

Pienen organisaation ohjelmistotuotantoprosessin parantaminen

Anne Valsta

Helsinki 9.11.2004
Seminaari: Ohjelmistotekniikan laatumallit

Tietojenkäsittelytieteen laitos
HELSINGIN YLIOPISTO

Anne Valsta

Pienen organisaation ohjelmistotuotantoprosessin parantaminen

HELSINGIN YLIOPISTO

Matemaattis-luonnontieteellinen tiedekunta

Tietojenkäsittelytieteen laitos

Ohjelmistotekniikan laatumallit, seminaarityö

9.11.2004, 15 sivua

Tiivistelmä

Pienen, kuten nykypäivinä minkä tahansa ohjelmistotuotanto-organisaation haasteena on olla kilpailukykyinen kansainvälisillä markkinoilla. Isot organisaatiot mittauttavat säännöllisin väliajoin kyvykkyytensä standardin asemaan nousseiden kypsyyksimallien avulla ja osoittavat siten uskottavuutensa markkinoille. Samalla ne myös edesauttavat mallien kehittymistä.

Pienen organisaation on lähes mahdotonta panostaa ohjelmistotuotannon prosessimallien kehittämiseen. Toisaalta organisaation koko antaa 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 toimivalle prosessimallille kasvaa.

Viime vuosina kiinnostus pienen organisaation erityispiirteisiin on virinnyt. Kypsyyksimallilta odotetaan, että se huomio organisaation elinkaaren: pienestä isommaksi kasvun. Dynamic CMM auttaa pientä kypsymään SW-CMM:n tason 3 vaatimukseen.

Avainsanat

pieni organisaatio, ohjelmistoprosessi, ohjelmistoprojekti, kypsyyksimalli

Pienen organisaation ohjelmistotuotantoprosessin parantaminen

Määritelmiä

Käsitteet

Kypsyysmalli

on prosessien kypsyyden/ kyvykkyyden arviointi-/mittausmalli, laatumalli.

Projekti

on aikataulutettu ja resursoitu tehtävän ilmentymä, jossa prosessia noudatetaan. Projekti kattaa myös joukon ohjaustehtäviä, joilla prosessin edistymistä seurataan.

Prosessi

on tehtäväketju, joka noudattaa mallia ja tuottaa jonkin tuloksen.
"A systematic approach to the creation of a product or the accomplishment of some task" (Osterweil).
"Processes are collections of software-related activities"
(Fenton, Pfleeger 1997: 74).

Systeemyö-, prosessimalli

kuvaava, miten ohjelmistotuotanto etenee ja mitä menetelmiä työssä käytetään sekä miten tuote dokumentoidaan (kuvausmalli) mm. Rational Unified Process (RUP), Rapid Application Development (RAD), erilaiset vaihejakomallit, Personal Software Process (PSP), Team Software Process (TSP), Agile Methods ja Extreme Programming (XP).

Lyhenteet ja standardit

CMM Capability Maturity Model

CMMI Capability Maturity Model IntegrationSM

ISO 12207 Software Life-Cycle Processes

ISO 15504 SPICE Software Process Improvement and Capability dEtermination

SPI Software Process Improvement

SW-CMM Software Capability Maturity Model

Pienen organisaation ohjelmistotuotantoprosessin parantaminen

Sisältö

1	Johdanto	1
2	Pieni organisaatio	2
2.1	Ohjelmistotuotanto prosessina	3
2.2	Ohjelmiston kehittäminen projektina	5
3	Ohjelmistotuotannon kypsyysmalli	6
3.1	Dynamic CMM for Small Organisations	6
3.2	IDEAL SM Model	9
4	Prosessin parantaminen pienessä organisaatiossa	10
4.1	Kokemusten hankkiminen	10
4.2	Link	11
5	Yhteenveto	14
	Lähteet	15

1 Johdanto

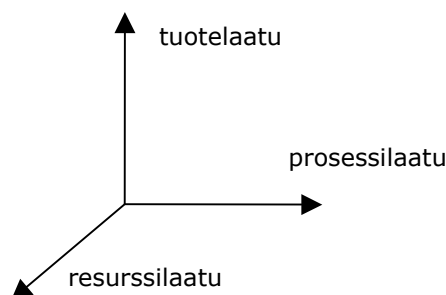
Ohjelmistotuotannon laatu on puhuttanut alan yhteisöjä jo pitkään. On kehitetty käytäntöjä, joilla varmistutaan, että tekeillä oleva ohjelmistotuote vastaa tarkoitustaan eli tilaajan vaatimuksia ja on aikaansaatu tehokkailla, itseään korjaavilla toimilla. On rakennettu, tuotteistettu ja standardoitu

- systeemityön elinkaarimalleja
- ohjelmistotuotannon prosessimalleja (esim. Rational Unified Process)
- ohjelmistotuotannon prosessien laatu- eli kypsyyksimalleja (esim. SW-CMM, SPICE).

Mallien kirjo ja niiden periaatteelliset sekä sisällölliset yhtäläisyydet tuottavat niitä valitseville päänvaivaa. Ohjelmistotuotantoprosessin parantaminen (SPI, Software Process Improvement) on niin mallien kehittäjien kuin ohjelmistotuotantotyötä tekevienkin kannalta tärkeä menettely toimivien käytäntöjen tuottamiseksi ja siten kilpailukyvyyn säilyttämiseksi.

Markkinoilla menestyy, jos pystyy nopeisiin toimitusaikoihin ja asiakasta tyydyttävien ohjelmistotuotteiden kehittämiseen. Vaatimuksiin pääseminen edellyttää kokonaisvaltaista panostamista niin tuote-, prosessi- kuin resurssilaatuun.

Kuvassa 1 tuotteiden laatua edustavat esim. käytettävyys ja ylläpidettävyys; toiminnan ja toimintatapojen eli prosessien laatua katselmoinnit ja tehokkuuden mittaaminen; resurssien laatua mm. henkilöstön osaaminen ja ajanmukaisten työvälineiden käyttö kehittämisessä. Näistä prosessilaatu on mielenkiintoinen, sillä sitä parantamalla voidaan saavuttaa edistymistä myös tuotteiden ja resurssien laadussa eli kokonaistehokkuuden kasvua. Toisaalta vain yhteen osatekijään satsaaminen saattaa vääristää kokonaisuuden.



Kuva 1. Laadun osatekijät

Kansainvälisen kaupan vapautuminen tarjoaa enenevässä määrin mahdollisuuksia myös pienille yrityksille ja organisaatioille. Kilpailuilla markkinoilla on osoittauttava uskottavaksi toimittajaksi. Uskottavuutta rakentaa pitkälti kansainvälisesti hyväksytty ohjelmistotuotantoprosessi, laatumalli ohjelmistotuotantotoiminnalle. (Guerrero, Eterovic.)

Pienellä organisaatiolla ei useinkaan ole kehittämiseen erikoistunutta henkilöstöä, joten muutosten toteuttamiseen tarvittava ammattitaito ja kokemus puuttuvat. Lisäksi pk-yritykset ovat joutuneet sopeutumaan toimintaympäristön ja asiakkaiden vaatimuksiin sekä niiden muutoksiin. (Kuitunen et. al.: 1.)

Ohjelmistotuotannon prosessimallit tähtäävät varsin suurten ja monimutkaisten ohjelmistojen kehittämishankkeiden läpiviennin onnistumiseen. Toiminnan kehittämisen mallit, menettelytapaohjeet ja parhaat käytännöt perustuvat suurten, kansainvälisten ja monialaisten yhtiöiden kokemuksiin ja niistä oppimiseen. Malleissa tarjottuja keinoja pidetään liian järeinä pienyrityksen käyttötarkoitukseen, eivätkä ne siten ole kovin käytännöllisiä. (Guerrero, Eterovic sekä Kuitunen et. al.: 11.)

Ketterät menettelyt on alunperin kehitetty pieniä organisaatiota (tiimit) ja projekteja varten. Ajatus perustuu hiljaiseen tietoon (tacit), motivoituneen henkilöstön ammattitaitoon ja säännölliseen viestintään. Ketteryys tarkoittaa, että ohjelmiston kehittäminen ja käyttöönotto tapahtuu nopeissa sykleissä ja yhteistyö asiakkaan kanssa on kiinteää. Ketterät menettelyt ovat enemmän ihmisorientoituneita kuin prosessorientoituneita. Menetelmät sopivat parhaiten pienille organisaatioille, jotka kehittävät innovatiivisia ohjelmistotuotteita. (Kalermo, Rissanen: 9, 87.)

Tarkastelen seuraavassa pienen organisaation piirteitä ja toimintatapoja, ohjelmistokehittämistä sekä projektia yhtenä vaihtoehtoisena toiminnan organisointimuotona ohjelmistotuotannossa. Sen jälkeen, luvussa kolme, esittelen pienen organisaation puitteisiin soveltuvan kypsyysmallin sekä ohjelmistotuotantoprosessien parantamistyöhön tarkoitettua menettelyä. Luvussa neljä kerron esimerkin pienestä organisaatiosta, jossa prosessien parantamiseen on ryhdytty ja hankittu onnistuneita kokemuksia. Lopuksi kokoan keskeiset havainnot pienen organisaation ohjelmistoprosessin parantamiseen tähtäävästä työstä.

2 Pieni organisaatio

Suomessa pk-yritykseksi tulkitaan alle 250 henkilöä työllistävä organisaatio (Kuitunen et. al.: 1). Pieni organisaatio voi myös olla vain muutaman henkilön paja, jossa toiminta on varsin tehokasta ilman, että sillä on ainakaan nimettyä laatumallia toimintansa taustavaikuttajana. Pienen ja nuoren yrityksen vahvuutena pidetäänkin sen joustavuutta.

Pienet yritykset toimivat usein alihankkijoina suurempien verkostoissa. Globaaleilla markkinoilla toimivat päähankkijat edellyttävät alihankkijoiltaan kykyä sopeutua kasvavan liiketoiminnan tarpeisiin. Pk-yritykset joutuvat tekemään monia strategisia valintoja varsinkin, kun niiden toimintaympäristössä tapahtuu muutoksia, jotka koskevat niiden kasvumahdollisuuksia. Tällaisia muutoksia ovat mm. tuotantomallien muuttuminen ja toiminnan kansainvälistyminen. Muutokset edellyttävät pk-yritykseltä osaamisen ja yhteistyötapojen kehittämistä. Siten oman toiminnan kehittämisen kannalta verkostosuhteiden hyödyntäminen ja aseman vahvistaminen tulevat yhä tärkeämmäksi. (Kuitunen et al.: 1.)

Pienessä organisaatiossa ajatellaan, ettei SW-CMM sovellu sen käyttöön. Näkemystä halutaan pitää myös syynä epäonnistumiseen prosessien parantamisessa. SW-CMM on massiivinen malli, joka suunniteltiin USA:n liittohallituksen järjestelmähankintojen menettelyiden kehittämiseen ja käytäntöjen parantamiseen. Sen kattavuus takaa myös sen soveltamisen moninaiseen toimintaympäristöön. SW-CMM:ia käyttäville organisaatioille on ominaista, että

- niissä työskentelee satoja henkilöitä useissa eri yksiköissä ympäri maailmaa
- niiden projektit ovat suuria kaikilla mitoilla mitattuna: miljoonien dollarien budjetti, kesto useita vuosia, tusinoittain projektityöntekijöitä
- niiden asiakkaina on maailman suurimmat ohjelmistohankkijat
- niissä otetaan käyttöön uusin teknologia, ohjelmistot ja metodit
- niissä kehitetään upotettuja, ajantasa- ja sähköisen asiointin järjestelmiä puolustus- ja ilmavoimien kaltaisilla vaativille asiakkaille
- niissä noudatetaan asiakkaan vaatimuksesta kansainvälisiä, hyväksytyjä standardeja.

(Guerrero, Eterovic sekä Kruchten P. esipuheessaan kirjassa Pollice et al.: Software Development for Small Teams.)

Pienessä organisaatiossa kaikki on toisin. Silti niille on yhtä tärkeää kuin isoillekin tehdä tuotteensa hyvin ja tehokkaasti. Toimintamekanismit eivät välttämättä poikkea toisistaan, mutta tietyt toimintaan vaikuttavat tekijät, kuten henkilöstön, tuotteiden, tuoteasiakkaiden ja tuoteprojektien määrä sekä kehittämiseen käytettävien resurssien määrä ja laatu, sitäkin enemmän. Yhtäältä jokainen organisaatio on ollut elinkaarensa alkuvaiheessa mitä todennäköisimmin pieni. Jotkin tekijät ovat mahdollistaneet sen kasvua. Kasvu tuo mukanaan uusia haasteita organisaation toimintatapoihin. Prosessit olisi syytä tehdä näkyviksi jo toiminnan alkuvaiheessa, sillä toiminnan kasvaessa niihin tulee väistämättä muutoksia ja lopulta ne eivät enää ole yhteneviä kaikille organisaatiossa. Yhdenmukainen prosessi ja yhdenmukaiset käytännöt parantavat organisaation kykyä toimia tehokkaasti ja helpottavat organisaation johtamista.

2.1 Ohjelmistotuotanto prosessina

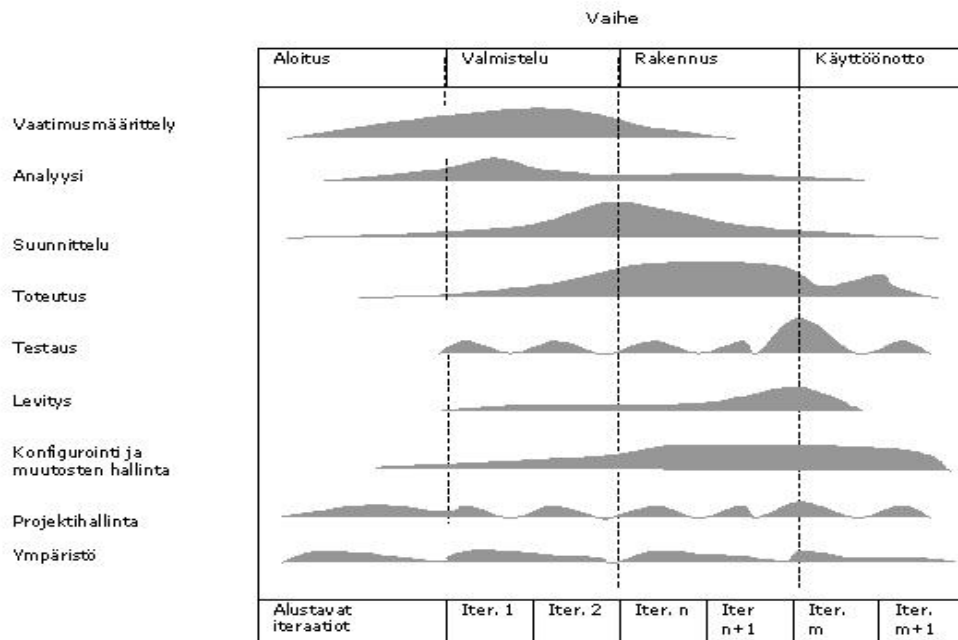
Käsitteet menetelmä (method) ja prosessi (process) eivät ole yksiselitteisesti määriteltyjä, eivätkä järjestelmällisesti käytettyjä (Kalermo, Rissanen: 44-45). Se on seikka, joka entisestään hankaloittaa pienenkin organisaation paneutumista malleihin ja hyvän prosessi- tai kypsyyssmallin valintaa. Malli voi olla liian raskas otettavaksi käyttöön tai edes tutustuttavaksi.

Lähtökohta toiminnan parantamisella on, että ohjelmistotuotantoa tarkastellaan prosessina/ prosesseina. Prosessin irrottaminen sen myötä syntyvästä tuotteesta ja tuotteeseen liittyvistä laatuvaatimuksista on haasteellista. Joidenkin ominaisuuksien aikaansaaminen tuotteeseen edellyttää tarkoituksenmukaisia tekoja eli toimenpiteitä prosessissa tai omaa (osa)prosessia, jota muuten ei tarvittaisi työssä. Tietyt työt riippuvat siis kohteesta, jota työstetään. Voisi sanoa, että kypsyyssmallit pyrkivät kattamaan kaikki mahdolliset toimenpiteet kaikki mahdolliset piirteet ja vaatimukset täyttävän tuotteen valmistamiseksi. On kuitenkin tilanteita ja organisaatioita, jotka tekevät yksinkertaisia ja vain rajattua tarkoitusta palvelevia tuotteita. Niiden tekemiseen ei tarvita kaiken kattavia työmalleja.

Ohjelmistotuotanto sisältää useita prosesseja, joista osa on tuotteen määrittelyyn, suunnitteluun ja toteuttamiseen tai kunnossapitoon liittyviä, osa valmistuksen ohjaukseen, kokoonpanon hallintaan ja jakeluun liittyviä. Prosessi alkaa aina asiakkaan vaatimuksista (lähtökohdat prosessille) ja päättyy tuotosten asiakkaalle toimittamiseen.

Kun organisaation koko sekä kehitettävän ohjelmiston koko ja monimutkaisuus kasvavat, tarvitaan sääntöjä. Systeemyön eli ohjelmistotuotannon malli on myös säännöstö, joka pyrkii yhdenmukaistamaan prosessin, joka valmistaa ohjelmistoja. Mallia voi noudattaa kirjaimellisesti tai toimia sen hengessä, ja silti epäonnistua. Malli on aina sovitettava kehittämistilanteeseen, kehittämisen kohteen vaatimukseen, sillä kaikki ohjelmistot eivät ole samanlaisia, eikä niitä siten tehdä aina samoja temppuja käyttäen. Mallin hengen noudattaminen vaatii lujaa ammattitaitoa: pitää osata valita kehittämistyöhön juuri oikeat vaiheet, tehtävät, menetelmät ja välineet sekä ihmiset. (Pollice et al.: 10-11, 19.)

RUP (Rational Unified Process) on iteratiivinen ohjelmistotuotantoprosessin malli. Siinä on perinteinen lineaarinen (vaihejako) malli istutettuna vaiheisiin, joissa ohjelmiston valmiusaste kasvaa. Kuvassa 2 kumpareet ilmentävät ko. tehtäväkonaisuuden painopisteen kehittämistyön aikana. Iteratiivinen lähestymistapa suosii käytötapausvetoista kehittämistä ja jopa ketterät menettelyt istuvat mallin vaiheisiin. (Pollice et al.: 237-248.)



Kuva 2. RUP-malli (Rational Unified Process Fundamentals Student Manual 2001: 2.12), suom. Rational Finland

Prosessimallin sokea noudattaminen johtaa varmuudella kaaokseen. Prosessimalli on aina vain malli, kehikko, kartta, jonka avulla organisaation tai projektin on päätettävä, mikä on sopiva käytäntö sen kulloisiinkin tarpeisiin. Ihmisillä tuntuu olevan taipumus koota liikaa prosessiin ja luottaa sitten sen kykyyn tuottaa oikea tulos. Prosessimallin sovittaminen omaan tarpeeseen ja tarkoitukseen ei ole vain tarvittavan setin poimimista, vaan on osattava valita yhteenkuuluvat ja toisistaan riippuvat prosessit, käytännöt, tehtävät oikeiden tulosten saavuttamiseksi. Haasteen valinnoille asettaa alati muuttuva kohde: samaa työtä tai tulosta ei tehtäne uudelleen. (Pollice et al.: 31.)

On syytä välttää valitsemasta mallia vain mallin itsensä (tunnettuus, laadukkuus, kattavuus) vuoksi. Mallin ja siis prosessien tulee istua organisaation osaamistason, resursseihin ja projektien/ organisaation toimintaympäristöön, käytäntöön ja kulttuuriin sekä tuotevalikoimaan. Montaa organisaatiota auttaisi nk. minimin nimeäminen: vähintään nämä toimenpiteet, kuvaukset ja lopputulokset tulee tehdä jokaisen tuotteen valmistamisen aikana. Mallit ovat skaalautuvia, mutta skaalaus on itse tehtävä. (Pollice et al.: 28-46.)

2.2 Ohjelmiston kehittäminen projektina

Projekti on varsin tavallinen työmuoto ohjelmistotuotteen valmistamisessa. Projekti tekee määräajassa tietyn tehtävän ja saa aikaan nimetyt tulokset, joille on asetettu tavoitteet. Myös projektin tekemiselle asetetaan tavoitteet, jotka usein ovat peräisin laatumallista, jolla ohjausprosessin kyvykkyys osoitetaan.

Projektilla on oma organisaatio, joka on saanut projektin ajaksi valtuuden projektin tehtävän tekemiseen. Projektiorganisaatiolla on oma ohjausryhmä, joka tekee projektia koskevat päätökset aikataulun ja budjetin puitteissa.

Humphreyn kehittämä TSP (Team Software Process) on ryhmätyöprosessi, jonka tavoitteet on johdettu kypsyysmallista (CMM). Vuorovaikutuksella on tärkeä sija ryhmätyöprosessissa so. tiimin/ ryhmän toiminnan kehittäminen, osaamisen jakaminen ryhmän jäsenten kesken. Ryhmä koostuu rooleista: projektipäällikkö, suunnittelupäällikkö, ohjelmistopäällikkö, laatu-/prosessipäällikkö ja tukipalvelupäällikkö. Roolien vastuut voidaan nähdä projektiryhmän vastuina, kun projektiryhmän koko on 4-6 henkilöä. Roolien tehtävänä on myös siirtää projektissa opittu linjaorganisaatioon ja kehittää siten oman työyhteisön toimintatapaa.

TSP koostuu joukosta syklejä, joiden myötä ohjelmistotuote syntyy. Sykleille on mallissa valmiit skriptit eli askeleet, jotka näyttävät pitkälti perinteiseltä vaihejoalta. Jokaisen syklin päättää jälkikäsittelevä vaihe (postmortem), jossa syklin mittatiedot kerätään ja analysoidaan ennen seuraavan syklin aloittamista tai kehittämistyön päättymistä.

Jälkikäsittelevä on tärkeä prosessin parantamisen väline. Ryhmäprosessissa arvioidaan ryhmän toiminnan kehittymistä kohti kypsää toimintaa ja roolissa onnistumista. Mittaamista varten TSP:ssa on joukko lomakkeita, jotka malli odottaa täytettävän. Se voi olla liikaa monessa hektisessä projektissa.

3 Ohjelmistotuotannon kypsyysmalli

Yleisesti tunnetut, de facto –aseman saavuttaneet ohjelmistotuotannon kypsyysmallit, kuten CMM, eivät sellaisenaan sovellu erilaisiin organisaatioihin (Lindvall, Rus). Mittavan mallin käyttöönottoa rajoittaa myös se, että pienellä organisaatiolla ei ole tarpeeksi resursseja pitkäntähtäimen kehittämistyöhön ulkopuolisen asiantuntijan avulla, eikä se pysty liioin osoittamaan omia asiantuntijoitaankaan tähän työhön. Niinpä mallissa tulisi olla itsearviointin ja soveltamisen ohjeistus. Mallissa tulisi huomioida myös organisaation elämänskaari, kasvu ja elinikä. (Orci, Laryd.)

Jotta jokaisen organisaation ei tarvitsisi itse keksiä prosessimallia, on mallille asetettava tiettyjä vaatimuksia

- malli skaalautuu organisaation ja sen toimintaympäristön mukaan
- malliin on helppo perehtyä ja omaksua se
- malli ohjaa prosessien arviointia ja parantamista
- malli elää organisaation kasvun mukana. (Orci, Laryd.)

3.1 Dynamic CMM for Small Organisations

Uumajan yliopistossa on kehitetty SW-CMM:iin pohjautuva pienelle organisaatiolle tarkoitettu prosessien kypsyysmalli: Dynamic CMM for Small Organisations (Orci 2000. Technical Report). Dynamic tarkoittaa joustavuutta, mallin skaalautuvuutta, organisaation kokotekijän suhteen. Kehittäjien tavoitteena on ollut, että mallista saa helposti yleiskuvan sekä käsityksen tarvittavien resurssien suuruusluokasta, rooleista, vastuista, tehtävistä ja dokumentaatiosta. Alkuperäistä CMM:ää on tulkittu, karsittu, jäsennetty uudelleen ja esitetty osin kuvaamisen keinoin. (Orci, Laryd.)

Mallin kehittämisen lähtökohtana on pidetty eo. listaa mallille asetettavista vaatimuksista. Malli on johdettu CMM:stä sen vuoksi, että CMM on yleisesti tunnettu ja laajalti käytetty kypsyysmalli. CMM oli myös Dynamic CMM –mallin kehittäjille tutuin. Dynamic CMM malli on rajattu tasolle 2. Kasvettuaan tai saavutettuaan tason 2 organisaatio voi siirtyä käyttämään SW-CMM:ia. (Orci, Laryd.)

Dynamic CMM on tarkoitettu lähinnä organisaation oman henkilöstön käyttöön ja organisaation sisäiseen arviointiin. Oletuksena on, että henkilöstö omaa jonkin verran kokemusta prosessin parantamisesta ja johtamisen keinoista. Vuoteen 2000 mennessä malli oli ollut käytössä yhdessä kohteessa, josta ei vielä tässä vaiheessa ole raportoitu. Kahden muun kokeilusta on neuvoteltu. Mallin kehittämisen on rahoittanut NUTEK, the Swedish National Board for Industrial and Technical Development. (Orci, Laryd.)

Organisaatiota, jonka henkilöstön koko on 1-50 henkilöä (EU:n luokitus), pidetään pienenä. Orci ja Laryd jakavat pienet organisaatiot vielä tarkemmin taulukon 1 mukaisesti. Organisaatiojakoa käytetään valittaessa Dynamic CMM aloituskohta.

Taulukko 1. Aloituskohta organisaation koon perusteella (Orci, Laryd).

tuote- määrä	henkilömäärä		
	1-2	3-15	> 15
1	XXS	XS	S
2-5	-	XS	S
> 5	-	S	S

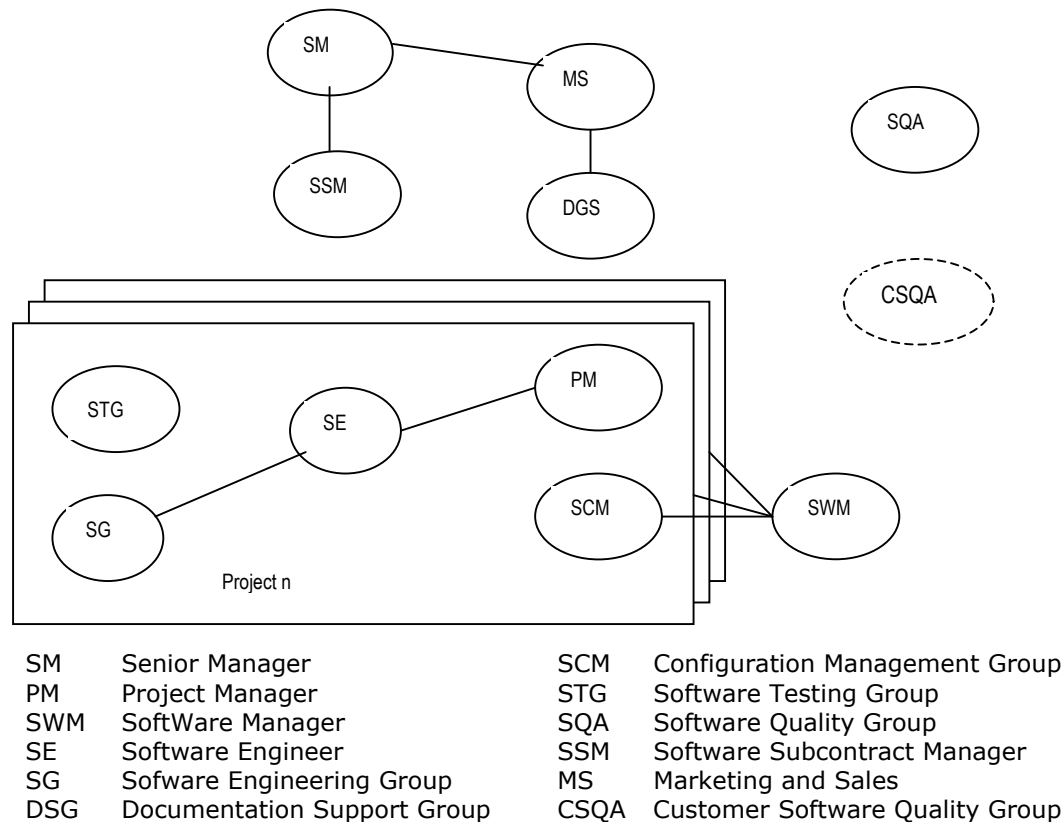
Orci ja Laryd perustelevat jakoa sillä, että EU-luokituksen mukaiseen joukkoon mahtuu mitä erilaisimpia organisaatioita. XXS-organisaatiossa on tunnetusti vahva vuorovaikutussuhde henkilöiden kesken. Heti, kun organisaation koko kasvaa kolmeen, vuorovaikutussuhde mutkistuu ja väärinkäsitysten riski kasvaa. Silti XXS- ja XS -organisaation johtaminen on vielä monin verroin helpompaa kuin liki 50-henkisen organisaation. Lähestyttäessä pienen organisaation henkilöstön ylärajaa on ilmeistä, että organisaatiolla on jo useita tuotteita ja versioita niistä markkinoilla sekä useita asiakkaita, joille tuotteita on räätälöity. (Orci, Laryd.)

Kun SW-CMM esittelee yli 25 organisatorista roolia jo tasolla 2, on selvää ettei se istu pienen organisaation puitteisiin mukauttamatta ja yhdistämättä rooleja. Silti voi olla, että pienessä organisaatiossa ei edes ole kyseisiä rooleja (yl. hallinnointiin liittyvät tukiroolit). Rooleilla on sisäisiä riippuvuuksia, vastuita ja tehtäviä, joita voidaan yhdistää samalle henkilölle. Dynamic CMM:n kehittäjät tutkivat mahdollisia rooliristiriitoja, ja ainoastaan laadunvalvonta (SQA, Software Quality Assurance) on rooli, jota ei saa sotkea sen enempää johtamisen kuin ohjelmiston kehittämisen rooleihin. Roolit voidaan jakaa kolmeen ryhmään

- itsestään selvät, esim. organisaation johtaja
- muodolliset (funktionaalinen tehtävä ja vastuu organisaatiossa), esim. ohjelmistokehittäjä, johon voidaan liittää muita vastuita, kuten dokumenttien hallinta tai muutoksen hallinta
- koon kannalta epäolennaiset, esim. alihankintojen hallinta. (Orci, Laryd.)

Dynamic CMM lähestyy organisaation prosesseja roolimallin avulla. Siinä nimetään XXS-, XS- ja S -organisaatioiden roolit ja niiden vastuut henkilömäärään suhteutettuna. Mitä pienempi organisaatio sitä enemmän rooleja on yhdisteltävä yksittäisten henkilöiden kesken. Toisaalta ohjelmistokehitys on näissä organisaatioissa varsin selkeää, yksinkertaista, kerralla yhden tuotteen tai sen osasovelluksen tekemistä: dokumenttien ja tuotteenhallinnan näkökulmat yleensä vielä puuttuvat. Rooleja, esim. testaus ja laadunvarmistus, voi myös ulkoistaa. Asiakas pitää sitoa kehittämiseen vahvemmin, kun itsellä ei ole tarvittavia resursseja tai osaamista (vrt. ketterät menetelmät). Laadunvarmistus voidaan mm. hankkia asiakkaalta, jolla on siihen oma organisaationsa. Laadunvarmistusryhmää tarvitaan vasta, kun useampi kuin yksi avainprosessialue on määritelty, kuvattu ja käytössä kuvatuskaltaisena (vrt. auditointi). (Orci, Laryd.)

Kuvassa 3 on esitetty S-organisaation roolikartta esimerkkinä Dynamic CMM:n kuvauksista ja aloituskohdasta. Kuvassa on yksittäisten projektien roolit ja organisaatiossa tarvittavat muut roolit, kun organisaatiolla on useita rinnakkaisia tuoteprojekteja ja useita asiakkailta sekä mahdollisia kumppaneita.



Kuva 3. Dynamic CMM:n S-organisaation roolit (Laryd, Orci.)

Dynamic CMM:n mukainen prosessien parantaminen organisaatiossa on pitkän tähtäimen työrupeama. Kyseessä on enemmänkin jatkuva, iteratiivinen parantaminen, jonka aloittaminenkin on harkittava tarkasti. Prosessin parantamista varten kannattaa perustaa projekti, jolla on ohjausryhmä, selkeästi määritellyt tavoitteet (kullekin iteraatiolle), aikataulu ja tarkistuspisteet. Ohjausryhmä nimeää projekti-ryhmät. Lisäksi tarvitaan nk. muutosagentti, mielipidejohtaja, jolla on asiantunte- musta niin SPI:sta kuin ohjelmistotuotannostakin. Tällainen henkilö tai jopa ryhmä henkilöitä (SEPG Software Engineering Process Group) panostaa yhteistyöhön projekti-ryhmien välillä, tukee ja koordinoi päätöksiä sekä motivoi prosessin mukai- seen toimintaan. (Orci, Laryd.)

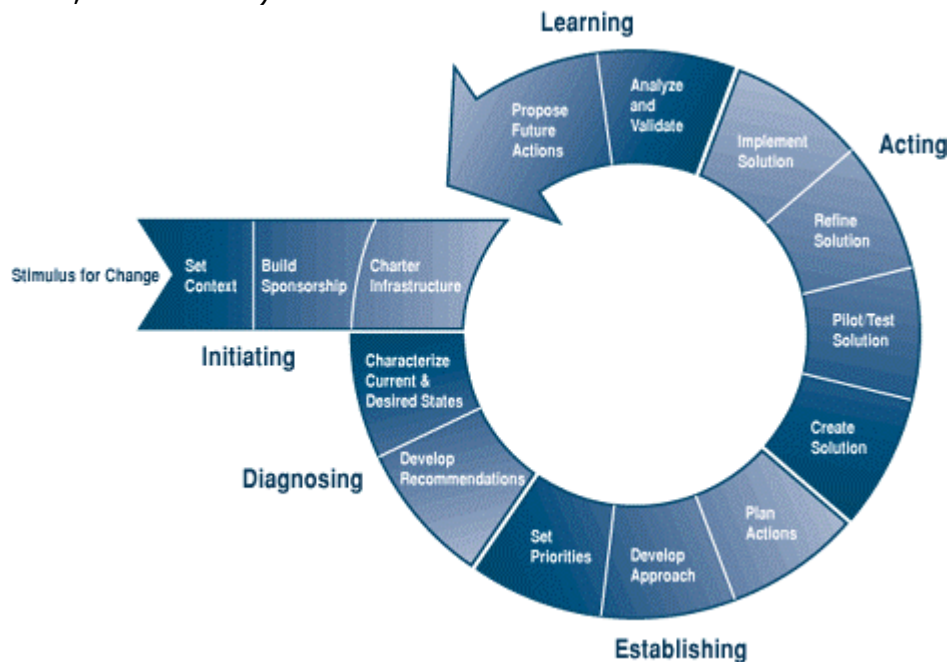
Projekti voi keskittyä yhteen avainprosessialueeseen (KPA, Key Process Area), pi- lottiin. Ensitöiksi, ellei prosessia ole vielä kuvattu, se tulee määritellä, dokumen- toida, kokeilla (esim. simulaation avulla) ja arvioida. Prosessissa saattaa ilmetä välittömästi korjattavaa, joten korjaamisen ja dokumentoinnin jälkeen se on jäl- leen koestettava ja arvioitava uusi tilanne. Näin jatketaan, kunnes prosessi on nk. säädöissään. Kierrokset noudattava plan-do-evaluate -sykliä (vrt. IDEALSM Model). (Orci, Laryd.)

3.2 IDEALSM Model

Ohjelmistotuotannon prosessien parantamisella näyttää olevan heijastusvaikutuksia laajalle alueelle organisaatiossa. Uudet menettelyt, työkalut ja prosessien rajapinnat mm. tukiprosesseihin, vaativat yhteistyötä näiden prosessien omistajien kanssa. Kokonaisuuden ja riskien hallinta on johtamisen tärkeimpiä haasteita. SEI (Software Engineering Institute, Carnegie Mellon University) on kehittänyt vuonna 1996 CMM:n rinnalle ohjeistuksen prosessien parantamistyölle: IDEALSM Model (Börjesson, Mathiassen).

IDEALSM Model tarjoaa käyttökelpoisen ja helposti ymmärrettävän lähestymistavan prosessien parantamiseen. Seuraamalla kuvassa 4 esitettyjä mallin vaiheita mahdollisuus toiminnan kokonaisvaltaisen paranemisen onnistumiseen kasvaa. (Gremba, Myers.)

Mallia on käytetty useassa SPI-hankkeessa, mm. Ericsson ja Link (ks. luku 4.2) (Börjesson, Mathiassen).



Kuva 4. IDEAL-malli (<http://sei.cmu.edu/ideal>)

Aloitusvaiheen (Initiating) lähtökohtana on liiketoiminnan vaatimusten mukainen muutostarve, johon organisaation johto sitoutuu. Se on myös paras onnistumisen tae. Aloitusvaiheessa muutostarve määritellään tarkasti,

- mihin organisaation liiketoimintastrategiaan muutos perustuu
- mitkä liiketoimintatavoitteet ja kohteet realisoituvat muutoksen myötä
- mitkä edut (ROI, kypsyystaso, asenneilmasto) saavutetaan?

Muutospolun riskien eliminoimiseksi aloitusvaiheessa nimetään tukioorganisaatio. Lisäksi nimetään muutosprosessin ohjausryhmä, joka voi olla myös pysyvä organisaatio. Ohjausryhmän koko riippuu niin organisaation koosta kuin kehittämistyön laajuudesta ja monimutkaisuudesta. Jos aloitusvaiheen tehtävät hoidetaan hutiloiden tai jätetään vaihe kesken ennen siirtymistä havainnointivaiheeseen, on toden-

näköistä, että hanke ajautuu karille ennen pitkää ja siihen satsatut resurssit ovat olleet turhia. (Gremba, Myers.)

Havainnointivaiheen (Diagnosing) keskeisin tavoite on hahmottaa ja mallintaa organisaation nykyinen tila ja johtaa haluttu tavoitetila. Apuna voidaan käyttää SW-CMM:ia. (Gremba, Myers.)

Valmistusvaiheen (Establishing) tavoitteena on laatia yksityiskohtainen suunnitelma, jossa parantamisteot on priorisoitu suositusten mukaisesti ja jossa on otettu huomioon rajalliset resurssit sekä ulkoiset tekijät. Niiden lisäksi uudet menettelyt ja teknologia, osaaminen ja organisaatiokulttuuri tulee huomioida suunnittelussa. Valmistusvaiheessa on suunniteltava myös prosessien käyttöönotto ja uuden strategian jalkauttaminen. (Gremba, Myers.)

Käynnistysvaiheessa (Acting) tarvitaan yleensä enemmän kalenteriaikaa ja resursseja kuin edeltäneissä vaiheissa yhteensä. Nyt kuvataan "paras arvaus" ratkaisuksi huomioiden aiempien vaiheiden vaatimukset ja tulokset. Haasteellisuutta lisää se, että uusi ja vanha käytäntö kohtaavat. Ratkaisun pilotointi/ testaus on tärkeää, sillä harvoin osutaan oikeaan ratkaisuun ensimmäisellä arvauksella. Kun uusi menettely täyttää sille asetetut odotukset, sen kuvaus viimeistellään ennen käyttöönottoa. Jalkauttamiseen on useita tapoja, esim. top-down- tai just-in-time -menettely. Menettely on syytä valita organisaation koon ja kulttuurin perusteella. Muutoksen aikaansaamiseen organisaatiossa on myös syytä varata aikaa ja resursseja. (Gremba, Myers.)

Kokemusvaihe (Learning) päättää parantamissyklin. IDEAL-mallin tavoitteena on kohentaa organisaation muutosalttiutta jatkuvaan parantamiseen. Kokemukset ovat tärkeä tuleva kehittämisvara. Selvitetään

- miten aiottu tavoite saavutettiin resursoiduilla panoksilla
- mikä toimi hyvin
- mitä pitäisi tehdä toisin/ tehokkaammin?

Vastaukset kerätään, analysoidaan sekä dokumentoidaan ja tulokset arvioidaan aloitusvaiheen vaatimuksiin verraten. Suositukset kehittämistyön parantamiseksi dokumentoidaan tulevaisuutta varten ja välitetään organisaation johdolle. (Gremba, Myers.)

4 Prosessin parantaminen pienessä organisaatiossa

SEI:n tilastojen mukaan vuonna 2003 ohjelmiston laatumalleja käytti 187 pientä organisaatioita. Määrä on yli kolminkertaistunut vuodesta 1999 (54). (Guerrero, Eterovic.)

4.1 Kokemusten hankkiminen

SPI-panostus voi perustua johonkin ohjelmistotuotannon kypsyyssmalliin (CMM, SPICE, ISO9000) pyrkien laajamittaisempaan toiminnan parantamiseen tai se voi keskittyä parantamaan tiettyä organisaation toimintatapaa, käytäntöä. EU on

tukenut voimakkaasti ohjelmistotuotannon prosessien parantamista mm. ESSI-ohjelman (European Software and Systems Initiative) avulla.

ESSI:n pyrkimyksenä on koota ja levittää pk-organisaatioiden erilaisia kokemuksia ohjelmistotuotannon prosessien parantamisprojekteista. Kokemusten mukaan menestystekijöitä mallille ovat sen joustavuus ja helposti räätälöitävät arviointi- ja parantamiskeinot. Mallin käyttöönottoa helpottaa

- toimiva tukiverkko pienille organisaatioille
- tukipalvelut, mentorointi
- edistymiseen liitetty ulkopuolinen taloudellinen tuki
- tarvittaessa ulkoisena palveluna puolueeton arviointi. (Orci, Laryd.)

Vuosina 1994-2000 EU rahoitti yli 450 prosessien parantamiskokeilua (Conradi, Fuggetta). Tulokset ovat osin yhä vaatimattomia ja johtaneet jopa täyteen kaaokseen. Kaikesta huolimatta juuri tällaisen epäkypsän organisaation jos minkä tulisi viipymättä käynnistää prosessien parantamiseen tähtäävät toimet. Toisaalta kaaoksen keskellä on lähes mahdotonta osoittaa aika ja paikka kehittämisellä, kun resurssit ovat täysin kiinni tilanteen hallintaan saamiseksi.

Viisainta olisi aloittaa prosessien parantamiseen tähtäävät toimet jo organisaation toimintaa perustettaessa. Ilmeistä on, ettei parannettavaa silloin vielä juurikaan ole, mutta parantaminen pitää jo tuolloin organisoida osaksi toimintaa. Niinpä organisaation ollessa vielä pieni sen tulisi ryhtyä tähän. Pienen organisaation johtaminen on vielä yksinkertaista. Pieni organisaatio kehittää aluksi yhtä tai korkeintaan muutamaa ohjelmistotuotetta. Projektit ovat peräkkäisiä, harvemmin rinnakkaisten projektien johtamiskäytäntöjä vaativia. Tällöin muutoksenhallinta on helppoa, samoin vuoropuhelu. (Orci, Laryd.)

4.2 Pienen ohjelmistotalon kokemukset ja muu tutkimus

Guerrero (Guerrero, Eterovic) selvitti SW-CMM:n soveltuvuutta pienen organisaation käytäntöihin ja niiden parantamiseen. Hän johti projektia, jossa paikannettiin niitä organisaation ja sen toiminnan piirteitä, jotka ovat tunnusomaisia SW-CMM:n tasolle 2. Projekti osoitti, että prosessin parantaminen on joiltain osin helpompaa pienelle kuin isolle organisaatiolle.

Guerreron tutkima Link on n. 70-henkinen chileläinen ohjelmistotalo. Yritys sekä kehittää että ylläpitää resurssisuunnittelu- ja luottokorttiohjelmistoja sekä muita myynnin järjestelmiä pääasiassa paikallisille vähittäiskaupoille. Henkilöstöstä parikymmentä työskentelee ohjelmiston kehittämisyksikössä seuraavissa tehtävissä: yksikön johtaja, projektipäällikkö (4), suunnittelija (13), laatuvaastava, kokoonpanonhallinnasta vastaava. Henkilöstön koulutustausta on joko teknillinen opisto tai lukio.

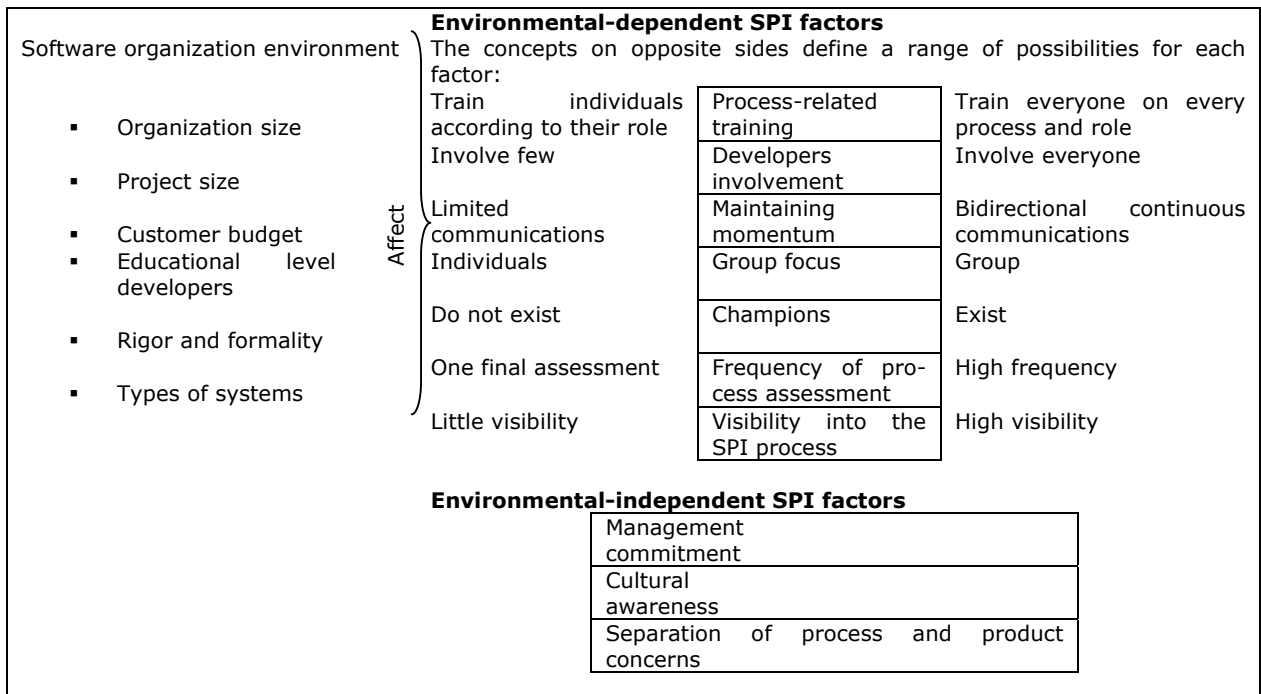
Linkin projektit ovat kooltaan pienehköjä: kesto 3-4 kuukautta, n. 5 000 \$ budjetti ja, niillä on yleensä vain yksi asiakas. Asiakkaan liikevaihto n. 200 000 \$ luokkaa. (Guerrero, Eterovic.)

Linkillä ei ollut aiempaa kokemusta prosessien parantamisesta, eikä kansainvälisistä standardeista. SW-CMM:n käyttöönotto oli linjassa Linkin 2002-2005 strategisten tavoitteiden kanssa. Tavoitteena oli viedä tuotteet kansainvälisille markkinoille. Ohjelmistoprosessien parantaminen oli nyt ajankohtainen ja yritys harkitsi myös sopimuskäytäntöihin, ISO 9001:2000 sekä BS7799 turvallisuusstandardiin panostusta. Aloite prosessien parantamisesta johti projektin käynnistymiseen maaliskuussa 2002. Link rahoitti itse projektin ja koko ohjelmistokehitysyksikkö osallistui ulkopuolisen konsultin johdolla kymmenkuiseen SPI-rupeamaan, johon käytettiin 2100 hth eli n. 280 työpäivää (n. 14 hlötyökuukautta) ja 50 000 \$. SPI eteni IDEAL-mallin (initiating, diagnosing, establishing, acting, learning) mukaisesti. (Guerrero, Eterovic.)

Link harkitsi myös CMMI:tä, mutta kustannukset olisivat ylittäneet Linkin budjetin. Vaikka SPI-projekti on yhä kesken, eikä lopullisia tuloksia vielä ole, on jo voitu osoittaa seuraavia hyötyjä:

- asiakas saa nyt työmäärä- ja kustannusarvion 2 vrk:n kuluessa tarjouspyynnöstä/ toimeksiannosta
- kunnollinen vaatimusmäärittäminen on vähentänyt uudelleen tehtävän työn määrää ja virheiden määrä per vaatimusten määrä on pudonnut lähes viiteen prosenttiin
- lisäksi Link on herättänyt kansallista kiinnostusta parantamisen (benchmarking) kohteena, ja Latinalaisen Amerikan kiinnostus SPI:a kohtaan on herännyt
- huhtikuussa 2004 Link arvioitiin SW-CMM tasolle 3. (Guerrero, Eterovic.)

Kuvassa 4 on esitetty kymmenen tekijää, joista osa on organisaatiosta tai sen toimintaympäristöstä riippuvia ja osa riippumattomia. (Guerrero, Eterovic.)



kuva 4. Prosessin parantamisen menestystekijät ja niiden suhde organisaation toimintaympäristöön (Guerrero, Eterovic)

Kuvan 4 organisaatiosta riippumattomat tekijät ovat lähtökohtia onnistumiselle tai epäonnistumiselle SPI:ssa. Organisaation johdon sitoutuminen on elintärkeää, ilman sitä kaikki on turhaa panostusta. Linkin

- johto satsasi SPI-työhön toimien aidon sponsorin tavoin sijoittaen ja mahdollistaen oikeat resurssit kehittämiseen
- johto ymmärsi myös organisaatiokulttuurin merkityksen, siksi kehittämiseen ryhdyttiin
- perustuotannon oli sujuttava kehittämisen aikanakin. Linkissä nimettiin koordinoitiryhmä, joka huolehti että päivittäiset työt jaettiin osa-aikaisille. Koordinoitiryhmään kuului sekä teknisen että hallinnollisen taustan omaava henkilö. (Guerrero, Eterovic.)

Linkin koko mahdollisti, että kaikki saattoivat osallistua koulutukseen ja valmentautua SW-CMM tason 2 rooliinsa: kehittäjät, laatuvaastava ja kokoonpanovastava kävivät projektin johtamiskoulutuksen. Isossa ja epäkypsemmässä organisaatiossa moinen ei olisi mahdollista. Niissä koulutus on hankittu ennen töihin tuloa tai organisaatio täsmäkouluttaa osajan tehtäviinsä. (Guerrero, Eterovic.)

Link rohkaisi koko ohjelmistotuotantohenkilöstöään valitsemaan avainprosessialueen, jonka parantamisesta haluaa vastata; prosessin omistajan roolin. Jälleen organisaation koko auttoi osallistamaan jokaisen. Isossa organisaatiossa prosessien omistajuuksia riittää vain harvalle ydinhenkilölle. (Guerrero, Eterovic.)

Kehittämisen näkyväksi tekeminen ja jalkauttaminen on helppoa pienessä organisaatiossa: kaikki tietävät, mitä kukin tekee. Linkissä kokouksiin voivat osallistua kaikki 20. Isossa organisaatiossa ongelmana on joko tietotulva tai tieto ei kulje, palaute on mahdollista vain ohjauskokouksissa ja edistymisraporteissa. (Guerrero, Eterovic.)

Linkin projektit ovat lyhytikäisiä, joten arviointituloksia saadaan myös tiuhemmin. Niinpä se on voinut suorittaa itsearviointin SW-CMM -dokumentaation varassa. Ulkopuolinen konsultti on tehnyt miniarvioinnin ja Link on arvioitu jo virallisesti (CMM-Based Appraisal for Internal Process Improvement, CBA-IPI) SW-CMM tasolle 3. Laajojen ja pitkäkestoisten projektien elämänkaaren pituus saattaa rajoittaa todellisten mittatulosten saamista. (Guerrero, Eterovic.)

Guerreron ja Eterovicin mukaan suuruus voi olla haitta ryhdyttäessä laajamittaiseen prosessien kehittämiseen ja parantamiseen. Samaan tulokseen on tullut myös Dybå tutkiessaan pienten ja suurten organisaatioiden SPI-hankkeiden onnistumista. Hänen tutkimuksessaan, jonka vastausprosentti oli 77,9 (n=154), oli mukana 120 organisaatiota.

Dybå'n tutkimuksen mukaan isojen ja pienten ohjelmistotuotanto-organisaatioiden välillä on perustavaa laatua olevia eroja. Suuret raportoivat päässeensä korkeammalle kypsyystasolle kuin pienet. Verrattaessa menestyneitä, niin suuria kuin pieniäkin, ei ero kuitenkaan ollut näkyvää. Pienet organisaatiot raportoivat henkilöstön laajasta osallistumisesta ja uuden taidon hankkimiseen käytetystä ajasta, joka ei näkynyt suurten kohdalla merkittävänä parantamisen strategiana. SPI:n hyväksikäytön suhteen eroa pienten ja suurten välillä ei ollut havaittavissa, sen sijaan ero oli merkittävä menettelyyn paneutumisessa. Pienet panostavat

huomattavan paljon toimintaympäristönsä kehittämiseen verrattuna suuriin. Suurissa ohjelmistotuotanto-organisaatioissa luotetaan yhä kokemuksista oppimiseen enemmän kuin uusien mahdollisuuksien tutkimiseen tai kokeiluihin. Pienet organisaatiot ovat joustavammin ympäristössään tapahtuviin muutoksiin reagoivia kuin suuret. Mitä suuremmaksi organisaatio kasvaa, sitä muodollisempaa on sen käyttäytyminen, organisaatiokulttuuri. Vakaisissa oloissa laitostuneet käytännöt ovat hyvinkin parhaita käytäntöjä. Mitä syvemmälle käytännöt ovat juurtuneet, sitä vaikeampaa niitä on muuttaa ja sitä varmemmin niistä tulee parantamisen este. Eräs selitys on myös, että parhaat käytännöt –lähestymistapa ei pidä virheitä hyväksyttävänä. Virheet kuitenkin ovat oppimisen kannalta tärkeitä, ne osoittavat puutteet ja käsitykset (nk. itsestään selvyudet) toiminnassa. (Dybå.)

5 Yhteenveto

Organisaation toiminnan, ohjelmistotuotannon, parantamisen tarve lähtee aina organisaatiosta itsestään ja sen liiketoiminnan tavoitteista. Johdon sitoutuminen on sekä keskeinen tekijä että haaste prosessin parantamiselle. On varsin vaikea osoittaa pysyvästi resurssit jatkuvalle parantamiselle pitäen sitä perusteltuna liiketoimintastrategiana. Pienessä organisaatiossa ei tällaiseen juurikaan ole mahdollisuutta varsinkaan kiihkeinä kasvun aikoina. Silti prosessit tulee määrittää ennen kuin toiminta on kasvanut ja saavuttanut mitat, joissa hallittu prosessi on edellytys toiminnan onnistumiselle, uskottavuudelle ja jatkumiselle.

Pienen organisaation on mietittävä tarkasti resurssit, jotka se voi panostaa prosessien kehittämiseen. Dynamic CMM tarjoaa lähtökohdat pienen organisaation prosessien parantamistyölle. Se nimeää keskeiset roolit siten, että organisaation perustoiminta on samanaikaisesti mahdollista. Dynamic CMM on kypsyysmalli, jonka käyttöönottoa tukee IDEALSM Model. IDEAL tarjoaa kokonaisvaltaisen lähestymistavan, etenemissuosituksen, prosessien parantamiselle kohti haluttua kypsyystasoa.

Pienen organisaation koko asettaa reunaehdot prosessien parantamiselle. Siksi kritiikitön ja vääristä lähtökohdista kumpuava kehittäminen kypsyysmallia käyttäen ei johda haluttuun lopputulokseen. Organisaation kasvaessa jokaisen sen jäsenen on ymmärrettävä yhdenmukaisen käytännön merkitys työn onnistumiseksi. Silti laadun takeena on vain osaava henkilöstö ja tarpeenmukaiset välineet sekä menettelyt ohjelmistojen kehittämiseen. Kypsyysmalli voi edellyttää enemmän.

Pienten organisaatioiden lähestymistapa on joustava, eikä organisaation koko rajoita sen mahdollisuuksia ohjelmistotuotantoprosessien parantamiselle. Päinvastoin, niillä näyttää olevan etua koosta ja toimintakykyensä säilyttämiseksi ne panostavat uusien mahdollisuuksien selvittämiseen tehden kaikkensa ruokkiakseen henkilöstön osaamista ja sitä kautta menestymistään markkinoilla.

Ketterät menettelyt voisivat olla eräs vartenotettava menetelmistö pienen organisaation ohjelmistotuotantoprosessille. Link osoitti, että SPI:ssä keskeiset organisaation koosta johtuvat tekijät ovat pitkälti samoja kuin ketterien menettelyiden arvot ja periaatteet. CMMI:ssä ketteryyteen on kiinnitetty huomiota.

Lähteet

- Börjesson, A. Mathiassen, L.
Successful Process Implementation. IEEE Software, July/August 2004, 36-44.
- Conradi, R. Fuggetta, A.
Improving Software Process Improvement. IEEE Software, July/August 2002, 2-9.
- Dybå, T. Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context.
<http://delivery.acm.org/10.1145/950000/940092/p148-dyba.pdf?key1=940092&key2=9868749901&coll=GUIDE&dl=GUIDE&CFID=30397676&CFTOKEN=37348254> [viitattu 21.10.2004]
- Fenton, N. E. Pfleeger, S. L. 1997.
Software Metrics, A Rigorous & Practical Approach. PSW publishing Company, Boston, 2. edition.
- Gremba, J. Myers, C. The IDEALSM Model: A Practical Guide for Imprvment.
<http://www.sei.cmu.edu/ideal/> [viitattu 21.10.2004]
- Guerrero, F. Eterovic, Y.
Adopting the SW-CMM in a Small IT Organization. IEEE Software, July/August 2004, 29-35.
- Kalermo, J. Rissanen, J. 2002. Agile Software Development in Theory and Practice. Master's thesis, University of Jyväskylä.
http://www.cs.jyu.fi/sb/Publications/KalermoRissanen_MastersThesis_060802.pdf [viitattu 19.10.2004]
- Kuitunen et al.
Kuitunen, K. Ilomäki, S. Simons, M. Valjakka, T. (toim.) 2003.
Kehity kasvuun. Pk-yrityksen kasvu ja kehittäminen. Työelämän kehittämisohjelma. Työministeriö, Raportteja 29. ISBN 951-735-764-8.
- Laryd, A. Orci, T. Dynamic CMM for Small Organizations.
<http://www.uml.org.cn/cmm/pdf/1116/laryd00dynamic.pdf> [viitattu 21.10.2004]
- Orci, T. Laryd, A. Dynamic CMM for Small Organisations – Implementation Aspects. http://www.iscn.at/select_newspaper/requirements/umea.html [viitattu 21.10.2004]
- Lindvall, M. Rus, I. Process Diversity in Software Development. IEEE Software, July/August 2000, 14-18.
- Pollice et al.
Pollice, G. Augustine, L. Lowe, C. Madhur, J. 2003.
Software Development for Small Teams. A RUP-Centric Approach. Addison-Wesley. ISBN 0-321-19950-2.