

Pienen organisaation ohjelmistotuotannon käytäntöjen parantaminen



Anne Valsta on Systeemi-työn opettaja, PASMA-hankkeen projektipäällikkö Helia, tietojenkäsittelyn koulutusohjelma anne.valsta@helia.fi

Pienen, kuten minkä tahansa, ohjelmistotuotanto-organisaation haasteena on olla kilpailukykyinen kansainvälisillä markkinoilla. Kun suuret organisaatiot mittauttavat säännöllisin väliajoin kyvykkyytensä standardin asemaan nousseiden kypsyys- eli laatumallien avulla ja osoittavat siten uskottavuutensa markkinoille, ne samalla edesauttavat mallien kehittymistä.

Pieni organisaatio ja sen mittoihin sovitettu laatumalli

Pienten organisaatioiden mahdollisuudet kehittää itse tai ottaa käyttöön de facto standardin asemaan nousseita ohjelmistotuotannon malleja ovat rajalliset. Yhtäältä liiketoiminnan arkirutiinit vievät kaiken ajan ja resurssit, toisaalta paineet kansainvälisten markkinoiden ja ison organisaation kumppanina tai alihankkijana vaativat tunnetun mallin kautta osoitettavaa toiminnan laatua: tehokkuutta, osaamista ja taloudellisuutta.

EU:n luokituksen mukaan pienellä organisaatiolla tarkoitetaan 1-50 henkilön yritystä. Organisaatio voi olla pysyvä tai tilapäinen, kuten tyypillinen projektiorganisaatio tai tiimi, jonka tehtävä on määräaikainen. Organisaation koko voi vaihdella paljonkin sen kasvun eri vaiheissa ja merkitä erilaisten vastuuroolien muodostumista, tehtäväkenttien vaihtumista tai painottumista eri aikoina.

Pienen organisaation koko ja joustavuus tarjoavat sille muita paremmat mahdollisuudet toimia tehokkaasti ja tuottavasti. Pienen organisaation paras resurssi on osaava henkilöstö ja hyvä vuorovaikutussuhde asiakkaiden kanssa. Kasvun myötä tilanne mutkistuu, ja tarve mallille kasvaa. Yhdistämällä toimivat prosessit ja osaava henkilöstö saadaan liiketoiminnalle huomattavaa hyötyä.

Pienen organisaation on varsin vaikea osoittaa tarvittavat resurssit ohjelmistotuotannon prosessien kehittämiseen ja jatkuvaan parantamiseen. Prosessit tulee kuitenkin määrittää ennen kuin toiminta on kasvanut ja saavuttanut mitat, joissa

hallittu prosessi on edellytys toiminnan onnistumiselle, uskottavuudelle ja jatkumiselle. Software Process Improvement (SPI) -paradoksi onkin: "Organisaatio, jossa on vaikea johtaa parantamista ja löytää tarvittavat resurssit (aika, henkilöstö, raha), tarvitsisi eniten SPI:a. Sen sijaan organisaatiossa, jossa on ihanteellinen tilanne (pieni, joustava, hyvä vuorovaikutus, vähän ristiiriitoja), ei todellista tarvetta SPI:lle vielä ilmene."

Ruotsissa, Uumajan yliopistossa, on kehitetty pienen organisaation ohjelmistotuotannon laatumalli: CMM for Small Organisations (nk. Dynamic CMM). Se pohjautuu Software CMM:iin (Capability Maturity Model) ja lähestyy ohjelmistotuotannon parantamisen haasteita pienen organisaation henkilöstön roolittamisen kautta. Kyseinen laatumalli jakaa pienet organisaatiot henkilö- ja tuotemäärän mukaisesti kolmeen luokkaan: XXS (extra extra small), XS (extra small) ja S (small). Luokitus on kuvattu taulukossa 1.

Taulukko 1. Organisaation kokoluokitus

tuotemäärä	henkilömäärä		
	1-2	3-15	16-50
1	XXS	XS	S
2-5	-	XS	S
> 5	-	S	S

Taulukossa 2 on kuvattu Dynamic CMM -mallin mukaiset roolit. Rooleilla on vastuita ja tehtäviä, joita voidaan tarvittaessa yhdistää samalle henkilölle. Mitä pienempi organisaatio, sitä enemmän rooleja on yhdisteltävä yksittäisten henkilöiden kesken. Ainoastaan laadunvalvonta (SQA, Software Quality Assurance) on rooli, jota ei saa sekoittaa sen enempää johtamisen kuin ohjelmiston kehittämisenkään rooleihin.

Rooleja, kuten testaus ja laadunvarmistus, voi myös ulkoistaa. Asiakas pitää sitoa kehittämiseen vahvemmin silloin, kun itsellä ei ole tarvittavia resursseja tai osaamista. Laadunvarmistus voidaan mm. hankkia asiakkaalta, jolla on siihen oma organisaationsa. Laadunvarmistusryhmää

Taulukko 2. CMM for Small Organisations mallin roolit.

Rooli x = on nimettävä + = voidaan ulkoistaa esim. asiakkaalle		Yrityksen koko			Kommentit
		XXS	XS	S	
SM	Senior Manager	x	x	x	
PM	Project Manager	x	x	x	XXS:ssa PM=SM
SWM	SoftWare Manager	x	x	x	XXS:ssä SWM=SM
SE	Software Engineer	x	x	x	
SG	System Engineer Group	x	x	x	XXS:ssa SG=SE
STG	System Test Group	+	x	x	
SCM	Software Configuration Management Group	x	x	x	XXS:ssä SCM=SWM
SCCB	Software Configuration Control Board			x	Pienissä organisaatioissa ei välttämättä eritellä rooleja SCM ja SCCB.
SQA	Software Quality Assurance Group			x	
CSQA	Customer Software Quality Assurance	+	+	x	Jos organisaatiolla ei ole SQA:ää, asiakkaan roolia voi korostaa.
MS	Marketing and Sales		x	x	
DSG	Documentation Support Group			x	
SSM	Software Subcontract Management			x	Käytännössä lienee jonkin SE:n hoidossa.

tarvitaan vasta, kun toiminta on laajentunut useampaan avainalueeseen (prosessialue) ja organisaation toimintaa halutaan kehittää esim. mittaamisen avulla.

Dynamic CMM:n kehittämisen tavoitteiksi oli asetettu, että pienen organisaation laatumallin tulee skaalautua organisaation ja sen toimintaympäristön mukaan. Laatumallin tulee olla helposti lähestyttävä sekä helppo omaksua ja ottaa käyttöön. Lisäksi tällaisen laatumallin tulisi ohjata prosessien arviointia sekä parantamista ja elää organisaation kasvun mukana. Laatumalli edellyttää, että organisaatiolla on järjestelmällinen tapa toimia. Laatumalli ei kerro, missä järjestyksessä asiat ja teot on tehtävä, mitä menettelyitä niissä käytetään ja millaisin kuvauksin tulokset esitetään. Sitä varten yrityksillä on systeemyön mallit: prosessit, menetelmistöt ja kuvausmallit. Pienen organisaation systeemyön mallille voisi hyvin asettaa samat vaatimukset kuin sen laatumallillekin, vai kuinka? Sen tulisi mukautua arkeen, auttaa valitsemaan oikeat tehtävät ja niihin tehokkaat menettelyt sekä ohjata tarpeellisen, ylläpidettävyyttä parantavan dokumentaation tuottamista työn yhteydessä.

Pienen organisaation systeemyön malli

Syksyllä 2004 kiinnostuin entistä vakavammin selvittämään pienten ohjelmistotuotanto-organisaatioiden toiminnan parantamiseen tähtäviä

tekijöitä. Olimme Heliassa jo kolmena keväänä kehittäneet opiskelijavoimin ohjelmistotuotannon prosessia pienkehittämisen ja oppilaitoksen tarpeisiin. Oppilaitoksen harjoitustyöthän ovat pääosin pienehköjä ohjelmistohankkeita. Kun Etelä-Amerikasta asti kantautui uutisia pienten organisaatioiden "tiikerin loikista" CMM:n kypsyytasolle 3 ja Ruotsissa oli meneillään laaja ESR-hanke laatumallin (Dynamic CMM) kehittämiseksi pienille organisaatioille, tunsimme olevamme oikealla tiellä. Niin käynnistyi PASMA-niminen tutkimus- ja kehittämishanke.

PASMA 1.0

Hankkeen tavoitteena on löytää tekijät, joilla on merkitystä ohjelmistotuotannon parantamiseksi. Tarkoituksena on hankkia tarvittavat tiedot ohjelmistotuotantotyötä ja sen parantamista palvelevan mallin kehittämiseksi ja tuotteistamiseksi nimenomaan pienen organisaation käyttöön. Haastatteleamalla niin pienten kuin suurtenkin organisaatioiden menetelmäkehityksestä vastaavia selvitin, millainen on hyvä ja pienen organisaation kaipaama toimintamalli. Kartoituksen tulosten perusteella ja jo kehittämäämme mallia auditoiden pyrimme löytämään toimintaa tehostavan ja yhtenäistävän prosessimallin ominaisuudet sekä vaatimukset tarpeenmukaiselle mallille.

PASMA 1.0 on systeemyön eli ohjelmistotuotannon prosessimalli, jota Heliassa kehitetään nyt jo viidennen kurssitoteutuksen aikana. Mallin kehit-

tämistyöhön on käytetty opiskelijatyönä ja ohjaavien opettajien panoksena yhteensä n. 13.000 tuntia, joka vastaa reilua kuutta henkilötyövuotta. Toisaalta mallin kehittäminen on vaatinut myös mallintamiseen ja erilaisiin ohjelmistotuotannon menetelmiin perehtymistä niin opiskelijoilta kuin ohjaajiltakin, joten todellisuudessa mallintamistyöhön paneutumiseen on kulunut arviolta 10 henkilötyövuotta.

PASMA 1.0 malli kattaa ohjelmiston kehittämisen tehtäväalueet sekä projektimaisen kehittämisen ohjaustehtävät. PASMA 1.0 on nyt pilotoitavana. Pilotoitavassa versiossa on 45 tehtäväkorttia. Tehtäväkortti on oikeastaan muistilista: se kertoo, miksi tehtävä tehdään, mitä tehtävän aloittaminen tarvitsee ja mitä sen tuloksena syntyy. Lisäksi se kertoo tehtävään liittyvät tarkistukset ja asiantuntijaroolit, joita tehtävä vaatii. Rooleina on käytetty Dynamic CMM:n roolinimiä ja vastuut on mitoitettu organisaation koon mukaan, kuten tau-lukossa 2 on kuvattu.

PASMAN kehittämisessä otimme alun perin huomioon SPICE kypsyysmallin piirteitä. SPICEn avulla voidaan arvioida ja parantaa yksittäistä prosessia, kun CMM:ssa mittaamisen ja arvioinnin kohteena on kokonainen prosessialue. Pyrimme osoittamaan, että pilotoitavan prosessimallin kyvykkyytaso riittää pienen organisaation tarpeisiin. Työn alla oleva seuraava PASMA-versio auditoidaan käyttäen Dynamic CMM:ia. Dynamic CMM vastaa perus-CMM:n tasoa kaksi.

Kehittäminen on ollut haasteellista varsinkin pienen organisaation puitteet muistaen. Todellinen haaste on kuitenkin mallin käyttöön sitoutuminen. Pilotoinnin kautta pyrimme selvittämään mallin käyttöönottoon liittyvät seikat. Niiden perusteella pyrimme parantamaan myös mallia, jotta sen käyttöön siirtäminen sujuisi mahdollisimman kitkattomasti. Pienet yritykset odottavat Helian sitoutuvan PASMA-mallin jatkuvaan kehittämiseen. Sitä, mitä jatkuva kehittäminen merkitsee, selvitan tarinoista, joita useat yritysten menetelmäkehittäjät kirjoittivat viime keväänä tutkimuksen käyttöön.

Odotuksia ja vaatimuksia mallille

Tutkimuksen tulosten analysointi on vielä osin kesken, mutta mm. seuraavat seikat nousevat aineistosta esiin:

Pienet organisaatiot kokevat yritysten suuret systeemyö- ja prosessimallit vaikeasti lähestyttävänä, raskaina ja liian tarkkoina ohjeina. Odotuksissa on enemmänkin muisti- tai tarkistuslistan omainen ohje, jota voi soveltaa tarpeen mukaan, neuvotellen. Muistilistalta tulisi voida poimia kunkin projektin tarvitsemat tehtävät siten, että niistä saa apua esim. projektin työsuunnitelman laatimiseen ja resursointiin.

Koska pienessä organisaatiossa yleensä koetaan olevan hyvä ja sujuva vuorovaikutustilanne asiakkaan ja osaavan henkilöstön välillä, dokumentaatio näyttää helposti jäävän puutteelliseksi. Ylläpidossa tilanne sitten kostahtuu. Usein asiakkaan ei ole kovin kiinnostunut muusta kuin ratkaisuisista, ellei toimittaja voi osoittaa dokumentoinnista selvää hyötyä myös asiakkaalle. Pienessä organisaatiossa henkilöstövaihtuvuus ei ole suuri, joten yleensä organisaation sisällä joustetaan ja asiaan vihkiytynyt saadaan tekemään tarvittavat muutostyöt. Toisaalta pieni organisaatio on henkilöstövaihtuvuuden suhteen myös haavoittuvampi kuin iso. Siksi juuri dokumentointiin halutaan kiinnittää huomiota: mitä vähintään pitää saada talteen tulevaa varten?

Pienessä organisaatiossa on tarvittaessa helppo kouluttaa kerralla vaikka koko henkilöstö uuden menettelyn pariin. Kouluttamisesta on kuitenkin matkaa pysyväksi käytännöksi, ellei uutta toimintatapaa tueta ja vaadita. Saman ovat suuretkin organisaatiot panneet merkille. Varsinkin kovan kiiren ja paineen alla kokeneet projektipäälliköt saattavat livetä käytännöistä ja luottaa enemmän kokemukseensa kuin uuteen menettelytapaan. Sen enempää suurissa kuin pienissä organisaatioissa ei tunnuta ottavan vaarin projektien loppuraportteista: kokemusten systemaattisesta jakamisesta prosessin parantamiseksi ja ohjeistuksen saattamiseksi helpommin sovellettavaan muotoon.

Kun suurissa organisaatioissa yrityksen johto jo vakaasti uskoo yhte-näisten toimintatapojen ja määrämuotoistamisen tuovan menestymisedel-lytyksiä, pieni luottaa henkilöstön ammattitai-toon ja avoimeen vuoro-puheluun.

Organisaation koossa tai henkilöstön maantieteellisessä sijoittumisessa tapahtuvat muutokset kuitenkin nostavat pienissäkin yrityksissä esiin mallien ja sovitujen toimintatapojen tarpeellisuuden, kun suora vuorovaikutus vähenee.

Testaus, laadunvarmistuksen käytännöt ja vaatimusmäärittäminen tuntuvat kaipaavan eniten ohjeistusta, oli sitten kyseessä pieni tai suuri organisaatio. Se ei sinänsä ihmetytä, sillä ohjelmiston hyvyys on osoitettavissa suhteessa vaatimuksiin, joita ohjelmistolle on asetettu. Vaikka näille tehtäväkokonaisuuksille on jo koko joukko hyviksi koestettuja käytäntöjä, niin haasteellisuus ilmenee alati muuttuvassa kehittäminen kohteessa. Oikeastaan vain pitkälle tuotteistettu, tuoteperhe-tyyppinen kehittäminen selvinnee näistä tehtävä-alueista kunniakkaasti voituaan vakiodia tuotteen kehittämisen ja ylläpidon.

Pienessä organisaatiossa henkilöstön rajallisen määrän vuoksi osa selvitys- ja vaatimusmäärittämisestä tehdään jo myyntityön yhteydessä. Työ vaatii vankkaa ammattitaitoa, jotta samaa työtä ei tarvitse tehdä uudelleen ohjelmiston määrittämisessä. Toisaalta tilannetta helpottaa, että pienessä organisaatiossa on rajalliset mahdollisuudet erilaisten toteutusympäristöjen hallintaan. Pienessä organisaatioissa on opittu hyödyntämään tehokkaasti kehittämisen välineitä. Asiakkaatkin seuraavat välinetoimittajien markkinointia tarkasti uskoen välineen kaikkivoipaisuuteen. Väline ei kuitenkaan eliminoi kunnollisen vaatimusmäärittämyksen tarvetta.

Malli toimii pienen organisaation referenssinä mm. tarjouskilpailussa. Mallin avulla organisaatio kertoo, miten se työnsä tekee, mitä käytäntöjä noudattaen. Isot asiakkaat ovat tottuneet suurten organisaatioiden kypsyystasoihin ja vaativat täsmällistä toimintatapaa myös pienemmiltä. Siksi toimintatapojen näkyväksi tekeminen mallin avulla on pienelle yritykselle entistä tärkeämpää.

Pienissä organisaatioissa projektimainen työtapa on yleistä, vaikkakaan projektien seurannan (hallinnollisiin) raportteihin ei ehkä panosteta yhtä paljon kuin isompien organisaatioiden mittavissa projekteissa. Paradoksaalista kyllä, juuri kiire ja seurannan puutteet liittyvät toisiinsa. Pienessä organisaatiossa tämä tiedostetaan, mutta samalla täsmällisemmän projektien hallinnan nähdään olevan suurin kulttuuriin liittyvä muutos.

Pienen ja suuren organisaation ohjelmistotuotannon tehtävät eivät juuri poikkea toisistaan. Ohjelmistotuotannon tehtävät ovat kohtalaisen vakiintuneet, vaikka tekemisen menetelmät, kuvaamisen kielet ja toteutusympäristöt muuttuvat. Malli on joka tapauksessa aina osoitus siitä, että organisaatio voi osoittaa kumppaneilleen toimivansa järjestelmällisesti ja yhtenäisesti (yhdenmukaisesti) ohjelmistokehitystyössä. Prosessien parantaminenkin onnistuu vasta, kun ne on tehty näkyviksi.

Lähteet

Jacobson, I. Booch, G. Rumbaugh, J. 2000. The Unified Software Development Process

Orci, T. Laryd, A. 2000. CMM for Small Organizations, Level 2

Pollice, G. Augustine, L. Lowe, C. Madhur, J.

2003. Software Development for Small Teams. A RUP-Centric Approach.

PASMA-hankkeen tutkimustulokset ja pilotointikokemukset

Hyvä tietotekniikan liiton jäsen!

Päivitä tietosi

Voit päivittää jäsentietosi verkkosivuillamme www.tt-tori.fi. Tietojen päivittämiseen tarvitset käyttäjätunnuksen (= jäsen-numerosi, merkitty jäsenlehtiin) ja salasanasasi (= posti-numerosi). Jos olet muuttanut salasanasasi tai kirjautuminen ei muutoin onnistu, voit lähettää tunnusten tarkistuspyynnön osoitteella jasenasiat@ttlry.fi.

Toivomme sinun erityisesti varmistavan, että sähköpostiosoitteesi jäsentiedoissa on oikea.



Henkilökohtaisempaa palvelua - Sinun eduksesi

Tietotekniikan liitto jäsenyhdistyksineen, osaamisyhteisöineen ja kerhoineen haluaa palvella jäseniään henkilökohtaisemmin ja paremmin, tarjota tietoa juuri Sinua kiinnostavista aiheista. Palvelun parantamiseksi uusimme verkkopalvelumme helmikuun 2006 lopussa. Uuden palvelun avauduttua voit auttaa meitä räätälöimään palvelumme juuri Sinulle.

Päivität vain tiedot itseäsi kiinnostavista aiheista ja saat tietoa juuri niistä. Voit päivittää valintasi aina halutessasi. Tietoja ei anneta ulkopuolisille tahoille vaan niitä käytetään ainoastaan TTL:n ja sen piirissä toimivien yhteisöjen tarkoituksiin.

Tietotekniikan liitto ry

Lars Sonckin kaari 12	www.tt-tori.fi	jasenasiat@ttlry.fi
02600 Espoo	etunimi.	p. 020 741 9898
	sukunimi@ttlry.fi	f. 020 741 9889