

hyväksymispäivä arvosana

arvostelija

## **TFS:n ja ketterien prosessien hyödyntäminen CMMI-mallissa**

Anu Ranta

Helsinki 25.03.2009

HELSINGIN YLIOPISTO  
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta – Fakultet – Faculty		Laitos – Institution – Department	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen laitos	
Tekijä – Författare – Author			
Anu Ranta			
Työn nimi – Arbetets titel – Title			
TFS:n ja ketterien prosessien hyödyntäminen CMMI-mallissa			
Oppiaine – Läroämne – Subject			
Tietojenkäsittelytiede			
Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages	
	25.3.2009	x sivua + x liitesivu	
Tiivistelmä – Referat – Abstract			
<p>ACM Computing Classification System (CCS):  A.1 [Introductory and Survey],  I.7.m [Document and text processing]</p>			
Avainsanat – Nyckelord – Keywords			
Team Foundation Server, CMMI, ketterät prosessit			
Säilytyspaikka – Förvaringställe – Where deposited			
Muita tietoja – Övriga uppgifter – Additional information			

# Sisältö

<b>1</b>	<b>Johdanto .....</b>	<b>1</b>
<b>2</b>	<b>CMMI-malli .....</b>	<b>2</b>
2.1	Mallin historia .....	3
2.2	CMMI menetelmät .....	5
2.3	Prosessialueet .....	7
2.4	Kypsyys ja kyvykkyys .....	8
<b>3</b>	<b>Ketterät prosessimallit .....</b>	<b>9</b>
3.1	Ketterän kehityksen mentelmät .....	9
3.2	Scrum .....	9
3.3	eXtreme Programming .....	9
<b>4</b>	<b>CMMI ja ketterät prosessimallit .....</b>	<b>10</b>
4.1	Ketterät menetelmät ja CMMI tavoitteet .....	11
4.2	Menetelmien yhtäläisyydet ja eroavaisuudet .....	11
4.3	Tapaustutkimuksia .....	11
<b>5</b>	<b>Team Foundation Server ja MSF for CMMI process improvement .....</b>	<b>12</b>
5.1	Team Foundation Server .....	12
5.2	MSF for CMMI process improvement.....	12
<b>6</b>	<b>Analyysi TFS:n kyvystä yhdistää ketterät menetelmät ja CMMI tavoitteet .....</b>	<b>12</b>
<b>7</b>	<b>Yhteenveto .....</b>	<b>12</b>

# 1 Johdanto

CMMI (Capability Maturity Model Integrated) [CMM02] on maailmalla hyvin tunnettu prosessin kehittämisen malli. CMMI-malli siinä missä muutkin prosessin kehittämiseen pyrkivät mallit luottavat siihen, että hyvin hallitut organisaatiot, joilla on tarkkaan määritellyt prosessit, tuottavat ohjelmistoja, jotka täyttävät niille asetetut vaatimukset ja valmistuvat asetetussa aikataulussa ja budjetissa ennemmin kuin huonosti hallitut organisaatiot [FrK07]. Mallia pidetään kuitenkin usein raskaana ja byrokraattisena.

Ketterät prosessimallit suhtautuvat prosesseihin keveämmin. Ketterissä menetelmissä pääpaino on enemmän yksilöissä ja vuorovaikutuksessa kuin prosesseissa ja työkaluissa. Ketterät menetelmät painottavat, että ohjelmiston kehitys on luova prosessi, joka syntyy ihmisten välisen yhteistyön, eikä byrokratian ja tiukkojen prosessien kautta. Menetelmän pyrkimyksenä on tuottaa jatkuvasti pienissä sykleissä hyödyllisiä versioita ohjelmistosta. Näin asiakas on kokoajan mukana ohjelmiston kehityksessä ja muuttuviin vaatimuksiin on mahdollista reagoida herkemmin [FrK07].

CMMI-mallin ja ketterien menetelmien yhdistämisestä on eri näkemyksiä. Suurimmaksi osaksi niiden yhteensovittamista on tähän mennessä pidetty huonona ideana. Toisenlaisia näkemyksiä asiasta on kuitenkin viime aikoina noussut esiin. Näiden näkemysten mukaan ohjelmistokehitystä voidaan tehostaa yhdistämällä ketterät menetelmät ja perinteisen CMMI-mallin tavoitteet [Pik08].

Team Foundation Server (TFS) on Microsoftin luoma sovelluskehitysprojektin hallintapalvelin. Palvelimen avulla voidaan hallita kehitysprojektia koko sen elinkaaren ajan. TFS sisältää oletuksena kaksi prosessimallia: MSF for Agile Software Development ja CMMI for Process Improvement. CMMI for Process Improvement on laajennos MSF for Agile mallista, joten sen luvataan olevan hyvin ketterä.

Tämän tutkielman tavoitteena on analysoida sitä, kuinka TFS projektin elinkaaren hallintatyökaluna tukee ketterien prosessien hyödyntämistä CMMI-mallissa. Toisin sanoen ketterien menetelmien yhdistämistä CMMI-mallin asettamiin tavoitteisiin.

## 2 CMMI-malli

Nykypäivänä yritys on yhä useammin tilanteessa, jossa sen toiminta on hajautettu ympäri maailmaa. Toiminta ei ole hajautettu vain maantieteellisesti yrityksen sisällä vaan yritys harvemmin tuottaa itse kaikkia tarjoamansa tuotteen tai palvelun komponentteja. Osa komponenteista tuotetaan itse osa hankitaan ulkopuolelta. Samanaikaisesti yritykset haluavat tuottaa tuotteita ja palveluita tehokkaammin, nopeammin ja alhaisemmin kustannuksin [CMM02]. Organisaatioiden on pystyttävä näin hallitsemaan tehokkaasti monimutkaisia kehitys- ja ylläpitoprosessejaan, mikä asettaa melkoisen haasteen prosessinhallinnalle.

CMMI (Capability Maturity Model Integrated) [CMM02] on prosessin kehittämisen malli, jota voidaan soveltaa sekä tuotteiden että palveluiden tuottamiseen. Mallin avulla organisaatio voi yhtenäistää toimintatapojaan tai parantaa prosessejaan. Malli koostuu tuotteiden ja palveluiden ylläpidon parhaista käytännöistä. Se osoittaa käytännöt, jotka kattavat tuotteen koko elinkaaren suunnittelusta aina ylläpitoon asti. Painopiste on osaluissa, jotka ovat välttämättömiä tuotteen rakentamisen ja ylläpidon kannalta [CMM02]. On kuitenkin tärkeä huomata, että CMMI-malli ei sisällä määrääviä prosesseja, joita voidaan sellaisenaan suoraan käyttää. Sen sijaan CMMI tarjoaa menetelmän arvioida organisaation kyvyn rakentaa ohjelmia toistettavalla ja ennustettavalla tavalla [MäP06].

Organisaatiot aloittavat prosessien kehittämisen eri syistä. Joillekin syy on juuri kehitysprojektien hajaantuminen maailmanlaajuisesti. Toisia ajaa eteenpäin halu näyttää, että organisaatio noudattaa globaaleja standardeja. Syy saattaa olla myös merkittävässä liikemaailman tekijöissä kuten sopimuksen häviämisessä. Uusi markkinamahdollisuus voi myös vetää huomion prosessien kehittämiseen. Jotkut tajuavat, että tarvitaan prosessitason kehitystä, kun heidän tuotteissaan ilmenee jatkuvasti virheitä vielä julkaisunkin jälkeen. Toisia ajaa eteenpäin säännökset/vaatimukset, kuten on saavutettava tietty sertifikaatti CMMI:n tietyllä tasolla, näyttääkseen että he noudattavat globaaleja standardeja. Merkittävät liikemaailman tekijät, kuten sopimuksen häviäminen tai uusi markkinamahdollisuus, voivat myös vetää huomion prosessin kehittämiseen.

## 2.1 Mallin historia

Ensimmäinen kypsyystasomallin mukainen viitekehys syntyi Watts Humphreyn [Hum89] johdosta vuonna 1989. Kirja kuvaa peruseriaatteet ja konseptit, joihin moni kypsyysmalli pohjautuu [CMM02]. Muutama vuosi aikaisemmin Yhdysvaltain puolustusvoimien huomio kiinnittyi ohjelmistoihin, joita ei joko koskaan toimitettu tai ne toimitettiin myöhässä kattaen vain osan odotetuista ominaisuuksista. Puolustusvoimat pyrkimyksenä oli selvittää, mihin kului näihin ohjelmistoihin investoidut suuret summat rahaa. Puolustusvoimien aloitteen pohjalta syntyi Software Engineering Institute (SEI) Carnegie Mellon Yliopistossa. SEI:n perustama työryhmä koostui akateemisista henkilöistä, tutkijoista sekä teollisuuden edustajista. Työryhmän pyrkimyksenä oli tuoda esiin toimintatapoja, joiden avulla välttyttäisiin jatkuvilta ohjelmistoprojektien epäonnistumisilta. Työryhmän tuloksena syntyi CMM [GDA08].

Alkuperäisen CMM parissa työskennellyt työryhmä etsi ratkaisua ohjelmistokehitystä riivaavaan ongelmaan sillä ajatuksella, että ohjelmisto oli vain osa suurempaa järjestelmää. Puolustusvoimain ohjelmistojen luonteen vuoksi oletuksena olikin, että jos ohjelmistossa esiintyisi virhe, menetettäisiin elämiä. Tämän vuoksi järjestelmien kehitys oli hyvin varovaista ja harkittua [GDA08].

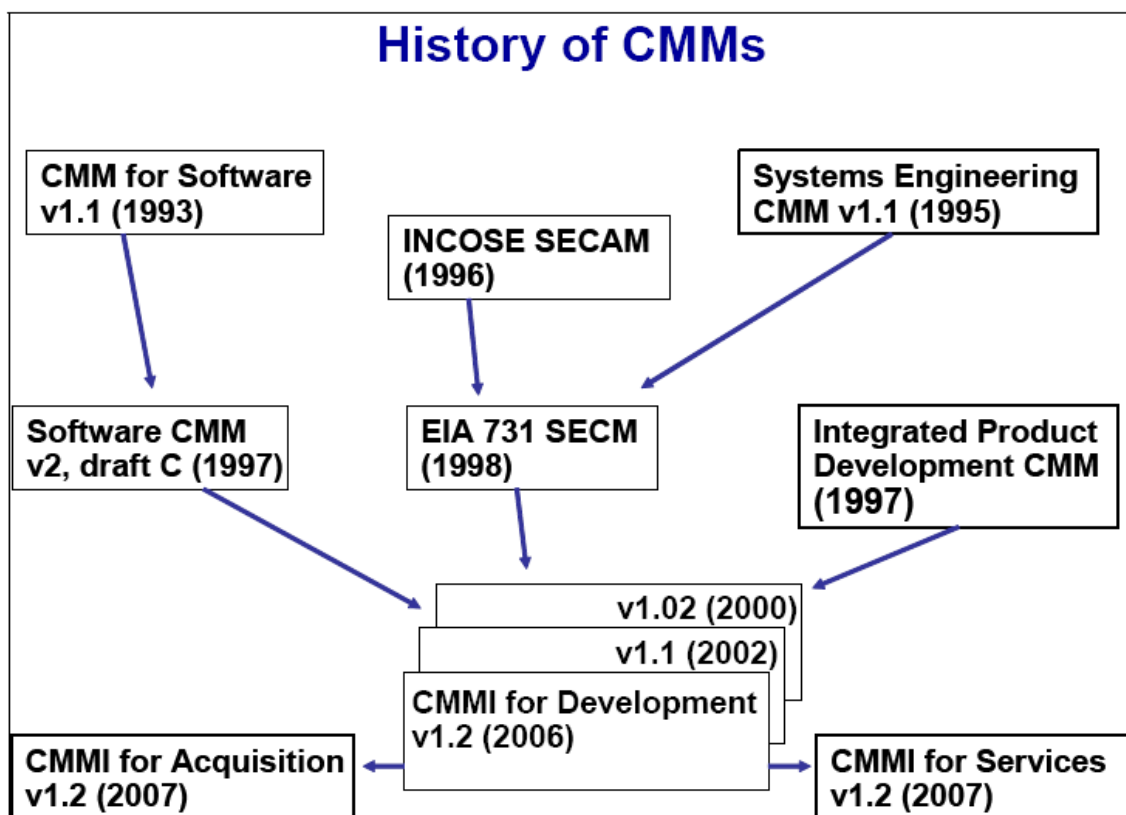
Muutaman vuoden kuluessa CMM oli laajentunut useammaksi, yrityksen eri prosessialueille suunnatuiksi malleiksi. Mallit olivat kehitetty jäsentämään lähinnä muiden kuin ohjelmistojen kehitysprojekteja. Omat mallit saivat muiden muassa ohjelmistotekniikka, ohjelmistojen hankinta, työvoiman hallinta ja kehittäminen sekä yhdistetty tuotteiden ja prosessien kehitys [CMM02]. CMM ja sen muunnoksien käyttö kasvoi kansainvälisesti. Organisaatiot, jotka pyrkivät omaksumaan useamman mallin projektin eri osa-alueilla, huomasivat kuitenkin pian sen asettamat haasteet [GDA08]. Vaikka mallien oli todettu olevan hyödyllisiä monille organisaatioille, oli niiden käyttö yhdessä ongelmallista. Mallien väliset erot, arkkitehtuuri, sisältö ja näkökulma rajoittivat yritysten hyödyntämistä niitä onnistuneesti. Lisäksi usean erillisen mallin käyttö organisaatiossa oli kallista niiden tarvitseman koulutuksen ja arvioinnin vuoksi [CMM02]. Tämän seuraksena yritykset anoivat SEI:tä yhdistämään lukuisat mallit yhdeksi malliksi [GDA08].

CMM Integration (CMMI) projekti pystytettiin ratkaisemaan usean mallin aiheuttamat ongelmat. CMMI työryhmän tavoitteena oli yhdistää kolme erillistä CMM-mallia: The

Capability Maturity Model for Software (SW-CMM), The Systems Engineering Capability Model (SECM) ja The Integrated Product Development Capability Maturity Model (IPD-CMM). Syntyvä malli oli tarkoitus suunnata yrityksille, jotka pyrkivät koko yrityksen kattavien prosessien kehittämiseen [CMM02].

Käyttäen lähteinään näitä hyvin arvostettuja malleja työryhmä loi yhtenäisen mallin, jonka omaksumisen pitäisi olla mahdollista sekä yrityksissä, joissa käytetään CMMI:n lähteenä olleita malleja, että kokonaan uusissa yrityksissä. Näin ollen CMMI on syntynyt SW-CMM, SECM ja IPD-CMM mallien evoluution tuloksena [CMM02].

Viime vuosina CMMI-malli on laajentunu kattamaan myös ohjelmistojen ja palveluiden hankinnan. Tätä varten on luotu oma malli: CMMI for Acquisition (CMMI-ACQ). Tänä vuonna on lisäksi julkaistu oma malli palveluntarjoajia varten: CMMI for Services (CMMI-SVC).

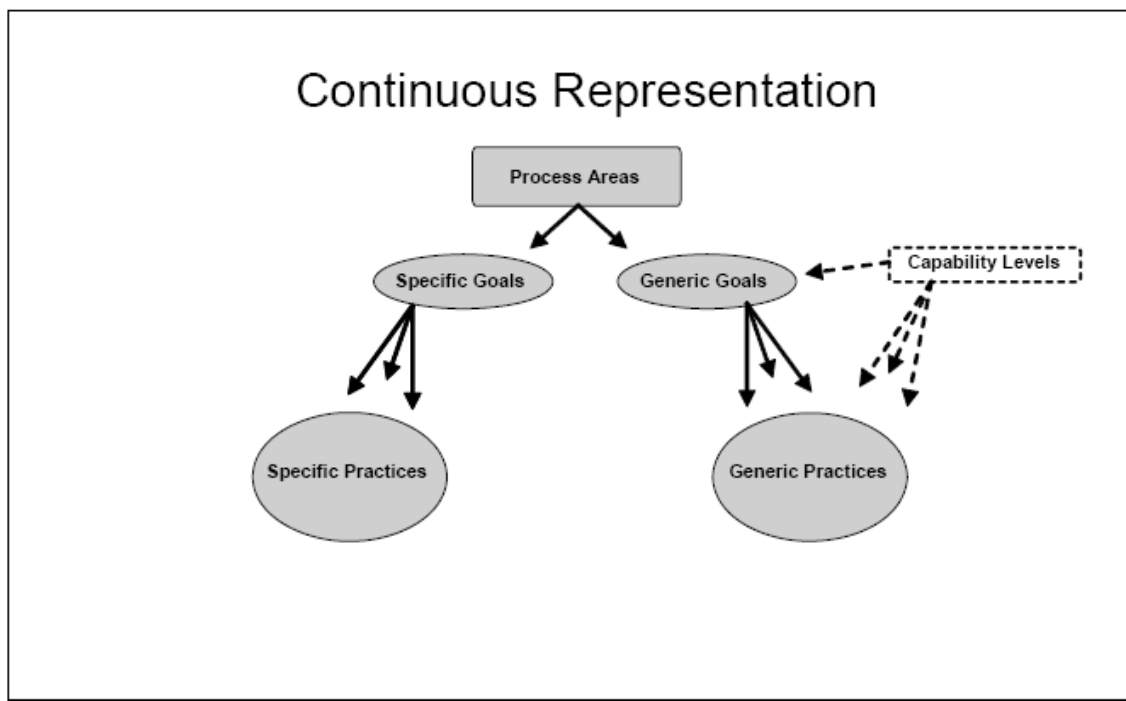


Kuva 1: Kypsyysmallien historia. [CMM02]

## 2.2 CMMI menetelmät

CMMI-mallin soveltamistavaksi voidaan valita joko tasomalli tai jatkuva malli. Jatkuva malli tarjoaa maksimaalista joustavuutta, kun CMMI-mallia käytetään prosessien parantamiseen. Organisaatio voi valita yksittäisen prosessialueen, jonka ongelmakenttää se haluaa kehittää, tai vaihtoehtoisesti keskittyä useampaan prosessialueeseen samanaikaisesti. Lisäksi jatkuva malli antaa organisaatiolle mahdollisuuden kehittää eri prosesseja eri tasoilla. Rajoittavuusiakin tosin löytyy, sillä osa prosessialueista on toisistaan riippuvaisia.

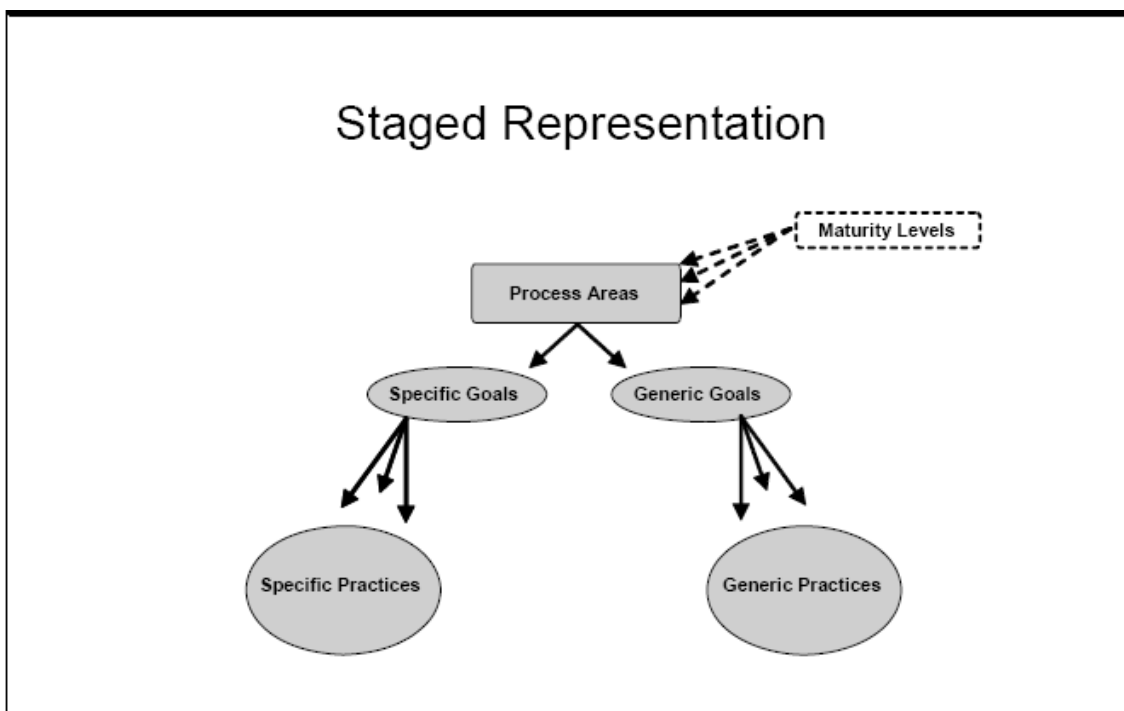
Jaksoittaisessa mallissa jokainen prosessialue on arvioitu kyvykkyystasoilla [MäP06]. Jatkuva malli on hyvä valinta organisaatiolle, jossa tunnistetaan hyvin kehitystä vaativat prosessit sekä ymmärretään CMMI-mallin mukaisten prosessialueiden riippuvuudet [CMM02].



Kuva 1. Jatkuvan mallin rakenne [CMM02]



Tasomalli sen sijaan tarjoaa systemaattisen, portaittaisen tavan tavoitella mallin mukaista prosessien kehitystä taso kerrallaan. Yksittäisen tason saavuttaminen varmistaa, että seuraavan tason edellyttämä prosessipohja on kunnossa. Prosessialueet on organisoitu kypsyystasoin [MäP06]. Tasomalli kuvaa kypsyystasojen mukaisen järjestyksen, jossa prosessialueet otetaan käyttöön. Kypsyystasot määrittävät organisaatiolle kehityspolun alkutasolta tavoitellulle tasolle. Tasomalli on hyvä valinta organisaatiolle, jos organisaatio ei tiedä, mistä aloittaa ja mitä prosesseja kehittää. Tasomalli antaa selkeän prosessivalikoiman, joihin keskittyä kullakin tasolla [CMM02].



Kuva 2: Tasomallin rakenne [CMM02]

## 2.3 Prosessialueet

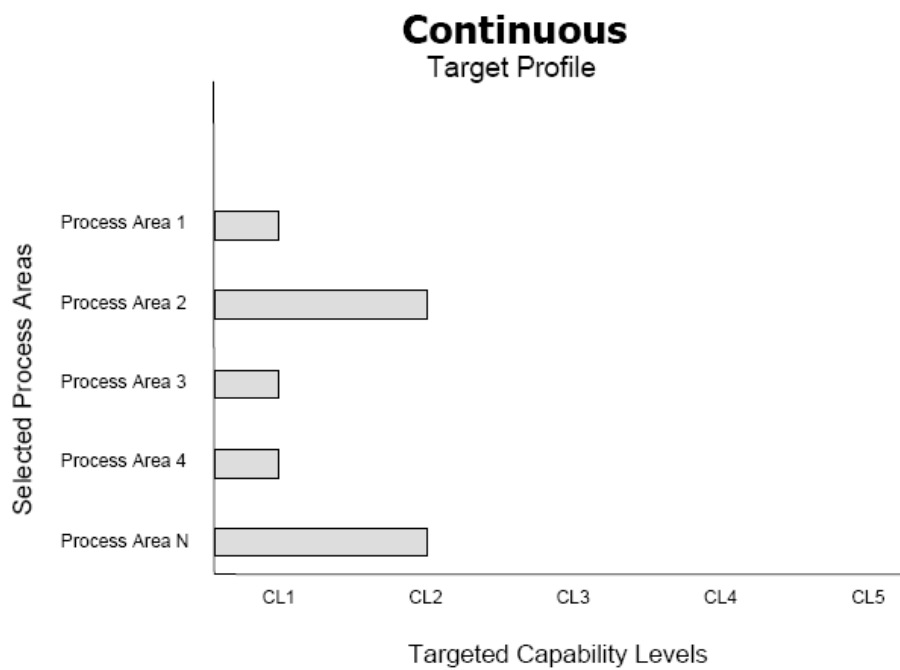
CMMI kypsyyssmalli sisältää 22 avainprosessialuetta. Molemmat CMMI menetelmät, jatkuva ja tasoittainen, sisältävät nämä kaikki 22 prosessialuetta.

- Configuration Management (CM)
- Decision Analysis and Resolution (DAR)
- Integrated Project Management +IPPD (IPM+IPPD)6
- Measurement and Analysis (MA)
- Organizational Innovation and Deployment (OID)
- Organizational Process Definition +IPPD (OPD+IPPD)6
- Organizational Process Focus (OPF)
- Organizational Process Performance (OPP)
- Organizational Training (OT)
- Product Integration (PI)
- Project Monitoring and Control (PMC)
- Project Planning (PP)
- Process and Product Quality Assurance (PPQA)
- Quantitative Project Management (QPM)
- Requirements Development (RD)
- Requirements Management (REQM)
- Risk Management (RSKM)
- Supplier Agreement Management (SAM)
- Technical Solution (TS)
- Validation (VAL)
- Verification (VER)

Käydään tarkemmin läpi prosessialueet ja niiden vaatimukset.

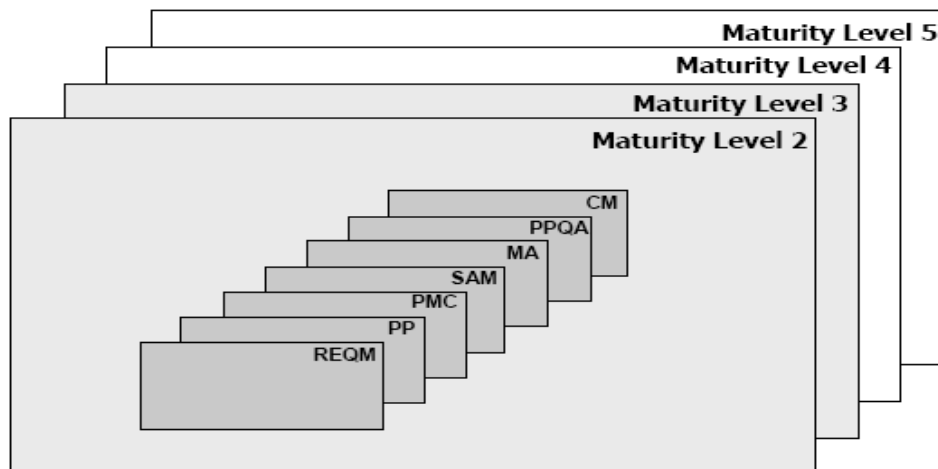
## 2.4 Kypsyys ja kyvykkyys

Kuvataan jatkuvan mallin ja tasomallin mukaiset tavat lähestyä prosessialueita kyvykkyystasoisin ja kypsyytasoisin [CMM02].



Kuva 3: Prosessialueiden käyttö jatkuvassa mallissa [CMM02]

## Staged Selected Maturity Level



Kuva 4: Prosessialueiden käyttö tasomallissa [CMM02]

## 3 Ketterät prosessimallit

### 3.1 Ketterän kehityksen mentelmät

Perusominaisuudet, ketteryyden tekijät

[BoT05], [CoH01]

### 3.2 Scrum

Kuvataan Scrum.

[BeS02], [RiJ00], [Sch97]

### 3.3 eXtreme Programming

Kuvataan XP [Bec99], [Bec00].

## 4 CMMI ja ketterät prosessimallit

Yrityksillä on ollut vahva luotto CMMI-malliin sen tarjoamien parheiden käytäntöjen ja niiden kuvaavan yhteensopivuuden vuoksi. Ketterät menetelmät lupaavat myös lisääntyntä asiakastytyväisyyttä, tuottaa erittäin laadukasta ohjelmistoa ja nopeuttaa kehitysaikoja, mikä on omalta osaltaan kasvattanut yritysten kiinnostusta ketteriä menetelmiä kohtaan. Toisaalta ketteriä menetelmiä on kritisoitu järjestyksen puutteesta ja niiden on väitetty olevan sopia vain tietyn tyyppisiin projekteihin [MäP06].

Ohjelmistokehityksen parissa toimivat yritykset ovat kasvavassamäärin kiinnostuneita omaksumaan ketteriä kehitysmenetelmiä liiketoiminnassaan. Organisaatiot, jotka ovat ponnistelleet parantaakseen prosesseja CMMI-mallin avulla, muuttavat nyt prosessejaan ketterien menetelmien suuntaan. Yritykset ovat ymmärtäneet, että ketterät menetelmät voivat myös tarjota kehitystä. CMMI-malli ei kuitenkaan aina ole sellaisenaan sovellettavissa ketterien menetelmien yhteydessä [MäP06].

Viime vuosina ketterät menetelmät kuten eXtreme Programming ovat nostaneet suosiotaan. Samanaikaisesti useammat ja useammat organisaatiot luottavat prosessien kehitysmalleihin arvioidakseen ja kehittääkseen omia prosessejaan tai alihankkijan prosessia, koska on käynyt yhä selvemmäksi, että useimmat projektien epäonnistumiset johtuvat epäjohtonmukaisista ja vallattomista prosesseista. Monet organisaatiot jopa vaativat CMMI noudattamista projekteissa, joissa käytetään ketteriä menetelmiä. Asiakkaat tai organisaatiot, jotka aloittavat hajautettuja projekteja, usein luottavat kypsyystasomalleihin arvioidessaan ja valitessaan alihankkijoita. Mallien avulla voidaan arvioida alihankkijan prosessien kypsyttä. Suuremmissa organisaatioissa noudatetaan toimintatapoja, jotka vaativat, että organisaation kaikkien osapuolten täytyy saavuttaa tietyt kypsyystasot [FrK06].

Suunnittelupohjainenkehitys ja ketterät menetelmät nähdään usein toistensa vastakohtina [JaR02]. Suunnitelupohjaiset lähestymistavat olettavat, että sovelluskehitys on toistettava ja ennustettava prosessi. Ketterän kehityksen puolestapuhujat eivät usko, että nämä oletukset pitävät paikkaansa projekteissa, joihin kuuluu minkään tasoista suunnittelua.

Osa organisaatioista pyrkii hyödyntämään molempia CMMI-mallin tavoitteita ja ketteriä menetelmiä kehittääkseen toimintaansa. Kuitenkin, molemmilla menetelmillä

on puutteensa ja tekijänsä, jotka täytyy selvittää parempien tulosten saavuttamiseksi. Haaste on tasapainottaa nämä kaksi menetelmää niin, että hyödynnetään niiden vahvuudet, ja näin kompensoidaan niiden heikkoudet [FrK06].

#### **4.1 Ketterät menetelmät ja CMMI tavoitteet**

Verrataan, miten hyvin ketterät menetelmät pystyvät täyttämään CMMI:n mukaiset tavoitteet. Kuinka pitkälle ketterät menetelmät tukevat CMMI-mallin mukaisia prosessialueita ja missä vaiheessa esiintyy konflikteja

[JaR02], [MFF07], [Pau01], [Vri03].

#### **4.2 Menetelmien yhtäläisyydet ja eroavaisuudet**

CMMI-mallia ja ketteriä menetelmiä on vertailtu useimmissa tutkimuksissa. Luodaan tässä katsaus niissä löydettyihin yhtäläisyyksiin ja eroavaisuuksiin

[BoV04], [JaJ08], [JaR02], [SNJ07].

#### **4.3 Tapaustutkimuksia**

CMMI-mallin käytöstä ketterän kehityksen projekteissa on tehty useita tapaustutkimuksia. Tapaustutkimusten pyrkimys kerätä tietoja CMMI:n käytöstä ketterän kehityksen yhteydessä. Tutkimuksen tarkoitus oli selvittää, kuinka CMMI-mallia voitaisiin käyttää ketterien mallien yhteydessä tai tilanteessa, jossa organisaatio suunnittelee muuttavansa prosessinsa ketterään suuntaan [MäP06].

Käydään läpi eri tapaustutkimuksia ja tehdään yhteenveto niissä esiintulleista seikoista

[AbK04], [And05], [BNW01], [Bak05], [Bak06], [BeS02], [JaJ08], [Pik08], [SaA05].

## **5 Team Foundation Server ja MSF for CMMI process improvement**

Luodaan kuvaus Team Foundation Serveriin ja sen tarjoamiin prosessimalleihin. Keskitytään pääasiassa tutkielmassa arvioitavaan CMMI process improvement prosessimalliin.

### **5.1 Team Foundation Server**

Kuvataan Team Foundation Serverin arkkitehtuuri sekä toimintaperiaatteet.

### **5.2 MSF for CMMI process improvement**

Kuvataan TFS:n tarjoama CMMI:n vaatimusten mukainen ketterä prosessimalli.

## **6 Analyysi TFS:n kyvystä yhdistää ketterät menetelmät ja CMMI tavoitteet**

Oma analyysi siitä, kuinka hyvin Team Foundation Server ja MSF for CMMI process improvement –prosessimalli onnistuu hyödyntämään ketteriä menetelmiä ja samalla täyttämään CMMI:n mukaiset vaatimukset.

## **7 Yhteenveto**

Yhteenveto

## Lähteet

- AbK04 Abrahamsson, P., Kähkönen, T., Achieving CMMI Level 2 with Enhanced Extreme Programming Approach, Proceedings of the 5th International Conference on Product Focused Software Process Improvement, 2004.
- AGI01 The Agile Manifesto. [<http://agilemanifesto.org/>]
- And05 Anderson, D., Stretching Agile to fit CMMI Level 3 the story of creating MSF for CMMI® Process Improvement at Microsoft Corporation. Proceedings of the Agile Development Conference (ADC'05), 2005.
- Bec00 Beck, K., Extreme Programming Explained: Embrace Change, AddisonWesley, 2000.
- BNW01 Bartosz, W., Nawrocki, J., Wojciechowski, A., Toward Maturity Model for eXtreme Programming, Proceedings of the 27th Euromicro Conference, 2001.
- Bak05 Baker, S., W., Formalizing Agility: An Agile Organization's Journey toward CMMI Accreditation. Proceedings of the Agile Development Conference (ADC'05), 2005.
- Bak06 Baker, S., W., Formalizing Agility, Part 2: How an Agile Organization Embraced The CMMI. Proceedings of AGILE 2006 Conference (AGILE'06), 2006.
- Bec99 Beck, K., Embracing Change with Extreme Programming. IEEE Computer, Lokakuu 1999, sivut 70-77.
- BeS02 Beedle, M., Schwaber, K., Agile Software Development With Scrum, Prentice Hall, 2002.
- BoT05 Boehm, B., Turner, R., Management Challenges to Implementing Agile Processes in Traditional Development Organizations, Software, vol. 22, issue 5, 3039, 2005.
- BoV04 Bos, E., Vriens, C., An agile CMM, Proceedings of 4th Conference on Extreme Programming and Agile Methods -XP/Agile Universe, 2004.



- CMM02 CMMI® for Development, Version 1.2, (CMMI-DEV, V1.2), 2002.
- CoH01 Cockburn, A., Highsmith, J., Agile Software Development: The People Factor. IEEE Computer, 34(11):131–133, 2001.
- FrK07 Fritzsche, M., Keil, P., Agile Methods and CMMI: Compatibility or Conflict?, e-Informatica Software Engineering Journal, Volume 1, Issue 1, 2007.
- GDA08 Glaser, H., Dalton, J., Andersson, D., Konrad, M., Schrum, S., CMMI® or Agile: Why Not Embrace Both, (CMU/SEI-2008-TN-003), Software Engineering Institute, 2008.
- GoG03 Goldenson, D., Gibson, D., Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results, tech. report CMU/SEI-2003-009, Software Eng. Inst., Carnegie Melon Univ., 2003
- Hum89 Humphrey, W. S., Managing the Software Process. Addison-Wesley Publishing Company, Inc., 1989.
- JaJ08 Jakobsen, C., Johnson, K., Mature Agile with a Twist of CMMI, Agile apos 08 Conference, IEEE, sivut 212-217, 2008.
- JaR02 Jain, A., Turner, R., Agile Meets CMMI: Culture Clash or Common Cause, Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods XP/Agile Universe, 2002.
- MFF07 Marcal, A., de Freitas, B., Furtado Soares, F., Belchior, A., Mapping CMMI Project Management Process Areas to SCRUM Practices, Software Engineering Workshop, IEEE, sivut 13-22, 2007.
- MäP06 Mäntyniemi, A., Pikkarainen, M., An approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies. VTT Technical Research Centre of Finland, 2006.
- Pau01 Paulk, M. C., Extreme Programming from a CMM Perspective, Software, vol. 18, issue 6, 2001.
- Pik08 Pikkarainen, M., Toward a Framework for Improving Software Development Process Mediated with CMMI goals and Agile Practices. VTT Publications 695, Espoo, 2008.

- RiJ00      Rising, L., Janoff, N., S., The Scrum Software Development Process for Small Teams. IEEE Software, sivut 26-32, heinä-elokuu, 2000.
- SaA05      Salo, O., Abrahamsson P., Integrating Agile Software Development and Software Process Improvement: a Longitudinal Case Study. IEEE, sivut 193-202, 2005.
- Sch97      Schwaber, K., Scrum Development Process, OOPSLA Business Object Design and Implementation Workshop, 1997.
- SNJ07      Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P., Murphy, R. An Exploratory Study of Why Organizations do not Adopt CMMI. Journal of Systems and Software 80(6),883-895, 2007.
- Vri03      Vriens, C., Certifying for CMM Level 2 and ISO9001 with XP@Scrum, Proceedings of the Agile Development Conference. Proceedings of the Volume, Issue, sivut 120-124, 2003.