

hyväksymispäivä arvosana

arvostelija

## **Prosessien kypsyysmallit hajautetussa ohjelmistokehityksessä**

Tevje Olin

Helsinki 22.10.2013

HELSINGIN YLIOPISTO  
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta/Osasto – Fakultet/Sektion – Faculty/Section		Laitos – Institution – Department	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen laitos	
Tekijä – Författare – Author			
Tevje Olin			
Työn nimi – Arbetets titel – Title			
Prosessi- ja kypsyysmallit hajautetussa ohjelmistokehityksessä			
Oppiaine – Läroämne – Subject			
Tietojenkäsittelytiede			
Työn laji – Arbetets art – Level	Aika – Datum – Month and year	Sivumäärä – Sidoantal – Number of pages	
Seminaarityö	22.10.2013	10 sivua	
Tiivistelmä – Referat – Abstract			
Tässä seminaarityössä kuvaillaan hajautettuun ohjelmistokehitykseen liittyviä ongelmia ja esitetään muutama tapaustutkimukseen perustuva ratkaisu prosessien ja käytäntöjen kypsyiden mittaamiseen ja kehittämiseen.			
Avainsanat – Nyckelord – Keywords			
Hajautettu ohjelmistokehitys, kypsyysmallit, prosessimallit			
Säilytyspaikka – Förvaringställe – Where deposited			

Muita tietoja – Övriga uppgifter – Additional information

## **Sisältö**

<b>1 Johdanto</b>	<b>1</b>
<b>2 Globaali ja hajautettu ohjelmistokehitys</b>	<b>1</b>
<b>3 Hajautetun ohjelmistokehityksen prosessien kypsyysmallit</b>	<b>2</b>
3.1 Tutkimus.....	2
3.2 Kypsyysmallit.....	4
3.2.1 CMM.....	4
3.2.2 A Process Maturity Framework for Managing Distributed development.....	5
3.2.3 A Preliminary capability model.....	8
<b>4 Yhteenveto</b>	<b>9</b>
<b>Lähteet</b>	<b>11</b>

# 1 Johdanto

Tietotekniikan merkitys kasvaa jatkuvasti kaikkialla liiketoiminnassa alasta riippumatta. Tämä lisää vaatimuksia kehittää nopeammin ja kustannustehokkaammin laadukkaita ohjelmistoja. Vastauksena tähän yritykset rekrytoivat pätevää ohjelmistoalan työvoimaa sieltä missä sitä on saatavilla kustannuksiltaan edullisesti tai sieltä mistä ylipäättään vapaita ja osaavia resursseja löytyy. Tämän seurauksena globaali ja hajautettu ohjelmistokehitys yleistyvät jatkuvasti. Maantieteelliset välimatkat, kulttuurierot ja eri aikavyöhykkeiden väliset aikataululliset erot tuovat omat haasteensa globaaliin ja hajautettuun ohjelmistokehitykseen. Perinteiset ohjelmistotuotannon prosessimallit on päivitettävä vastaamaan hajautetun ohjelmistokehityksen tarpeita. Tämän artikkelin tarkoituksena on tarkastella prosessimallien ongelma-alueita, parannusehdotuksia ja muita mukautuskeinoja globaalin ja hajautetun ohjelmistokehityksen tueksi. Luvussa 2 esitellään lyhyesti globaalin ja hajautetun ohjelmistokehityksen sanastoa ja erityispiirteitä perinteiseen ohjelmistokehitykseen verrattuna. Luvussa 3 käydään läpi hajautetun ohjelmistokehityksen prosessimallien tutkimusta, prosessien kypsyysmalleja sekä niihin liittyviä ongelmia ja parannusehdotuksia.

## 2 Globaali ja hajautettu ohjelmistokehitys

Mitä globaalilla ja hajautetulla ohjelmistokehityksellä oikeastaan tarkoitetaan? On hyvä muistaa, että termit eivät ole täysin yksiselitteisiä ja aihe on vielä ohjelmistokehityksessä nuori. Carmel & Tijia:n mukaan hajautettu ohjelmistokehitys on alkanut 90-luvun alkupuolella, mutta kasvanut merkittäväksi tekijäksi vasta vuosituhaten vaihteen tienoilla ja sen jälkeen[CT05, 25].

Rafael Pridladnicki ja kumppanit määrittelevät hajautetuksi ohjelmistokehitykseksi minkä tahansa projektin, jossa yksikin kehitysryhmän jäsen työskentelee maantieteellisesti erillään muista[PAS10].

Globaalissa ohjelmistokehityksessä jonkun ryhmän jäsenistä tulee työskennellä toisessa maassa tai vaihtoehtoisesti ohjelmiston tilannut ja toteuttava yritys toimivat eri maissa. Terminä globaali ohjelmistokehitys kattaa useita eri merkityksiä, esimerkiksi sisäinen ulkoistaminen ja globaali ulkoistaminen, joista ensimmäinen tarkoittaa yrityksen

ohjelmistokehityksen siirtämistä yrityksen sisällä toisen maan tytäryhtiöön tai toimipisteeseen ja jälkimmäinen ohjelmistopalveluiden ostamista ulkopuoliselta toimijalta maan rajojen ulkopuolelta[PAS10].

Termien merkityksellä ja käytöllä on huomattava vaikutus tutkittavaan aihealueeseen. Jos esimerkiksi puhutaan globaalin ohjelmistokehityksen prosessimalleista sisäisen ulkoistamisen tai globaalin ulkoistamisen näkökulmista, voi tämä vaikuttaa lopputulokseen. Tässä seminaarityössä tarkastellaan lähinnä sisäistä ulkoistamista sekä hajautettua ohjelmistokehitystä aiemman määritelmän mukaisesti.

James D. Herbsleb'n mukaan hajautetun ohjelmistokehityksen keskeinen ongelma on etäisyyden hallinta ja tämän ratkaisemiseksi on keskityttävä parantamaan ja muokkaamaan kommunikaatiota, työvälineitä, prosesseja ja käytäntöjä[HER07].

### **3 Hajautetun ohjelmistokehityksen prosessien kypsyysmallit**

Tässä luvussa käsitellään ensin hajautetun ohjelmistokehityksen prosessimalleista ja prosessien kypsyysmalleista tehtyä tutkimusta yleisellä tasolla. Tämän jälkeen käydään lyhyesti läpi CMM:n periaatteet ja esitellään hajautettua ohjelmistokehitystä varten luotuja prosessien kypsyysmalleja.

#### **3.1 Tutkimus**

Vaikka hajautettu ohjelmistokehitys on yleistynyt huomattavasti viimeisen 10 vuoden aikana, on erityisesti hajautettuun ohjelmistokehitykseen tarkoitettuja prosessimalleja esitelty hyvin vähän. Prikladnicki, Rafael, Jorge Luis Nicolas Audy ja Forrest Shull toteavat 2010 julkaistussa artikkelissaan "Patterns in effective distributed software development": ”Päästäkseen alkuun hajautetussa ohjelmistokehityksessä yritykset tarvitsevat perusprosessimalleja joiden avulla ne voivat ottaa käyttöön testattuja tehokkaita käytäntöjä ohjelmistokehitysryhmien väliseen johtamiseen, jotta ne pystyvät tuottamaan ohjelmistoja aikataulussa ja budjetissa. Kuitenkin tällaiset prosessimallit puuttuvat vieläkin.”

Tutkimus on keskittynyt suurimmalta osin kypsyyssmallien mukauttamiseen hajautettuun ohjelmistokehitykseen kuten esimerkiksi Ramasubbu, Narayan, Mayuram S. Krishnan ja Prasad Kompalli vuonna 2005 julkaistussa artikkelissaan ”Leveraging global resources: A process maturity framework for managing distributed development.” esittelevät. Prikladnicki, Rafael, Daniela Damian ja Jorge Lis Nicolas Audy analysoivat järjestelmällisesti tutkimuksia läpi hajautetun ohjelmistokehityksen prosessimalleista vuonna 2008 julkaistussa artikkelissaan "Patterns of evolution in the practice of distributed software development: quantitative results from a systematic review.":

Table 6. Analysis of the evolution type

	Offshore Outsourcing						Internal Offshoring						Offshoring						Others						Total			
	Proposal			Need			Proposal			Need			Proposal			Need			Proposal			Need						
	C	M	S	C	M	S	C	M	S	C	M	S	C	M	S	C	M	S	C	M	S	C	M	S				
Business	2	1	-	3	1	-	1	2	-	-	-	-	2	-	1	1	-	-	-	-	-	-	-	-	-	-	-	14
Both	1	-	-	2	1	-	-	-	-	1	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	6
Technical	-	-	-	1	-	-	-	-	-	-	-	1	-	-	1	-	-	3	-	-	-	-	-	-	3	-	1	10
Total	3	1	-	6	2	-	1	2	-	1	1	-	2	1	1	5	-	-	-	-	-	-	-	-	3	-	1	30
%	75	25	-	75	25	-	33	67	-	50	50	-	50	25	25	100	-	-	-	-	-	-	-	-	75	-	25	-

Kuva 3.1[PDA08a]

Kuvassa 3.1 esitetään läpikäytyt artikkelit hajautetun ohjelmistokehityksen eri tyyppien mukaisesti. Kaaviossa ”C” tarkoittaa ”Capability” eli prosessien kyvykkyyttä käsittelevien artikkeleiden määrää, ”M” on ”Maturity” eli prosessien kypsyyden ja ”S” on ”Stage” eli vaihemalleja käsittelevien artikkeleiden määrä. Vaihemallit ovat puitemalleja joilla kuvataan prosessin tai prosessien kypsyyden kehittymistä eri vaiheissa. Kuten kaaviosta pystyy helposti päättämään, on hajautetun ohjelmistokehityksen vaihemalleja tutkittu suppeasti.

Perinteisten prosessimallien tutkimusta hajautetussa tai globaalissa ohjelmistotuotannossa vaikuttaisi olevan tutkittu erittäin vähän tai tietoa niiden tutkimuksesta ei ole helposti saatavilla. Todennäköisesti tähän vaikuttaa se, että perinteiset prosessimallit kuten vesiputousmalli tai spiraalimalli eivät niinkään ota kantaa toteutuksen hallinnointiin vaan pikemminkin prosessivaiheiden järjestykseen suunnittelussa ja toteutuksessa. Poikkeuksen tähän tekee ketterät ohjelmistokehityksen prosessimallit. Ketterässä ohjelmistokehityksessä onkin selkeästi kyse enemmän hallinnollisesta tavasta johtaa ohjelmistotuotantoa kuin perinteisissä prosessimalleissa. Muutamaksi esimerkiksi scrumin soveltamista hajautettuun ohjelmistokehitykseen

tutkivat Jeff Sutherland ja kumppanit vuonna 2007 julkaistussa artikkelissaan "Distributed scrum: Agile project management with outsourced development teams." sekä Emam Hossain ja kumppanit vuonna 2009 julkaistussa artikkelissaan "Using Scrum in global software development: A systematic literature review.". Layman ja kumppanit käsittelevät puolestaan hajautettua extreme programmingia 2005 julkaistussa artikkelissaan . "Essential communication practices for Extreme Programming in a global software development team."

Yllä kuvatusta voidaan päätellä, että ketteriä menetelmiä lukuun ottamatta ohjelmistokehityksen prosessimallit eivät ole erikseen standardisoituneet hajautettuun ohjelmistokehitykseen tai ne eivät ota niin voimakkaasti kantaa asioihin ja haasteisiin, joita hajautettu ohjelmistokehitys tuo mukanaan, vaan näiden sopeuttamista hajautettuun ohjelmistokehitykseen on helpompaa lähestyä kypsyyssmallien kautta. Todennäköistä myös on, että suurilla kansainvälisillä toimijoilla on omat sisäiset prosessimallit käytössään ja ne mukailevat löyhästi yleisesti tunnettuja prosessimalleja, mutta niistä ei ole tietoa julkisesti saatavilla.

## **3.2 Kypsyysmallit**

Yleisesti käytössä olevat ja testatut ohjelmistoprosessikehykset, kuten Software Engineering Institutun Capability Maturity Model ja ISO 9001, kuvailevat avainprosessialueet ohjelmistokehitykselle ja kypsyystasot. Jokainen avainprosessialue on pieni yksikkö, joka sisältää yhtenäisen joukon toisiinsa liittyviä tehtäviä [RKK05].

Yleisesti yhteistä prosessien kypsyyssmalleille on se, että ne pyrkivät pilkkomaan prosessit osiin ja mittaamaan eri osa-alueiden toteutumista ja kuvaamaan näiden perusteella prosessien kypsyystasoa.

### **3.2.1 CMM**

Karl Wieger kuvailee CMM:n perusominaisuudet hyvin artikkelissaan ”Misconceptions of the Capability Maturity Model ” vuodelta 1996. CMM:n viisi kypsyystasoa kuvaavat kehitystä korkeamman kyvykkyyden ohjelmistotuotantoprosesseihin. Nämä kypsyystasot ovat lähtötaso, toistettava, määritelty, johdettu ja optimoitu. Ensimmäistä tasoa lukuun ottamatta kaikki kypsyystasot vaativat määritellyn joukon avainprosessialueita täytettäväksi hyväksyttävällä tasolla ennen tason saavuttamista.

Jokainen avainprosessialue sisältää kahdesta neljään päämäärää, jotka tulee saavuttaa ennen kuin se voidaan määritellä hyväksyttävästi suoritetuksi. Tämän lisäksi jokainen avainprosessialue sisältää joukon avainkäytäntöjä, joiden tulee toteutua, jotta päämäärät voidaan katsoa saavutetuiksi. Kun tietyt avainprosessialueet on hyväksyttävästi suoritettu, saavutetaan sen mukainen taso CMM: n viisiportaisella asteikolla oheisen kuvan mukaan:

<b>Maturity Level</b>	<b>Key Process Area</b>
1: Initial	None
2: Repeatable	Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Subcontract Management, Software Quality Assurance, Software Configuration Management
3: Defined	Organization Process Focus, Organization Process Definition, Training Program, Integrated Software Management, Software Product Engineering, Intergroup Coordination, Peer Reviews
4: Managed	Quantitative Process Management, Software Quality Management
5: Optimizing	Defect Prevention, Technology Change Management, Process Change Management

Kuva 3.2 CMM:n viisi kypsyytstasoa ja niiden sisältämät avainprosessialueet[Wie96]

CMM:stä kuitenkin puuttuu avainprosessialueet, jotka tuovat kyvyn hallita hajautettuja ohjelmistoprojekteja[RKK05].

### 3.2.2 A Process Maturity Framework for Managing Distributed development

Ramasubbu ja kumppanit esittelevät vuonna 2005 julkaistussa artikkelissaan ”Leveraging Global Resources: A Process Maturity Framework for Managing Distributed Development” 24 avainprosessialuetta vastaamaan hajautetun ohjelmistokehityksen tarpeita. Nämä avainprosessialueet on rakennettu neljän eri käsitteen ympärille: Yhteistyövalmius, yhteinen tietämys, työn kytkennät ja teknologiavalmius.



**Our key process areas ranked by an expert review committee  
and compared to our model's rankings**

KPAs in our model	Maturity phase in our model	Expert rankings for level 1 match	Expert rankings for level 2 match	Expert rankings for level 3 match	Correct (%)
Belief and willingness	1	26	8	—	76.47
Personnel communication skills	1	30	4	—	88.24
Utilization of distributed technology infrastructure	1	30	4	—	88.24
Critical mass (team size)	1	28	5	1	82.35
Understanding cultural differences	1	30	3	1	88.24
Managerial training	1	17	17	—	50.00*
Setting shared business goals	1	27	3	4	79.41
Tailoring business goals	1	27	3	4	79.41
Budgeting and cost structures	1	28	—	6	82.35
Devising mechanisms for division of labor	2	28	6	—	82.35
Product development tools and processes	2	—	26	8	76.47
Consistency in project management processes	2	—	33	1	97.06
Ownership and responsibilities	2	—	30	4	88.24
Knowledge transfer	2	2	30	2	80.24
Top-management communication channels	2	13	6	15	17.65*
Performance monitoring	3	1	7	26	76.47
Managing complexity	3	1	3	30	88.24
Managing social networks	3	—	3	31	91.18
Enabling social communication via technology	3	—	5	29	85.29
Nurturing and leveraging core competencies	3	—	9	25	73.53*
Interorganizational innovation management	3	—	—	34	100.00
Best practices management	3	—	5	29	85.29
Contract management and nurturing partnership	3	—	4	30	88.24
Managing symbiotic relationship and continuous development	3	—	5	29	85.29

*Average model acceptance – 80.43%*

*\*Below cutoff value of 75% of expert acceptance*

**Kuva 3.3 Hajautetun kypsyysmallin avainprosessialueet ja kypsyystasot[RKK05]**

Taulukkoon on koottu Ramasubbun ja kumppaneiden kypsyysmallin avainprosessialueet ja niitä vastaavat kypsyysvaiheet. Kolme avainprosessialuetta ei saanut tukea tutkimuksesta [RKK05].

Yhteistyövalmiudella tarkoitetaan yleistä ohjelmistokehityksen hallinnointimallia, jonka avulla kehitys pystytään pilkkomaan liiketoimintatavoitteiksi, jotka muuntuvat helposti hallinnoitaviksi toisistaan riippumattomiksi kehityspalasiksi, jotka pystytään toteuttamaan maantieteellisesti erillisissä kehitysryhmissä. Yhteistyövalmiudesta pystytään johtamaan avainprosessialueita, jotka pyrkivät muodostamaan hallintamallien, sopimusmallien ja liiketoimintatavoitteiden formalisoinnin ja jatkuvan kehityksen[RKK05].

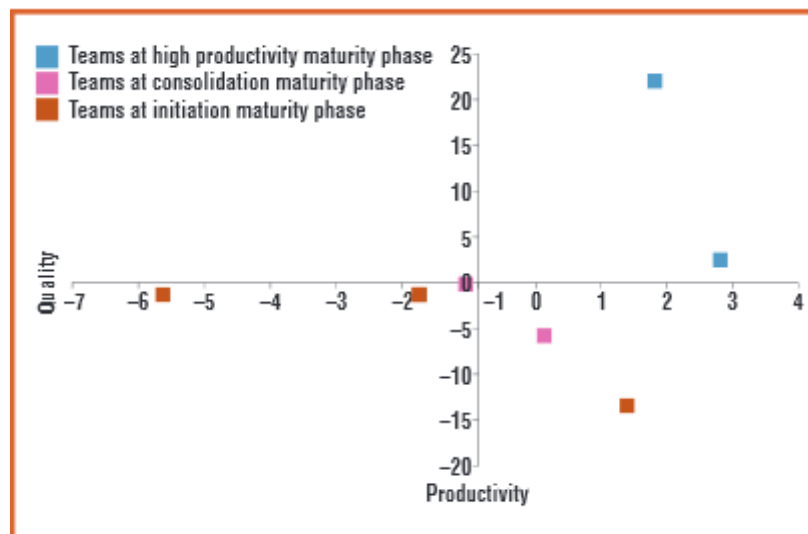
Yhteinen tietämys heijastaa eri paikoissa työskentelevien resurssien välillä jaettua

tietämystä. Tästä voidaan johtaa avainprosessialueet tiedonjakamiseen, sosiaaliseen kanssakäymiseen ja tietämyksen johtamiseen[RKK05].

Työn kytkennällä tarkoitetaan mekanismeja joilla voidaan jakaa työ hajautettuihin yksiköihin. Työn kytkentä korostaa työn jakamisen avainprosessialueiden merkitystä [RKK05].

Teknologiavalmius viittaa kehitysympäristöön ja henkilöstön valmiuteen sekä kykyyn käyttää yhteistyövälineitä kuten kommunikaatiojärjestelmiä, videokonferenssijärjestelmiä sekä hajautettua versionhallintaa. Teknologiavalmius tuo avainprosessialueisiin tarpeen valvoa ja mitata eri hajautusinfrastruktuurin käyttöä sekä parantaa sitä[RKK05].

Ramasubbu ja kumppanit jakoivat prosessien kypsyyden avainprosessialueiden toteutumisen mukaan kolmeen tasoon: lähtötaso, lujitusvaihe ja korkean tuottavuuden vaiheeseen. Avainprosessialueet jaettiin useaan avainkäytäntöön CMM:n tyypisesti ja niiden toteutumista arvoitiin seitsemän portaisella asteikolla. Tämän jälkeen ryhmä arvioi seitsemän kansainvälistä ohjelmistoprojektiryhmää. Haastattelujen ja läpikäyntien tulosten perusteella he jakoivat tiimit kypsyystasoihin: kaksi tiimiä korkean tuottavuuden, kaksi lujitusvaiheen ja kolme lähtötason kypsyysasteen. Tämän lisäksi he vertasivat tiimien lopputuotteiden virhetiheyttä ja toimintopisteitä luokittelemaansa hajautetun prosessimallin kypsyystasoon ja tulos vastasi näitä kypsyystasoja kuten seuraavasta kaaviosta näkyy:



Kuva 3.4 Tutkittujen tiimien sijoittuminen kypsyyssmallissa[RKK05]

Ramasubbun ja kumppaneiden kypsyyssmalli keskittyy CMM:ä huomattavasti enemmän projektin johtamiseen ja hallinointiin kuten kuvan 3.3 avainprosessialueista voidaan päätellä verrattuna kuvan 3.2 vastaaviin CMM:n avainprosessialueisiin.

### 3.2.3 A Preliminary capability model

Rafael Prikladnicki ja kumppanit julkaisivat 2008 ilmestyneessä artikkelissaan, "Patterns of evolution in the practice of distributed software development in wholly owned subsidiaries: A preliminary capability model.", oman hajautetun ohjelmistokehityksen prosessien kehitystä kuvaavan mallin. Malli on kehitetty kuvaamaan ulkomaisen tytäryhtiön kypsyyss- ja luotettavuustasoa kolmiportaisella asteikolla. Prikladnicki ja kumppanit ehdottavat mallissaan 28 avainattribuuttia tässä kehityksessä. Kuvassa 3.5 on listattu nämä ominaisuudet ja yhden tutkimuksessa mukana olleen ulkomaille perustetun tytäryhtiön kehityslukemat haastatteluihin perustuen.

Attribute	Evolve			Type	p-Value
	Y	N	%		
Cultural differences	13	0	100	People	0.0001
Trust acquisition	11	2	85	People	0.0112
Awareness of activities	12	1	92	People	0.0017
Awareness of process	13	0	100	People	0.0001
Awareness of availability	10	3	77	People	0.0461
Knowledge management	13	0	100	People	0.0001
Levels of dispersion	10	2	83	People	0.0193
Learning	12	1	92	People	0.0017
Training	11	2	85	People	0.0112
Skill development	8	4	67	People	0.1938
Perceived distance	9	4	69	People	0.1334
Requirements specification	9	4	69	Project	0.1334
Communication tools	10	3	77	Project	0.0461
Collaboration tools	10	3	77	Project	0.0461
Measurement	10	3	77	Project	0.0461
Infrastructure	10	2	83	Project	0.0193
Project mgmt activities	13	0	100	Project	0.0001
Sw development Life cycle	12	1	92	Project	0.0017
Risk management	12	1	92	Project	0.0017
Estimation	11	2	85	Project	0.0112
Configuration management	13	0	100	Project	0.0001
Project activities (dependency)	12	0	100	Project	0.0002
Type of projects	11	2	85	Portfolio	0.0112
Project allocation	10	3	77	Portfolio	0.0461
Sw process improvement	12	1	92	Subsidiary	0.0017
Relationship w/ headquarters	7	4	64	Subsidiary	0.2744
Standards	10	2	83	Subsidiary	0.0193
Organizational structure	12	1	92	Subsidiary	0.0017

Kuva 3.5 Kyvykkyysmallin avainattribuuttialueet[PDA08b]

Näiden attribuuttien tarkemman tason analyysi on tutkijoilla vielä artikkelin julkaisuajankohtana työnalla.

## 4 Yhteenveto

Hajautettu ohjelmistokehitys on erittäin nuori ja nopeasti levinnyt ilmiö, jonka tutkimus on vielä alkutekijöissään. Hajautetun ohjelmistokehityksen prosessimallit ovat selkeästi vasta lapsenkengissä. Hajautettuun ohjelmistokehitykseen liittyvästä tutkimuksesta ja kirjallisuudesta voidaan kuitenkin päätellä, että selkeimpänä muutoksena perinteiseen ohjelmistokehitykseen on uuden tason tuominen aiempien prosessien päälle. Kuten monella muullakin alalla ohjelmistokehityksessä ja sen ulkopuolella, myös hajautetussa

ohjelmistokehityksessä standartoidut menetelmät ja prosessimallit kehittyvät todennäköisesti laajalle levinneiden käytäntöjen perusteella enemmän kuin tarkasti tutkittujen ja tieteellisesti kehitettyjen menetelmien kautta. Tämä ei kuitenkaan tarkoita sitä, että näitä prosessimalleja ei kannattaisi kyseenalaistaa ja analysoida jatkuvasti paremman lopputuloksen saavuttamiseksi. Tähän ongelmaan ovat kypsyysmallit erinomainen väline. Kypsyysmallien avulla käytössä olevia prosessimalleja voidaan helpommin pilkkoa osiin ja mitata niiden kehitystä ja kypsymistä eri tilanteissa. Standartoidut prosessimallit myöskään harvemmin soveltuvat sellaisenaan eri projekteihin. Kypsyysmallit helpottavat myös tätä soveltamista pilkkomalla prosesseja helpommin mitattaviin osiin ja antamalla täten paremmat eväät valita oikeat prosessialueet kyseisen projektin tarpeisiin. Ehdotetuissa uusissa kypsyysmalleissa korostuvat uusissa osa-alueissa selkeästi johtamiseen ja kommunikaatioon liittyvät alueet vanhojen lisäksi.

## Lähteet

- [CT05] Carmel, E. ja Tjia, P., *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*, Cambridge Univ. Press, 2005.
- [PAS10] Prikladnicki, Rafael, Jorge Luis Nicolas Audy, and Forrest Shull. "Patterns in effective distributed software development." *Software, IEEE* 27.2 (2010): 12-15.
- [Her07] Herbsleb, James D. "Global software engineering: The future of socio-technical coordination." *2007 Future of Software Engineering*. IEEE Computer Society, 2007.
- [PDA08a] Prikladnicki, Rafael, Daniela Damian, and Jorge Lis Nicolas Audy. "Patterns of evolution in the practice of distributed software development: quantitative results from a systematic review." *Proceedings of the 12th Conference on Evaluation and Assessment in Software Engineering (EASE'08)*. 2008.
- [RKK05] Ramasubbu, Narayan, Mayuram S. Krishnan, and Prasad Kompalli. "Leveraging global resources: A process maturity framework for managing distributed development." *Software, IEEE* 22.3 (2005): 80-86.
- [Sut07] Sutherland, Jeff, et al. "Distributed scrum: Agile project management with outsourced development teams." *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*. IEEE, 2007.
- [HMH09] Hossain, Emam, Muhammad Ali Babar, and Hye-young Paik. "Using Scrum in global software development: A systematic literature review." *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*. Ieee, 2009.
- [JKHH04] Jiang, James J., et al. "An exploration of the relationship between software development process maturity and project performance." *Information & Management* 41.3 (2004): 279-288.
- [PDA08b] Prikladnicki, Rafael, Daniela Damian, and J. Audy. "Patterns of evolution in the practice of distributed software development in wholly owned subsidiaries: A preliminary capability model." *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*. IEEE, 2008.
- [CAR05] Carmel, Erran. "The offshoring stage model: An epilogue." *Available online*

*at auapps.american.edu/~carmel/papers/epilogue.pdf* (2005).

[LWDB05] Layman, Lucas, et al. "Essential communication practices for Extreme Programming in a global software development team." *Information and software technology* 48.9 (2006): 781-794.

[Wie96] Wiegers, Karl. "Misconceptions of the capability maturity model." *Software Development Magazine* 4.11 (1996): 57-64.