelmistokehityksen tilanteita ja niiden prosessien keskeisiä piirteitä sekä ennakkosuunnittelun ja ketteryyden ilmenemismuotoja.

Tämä on tärkeää siksi, ettemme aseta toiminnalle sellaisia vaatimuksia, jotka ovat sille epärelevantteja, vaan osaamme antaa vaikkapa kuumien tiimien tai tutkijoiden tehdä työnsä sillä tavalla, joka on niissä paras ja tuloksellisin, emmekä tyrkytä niihin tilanteisiin vaikkapa jotain projekti- tai laadunvarmistusmalleja, jotka on suunniteltu ”teolliseen” järjestelmäkehitystapaan. Ja toisinpäin – on osattava kirrkaasti nähdä ne tilanteet, joissa tarvitaan prosessityötä ja kriittisten asioiden systematisointia.

Niinpä, 2010-luvun menestystekijöitä eivät ole tasapäästäminen ja yhden koon ratkaisut, vaan esimerkiksi kontekstietoisuus, tarkoituksenmukaisuus, kriittisten tekijöiden tunnistaminen ja erilaistamisen riskinotto.



Erilaisia tyyppitilanteita

Esim. seuraavissa tyyppitilanteissa on tarve erilaisiin prosesseihin:

- **Asiakaslähtöisiin vaatimuksiin perustuva projektitoimitus**
 - Esimerkiksi uniikin tietojärjestelmän kehittäminen organisaatiolle.
 - Tarvitaan systemaattinen projektimalli.
 - Keskeistä on hallinta ja varmistus – "devil is in the details".
 - Ennakkosuunniteltua: Projektimalli (ketteräkin sellainen on tiukan epäketterän ennakkosuunnittelun tulosta!), projektointi, laadunvarmistuksen malli, testausprosessi, riskienhallinta, katselmoinnit, asiakkaan tekemän hyväksynnän menettelyt, hyväksymistestauksen peruspiirteet.
 - Ketterää: Testauksen sisällöt, vaatimusten iterointi tai ketterä kehittäminen, hyväksymistestauksen yksityiskohdat.
 - *Huom! Tässä ja kaikissa seuraavissa kohdissa eivät ennakkosuunnittelujen ja ketterien asioiden listat ole läheskään kattavia, vaan ainoastaan tilannetta ja toimintatyyliä visualisoivia.*
- **Vakiintuneen konseptin / alustan implementointi / räätälöintiprojekti**
 - Esimerkiksi tietyn toiminnanohjausjärjestelmä-alustan räätälöinti yritykseen.
 - Tarvitaan systemaattinen projektimalli.
 - Keskeistä on hallinta ja varmistus.
 - Keskeistä on hallinta ja varmistus – "devil is in the details".
 - Ennakkosuunniteltua: Räätälöinnin "teollinen" toteutustapa alusta loppuun.
 - Ketterää: Räätälöinnin ja integroinnin detaljit, integrointien testaus.
 - Hienoa: Tehokas tehdasmaisuus tekee toiminnasta sujuvaa ja vähentää riskejä – kunhan kohdetilanne on arvioitu oikein.
- **Systemaattinen uuden tuotteen kehittäminen**
 - Esimerkiksi tunnistettujen markkinatarpeiden tai asiakasryhmän vaatimusten perusteella tehtävä tuotekehitys.
 - Traditionaalinen tuotekehitysprosessi.
 - Konseptivaiheen, suunnittelun ja toteutuksen jatkumo, porttiajattelu.
 - Ennakkosuunniteltua: Tuotekehitysprosessi.
 - Ketterää: Prosessin yksittäiset menettelyt. Useimmiten varsinainen kehittäminen ei voi olla kovinkaan ketterää – inkrementaalinen kehittäminen uusien versioiden kautta on eri asia.
- **Innovaatiolähtöinen tuotekehitys**
 - Esimerkiksi tilanne, jossa uusia konsepteja kehitellään organisaation tutkimuskeskuksessa useita ja niistä poimitaan osa tuotteistukseen.
 - Olennaista on innovaation kehittäminen innovaattorille / innovaattoreille sopivalla tavalla.
 - Tuotteistus ja paketointi on suoraviivaista, nopeaa ja tehokasta.
 - Olennaista on oikean konseptin löytäminen ja sen muuttaminen innovaatioksi.
 - Ennakkosuunniteltua: Tuotteistusprosessi ja sen yhteydessä tapahtuva testaus ja muu laadunvarmistus.
 - Ketterää: Innovointiprosessi, innovoinnin yhteydessä tapahtuva testaus, proof of concept -testaus.



- **Tutkimuslähtöinen tuotekehitys**

- Esimerkiksi valtiollinen tutkimuskeskus tekee teknologiatutkimusta ja tuotteistaa sen tuloksia.
- Olennaista on tutkimuksen tekeminen sen pelisäännöillä ja herkkyys sen tulosten hyödyntämismahdollisuuksille.
- Tuotteistus ja paketointi on suoraviivaista.
- Olennaista on oikean konseptin / teknologian löytäminen ja sen muuttaminen innovaatioksi.
- Ennakkosuunniteltua: Puitteet, keksintöjen käsittely ja tuotteistus.
- Ketterää: Tutkimustyö, proof of concept -testaus.

- **Systemaattinen olemassaolevan tuotteen edelleen kehittäminen**

- Esimerkiksi nettiselaimen jatkuva parantaminen vuosi vuodelta.
- Kilpailutilanteessa se voi olla luonteeltaan ketterää ja kiivasta.
- Tuotehallinnan, ylläpidon ja kehittämisen näkökulmat yhdistyvät ja painottuvat eri tavoilla.
- Proaktiivinen ja reaktiivinen kehittäminen, parantaminen, korjaus.
- Ennakkosuunniteltua: Ylläpitomalli, kehittämisen prosessi, suuntaa-antava roadmap, tuotteen eheyden ja regressioriskin hallinta (ml. regressiotestaus).
- Ketterää: Mahdollinen ketterä kehittämismalli, ketterä reagointi asiakas- ja käyttäjätarpeisiin ja – tietoihin, uusien vaatimusten tunnistaminen, testaus ilmiöiden perusteella.

- **Yhteisöllinen kehittäminen**

- Esimerkiksi avoimen lähdekoodin käyttöjärjestelmän kehittäminen.
- Orgaaninen kehittämisprosessi yhteisössä.
- Olennaista on uusien toimintojen hyväksymisprosessi ja testaaminen.
- Kun ei ole kattavaa testausjärjestelyä, pitää uusia julkistuksia kysytellä erilaisten hyväksymistasojen läpi.
- Ennakkosuunniteltua: Yhteisön pelisäännöt, uusien toimintojen ja muutosten hyväksyminen, testausvaatimukset.
- Ketterää: Reagointi yhteisön impulsseihin, riskienhallinta.

- **Sosioteknisen järjestelmän osallistuva kehittäminen**

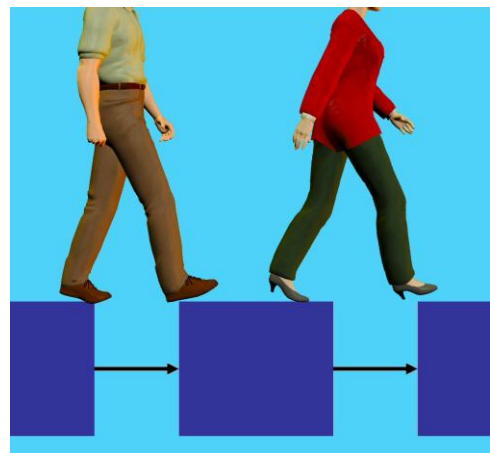
- Esimerkiksi käyttäjät ja kehittäjät luovat yhdessä valvomojärjestelmää ja sen käyttöprosesseja.
- Olennaista on rajata tämä kehittämistapa käyttöliittymäkerrokseen ja tehdä tekniikka toisella prosessilla.
- Systemaattista: Osallistuvan suunnittelun menettely ja työn ohjaus sen perusteella. Jäädetyksen jälkeinen laadunvarmistus.
- Ketterää: Reagointi prosessin aikana havaittuihin uusiin asioihin järjestelmästä ja sen käytöstä. Ketterä käyttöliittymäratkaisujen validointi.

- **Kuumat tiimit**

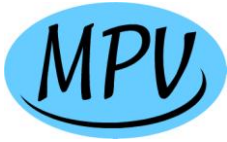
- Erityisessä tilanteessa kootaan poikkeuksellinen tiimi luomaan radikaali ratkaisu ongelmaan.
- Niiden pitää saada toimia millä tahansa tavalla, mikä niille on luontaista.
- ...ja saada aikaan mitä tahansa, mikä niistä on tärkeintä...
- Olennaista ei ole prosessi, vaan ihmisten yhteinen katalyyttinen kuumuus.
- Tiimiltä pitää joskus nimenomaan poistaa prosessivaatimukset!
- Mutta: silloin, kun tiimi ei ole kuuma, tilanne on hyvin erilainen. Kuumuus pitää osata tunnistaa tarkasti...
- Ennakkosuunniteltua: Tiimien käsittely organisaatiossa.
- Ketterää: Kaikki tiimin toiminta!

- **Organisaation omatarve-kehitys**

- Esimerkiksi omaan tietojärjestelmään laaditut laajennukset.
- Prosessi vapaa – riippuu henkilöstä, joka tekee. Vaarana olla luonteeltaan "hakkerointia".
- Testaus jää heikoksi, mikä voi olla ok, jos käyttö on hyvin hallittua.
- Ennakkosuunniteltua: Toteutettavat toiminnot, aikataulu.
- Ketterää: Kaikki muu!
- Kriittistä: Huomattava, milloin tämä malli ei enää toimi, vaan on vaihdettava moodi vakavaksi sisäiseksi kehityspalveluksi. Kaksi kriteeriä: järjestelmän riskitaso ja kehittäjän jatkuva läsnäolo korjaamassa mahdollisia ongelmia.



Hyvä prosessi / toimintamalli antaa turvallisen ja tuottoisan alustan, eikä ole kivireki tai älykkyystesti!



- **Yksilön tekemä ammattimainen ohjelmistokehitys**
 - Esimerkiksi yksityisyrittäjän tekemä ohjelmistotuotteen suunnittelu ja toteutus.
 - Prosessi vapaa.
 - Asiakas päättää, kelpaako ohjelmisto vai ei.
 - Oma geneerinen ammattimaisuus ohjaa tekemistä – oman tietotaidon puitteissa, ilman organisaation tukea...
 - Ennakkosuunniteltua: Ei mikään.
 - Ketterää: Kaikki!
- **Kuluttajan tekemä harrastus- ja omatarve-ohjelmistokehitys**
 - Esimerkiksi yksityishenkilön itselleen tekemä mobiililaitteen ”widgetti”, johon haetaan täsmädataa netistä tai johonkin nettipalveluun itselleen ja kontakteilleen tekemä sovellus.
 - Tällaisen roolin arvellaan kasvavan jatkossa.
 - Prosessi vapaa. On turha odottaa mitään prosessia – sellaisista ei ole ihmisillä mitään tietoa! Siksi välineiden on varmistettava asioita paljon enemmän kuin ammattilaisten välineiden.
 - Ennakkosuunniteltua: Ei mikään.
 - Ketterää: Kaikki!
- **Tiettyn ekosysteemiin puitteissa tehtävä kehitystyö**
 - Esimerkiksi mobiililaitteen kauppapaikkaan hyväksytyjen ohjelmistojen kehittäminen.
 - Prosessi vapaa.
 - Olennaisinta on täyttää ekosysteemin ”omistajan” (esimerkiksi kauppapaikan tai kelpuuttajan) laadunvarmistusvaatimukset.
 - Ennakkosuunniteltua: Ekosysteemin toiminta, roolit, ehdot; laadunvarmistusvaatimukset.
 - Ketterää: Ei mikään.
- **Jne...**

Yllä on varmasti vain osa erilaisista mahdollisista tilanteista. Toivottavasti se hieman valotti tilannetta ja auttaa tunnistamaan omassa toimintaympäristössä erilaisia tyyppitilanteita, joille on eduksi se, että ne saavat hieman erilaista kohtelua. Tärkeää on tunnistaa myös tilanteet, joissa sovelletaan mallia, joka on ”tarkoitettu” toisenlaiseen asetelmaan.