

Tallinna Pedagoogikaülikool
Matemaatika-Loodusteaduskond
Informaatika õppetool

Sander Zeemann

Tarkvaraprotsessi küpsuse
hindamise ja arendamise võimalusi
Capability Maturity Model'i näitel

Proseminaritöö

Juhendaja: prof. Peeter Normak

Tallinn 2002

Sisukord

I CAPABILITY MATURITY MODEL FOR SOFTWARE.....	5
SW-CMM'i ajalugu.....	5
Ülevaade SW-CMM'ist.....	5
CMM'i probleemid ja puudused	7
Protsessi küpsus ja selle indikaatorid.....	7
Tarkvaraprotsessi küpsuse hindamine – kellel seda vaja ja kes seda läbi viivad	9
CMM'i põhiprintsiibid	10
CMM'i tulevikusuunad.....	10
II CMM'I VÕRDLUS ISO JA XP METOODIKATEGA.....	11
CMM vs. ISO 9000	11
CMM vs. 15504 (SPICE)	12
CMM vs. XP.....	13
III SPI RAKENDAMISE TULEMUSED	16
KOKKUVÕTE TÖÖST	19
LISAD.....	21
Lisa 1. Akronüümid	21
Lisa 2. Sõnaseletused.....	21
KASUTATUD KIRJANDUS	22

Sissejuhatus

Tarkvara parim omadus on selle paindlikkus: seda saab programmeerida tegema ükskõik mida. Tarkvara halvim omadus on samuti selle paindlikkus – omadus ‘ükskõik mida’ on teinud tarkvaraarenduse raskesti planeeritavaks, jälgitavaks ja juhitavaks (W. Royce).

Üha enam sõltuvad igasugused majandusharud tarkvarast. Tarkvaraarendusfirmade jaoks on see muidugi hea uudis. Halb uudis on aga see, et nende süsteemide muutumine suuruse, kompleksuse, hajutatuse ja tähtsuse suunas sunnivad järjest rohkem laiendama nende teadmisi selle kohta, kuidas neid süsteeme välja töötada. Vanad meetodid ja tehnoloogiad ei ole tänapäeval just eriti otstarbekad. Püüd arendada aegunud süsteeme moodsa tehnoloogia suunas toob endaga kaasa veel omad probleemid. Me oleme jõudnud olukorda, kus ärimaailm eeldab tarkvaratooteid kasutades üha suuremat produktiivsust, funktsionaalsust ja paremat kvaliteeti, samal ajal oodates uute tarkvaratoodete kiiremat väljatöötamist ja juurutamist.

Tähtaegadest kinnipidamine on muutunud tõsiseks katsumuseks - levinud on isegi ütlus “parem on teha funktsionaalsuse ja kvaliteedi osas järeleandmisi, kui et kogu projekt viibib.” Probleemi teravdab asjaolu, et kvalifitseeritud tarkvaraspetsialistide juurdekasv ja nõudlus ei ole tasakaalus; eriti suur puudus on veel laia silmaringiga ja märkimisväärsete kogemustega visiooniga inimestest. Tulemus on see, et tarkvara arendamine ja hooldamine on raske ja muutub järjest raskemaks. Kvaliteetse tarkvara arendamine korratavalt ja ennustatavalt on veelgi raskem.

Enne kui hakata uuele protsessile üle minema tuleks kõigepealt mõista senist olukorda ja miks selline olukord üldse tekkida sai. 90-ndate keskel tehtud põhjalikumad analüüsid tarkvaraarenduse hetkeseisust jõudsid Walker Royce’i andmetel kõik samadele üldistele järeldustele:

- Projektide edenemine on väga ennustamatu – ainult 10% projektidest valmib tähtajaks ja ei ületa eelarve piire.
- See, et mingi projektijuhtimise teooria aitab graafikus ja eelarves püsida, on pigem juhus kui reegel.
- Tarkvara ümbertegemiste maht ja viis on teiste valdkondadega võrreldes omane ainult ebaküpsele protsessile.

(Walker Royce, “Software Project Management: A Unified Framework”, 1998, lk. 5)

Selline seis on valitsenud tarkvaraarenduse sektoris juba viimased kolm aastakümnet ja sellest räägitakse kui “tarkvaratootmise kriisist”.

Oma töös üritan ma anda ülevaate tarkvaraprotsessi arendamise ja küpsuse hindamise võimalustest, nende kasulikkusest organisatsioonile, samuti võrdlen kõige tuntumat ja tunnustatumat küpsusmudelit teiste mudelite ja metodoloogiatega. Kesksel kohal on Carnegie Mellon University Software Engineering Institute'is välja töötatud suutvusküpsusmudel ehk CMM. Valdav osa töö kirjutamisel aluseks võetud materjalist pärineb Walker Royce'i raamatust "Software Project Management: A Unified Framework" ja erinevatest SEI publikatsioonidest.

Töö põhiosad:

I peatükk kirjeldab Capability Maturity Model for Software'i (SW-CMM): ajalugu, lühiülevaadet, struktuuri, põhiprintsiipe, eesmärke, probleeme, puudusi, tulevikku jne.

II peatükis võrdlen CMM'i ISO ja XP metoodikatega.

III peatükk annab ülevaate tarkvaraprotsessi arenduse (Software Process Improvement - SPI) kasulikkusest organisatsiooni jaoks ning mida SPI rakendamine organisatsioonis endaga kaasa toob.

I Capability Maturity Model for Software

Tarkvaraprotsessi kvaliteedi hindamisel kasutatakse sageli Carnegie Mellon University Software Engineering Institute'i poolt välja töötatud maailmatasemel tunnustatud SW-CMM mudelit. Käesolev peatükk keskendubki suuremas osas just SW-CMM'ile (versioon 1.1).

SW-CMM'i ajalugu

Novembris 1986 hakkas SEI koos Mitre Corporation'iga välja töötama tarkvaraprotsessi küpsuse hindamise ja parendamise raamsüsteemi, et aidata organisatsioonidel tõsta oma tarkvaraprotsessi taset. Asja vastu hakkas esimesena huvi tundma USA valitsus, et oleks mingi meetod, mille abil saaks hinnata nende jaoks mingit tarkvara välja töötavate erafirmade suutlikust.

Septembris 1987 avaldas SEI esimese versiooni protsessi küpsuse hindamise raamsüsteemist ja sellega koos kasutamiseks ka seda toetava küsimustiku .

Nelja järgneva aasta jooksul arendati seda raamsüsteemi edasi; suureks abiks oli erinevatelt organisatsioonidelt ja ka valitsuselt saadud vastukaja. Aastaks 1991 valmis CMM versioon 1.0. Pidev arendustöö CMM'i kallal jätkus ja selle tulemusena valmis 1993. aastal versioon 1.1, millel põhineb ka antud peatükk.

Ülevaade SW-CMM'ist

SEI poolt välja töötatud suutvusküpsuse mudel (*Capability Maturity Model – CMM*) on levinud mall tarkvaraarendusprotsessi küpsuse hindamiseks. See on üks peamisi ja ka tuntumaid küpsusmudeleid firmade tarkvaraprotsessi hindamiseks ja arendamiseks.

Küpsustasemed – 5

↳ Protsessi võtmepiirkonnad – 18

↳ Eesmärgid – 52

↳ Võtmetegevused – 316

↳ Alamtegevused ja näited – palju-palju

Joonis 1. CMM'i struktuur.

CMM defineerib arendusprotsessi küpsusele viis taset, mis näitavad, milliseid protsessi võtmepiirkondi (*key process area – KPA*) vaadeldav organisatsioon toetab.

Tase 1. Algne tase – KPA'sid pole. Protsess on ebaküps, määratlemata.

Tase 2. Protsess on korratav. 2. taseme KPA'd: nõuete haldus, projekti planeerimine, tarkvaraprojekti jälgimine ja ülevaadete tegemine, alltöövõtude haldus, tarkvara kvaliteedikontroll, tarkvara konfiguratsioonihaldus.

Tase 3. Protsess on defineeritud. 3. taseme KPA'd: organisatsiooni fookus protsessil kui tervikul, organisatsioonipoolne protsessi defineerimine, koolitusprogrammid, terviklik tarkvaraarenduse juhtimine, tarkvaratoote arendus, gruppidevaheline koordineerimine, vastastikused läbivaatused.

Tase 4. Protsessi juhitakse. 4. taseme KPA'd: protsessi hindamine ja analüüs, kvaliteedijuhtimine.

Tase 5. Protsessi optimeeritakse. 5. taseme KPA'd: tehnoloogia innovatsioon, protsessi muudatuste haldus, toote defektide ärahoidmine.

Enamik edukusele pretendeerivaid tarkvarafirmasid peaksid endale eesmärgiks seadma protsessi 3. CMM taseme. Arendusprotsessi suutlikkuse hindamise (*software capability evaluation – SCE*) käigus tavaliselt tuvastatakse, kas “organisatsioon ütleb, mida ta teeb ja teeb, mida ta ütleb”, uurides selleks organisatsiooni sisepoliitikat ja projektide praktikat. Neid hinnatakse KPA raamistiku järgi. Hindamisprotsess ei ole perfektne, aga hea suhteline indikaator.

Tüüpiline SCE kasutab SEI CMM'i küsimustikku (SEI Maturity Questionnaire) kui osa põhjalikust auditist. Küsimustikku kasutatakse reeglina enne hindamise juurde asumist konteksti tekitamiseks. Hindamisprotsess sisaldab endas detailseid analüüse, intervjuusid jne.

Tarkvaraorganisatsioonide jaotuse kohta CMM'i tasemete järgi on tehtud palju erinevaid uuringuid. Järgnev tabel annab umbkaudse pildi tarkvaraorganisatsioonidest 1995. aastal. Need on muidugi umbkaudsed arvud, kuna kõiki tarkvaraarendusega tegelevaid firmasid pole

hinnatud. SEI andmete järgi (mai, 2001) hindamise läbinud firmadest 9% on 4. ja 5. tasemel (kokku 132; nendest umbes pooled paiknevad Indias), umbes 20% kuulub 3. tasemele.

CMM tase	Sagedus	Käitumise tase
1. Algne	70%	Ennustamatu, kõrge risk
2. Korratav	15%	Raskustes, aga jäävad ellu
3. Defineeritud	<10%	Stabiilne, ennustatav, arenev
4. Juhitav	<5%	Väga kindel, usaldusväärne
5. Optimeeritav	<1%	Kestvalt arenev

Tabel 1. Tarkvaraorganisatsioonide jaotus CMM-i skaalal aastal 1995. (Walker Royce, "Software Project Management: A Unified Framework", 1998, lk. 364)

CMM'i probleemid ja puudused

Üks peamisi SEI CMM'i puudusi on see, et KPA-d keskenduvad peamiselt klassikalises protsessis eksisteerivatele dokumendi kujul tehisele (*artifacts*): arhitektuur, nõudmised, allhangete lepingud, lepingud, plaanid ja raportid. Väga vähe KPA-sid puudutavad arenevaid väljatöötatud tehiseid (*engineering artifacts* - nõuete mudelid, arhitektuuri mudelid, lähtekood, programmid), protsessi automatiseerimise taset keskkonnas jms.

Teine puudus on CMM'ile omane viis kirjeldada konfiguratsioonihaldust ja kvaliteedikontrolli mitte teiste protsessi tegevuste juurde kuuluvatena, vaid kui kõikidest teistest protsessi tegevustest eraldiseisvaid distsipliine.

Protsessi küpsus ja selle indikaatorid

Praktikas on protsessi küpsuse tegelikuks indikaatoriks projekti käitumise ennustatavuse tase. Projekti käitumise kolme näitaja suhtes (projekti hind, ajagraafik ja tarkvarakvaliteet) ja CMM-i tasemete vaheline korrelatsioon peaks näitama järgmisi trende:

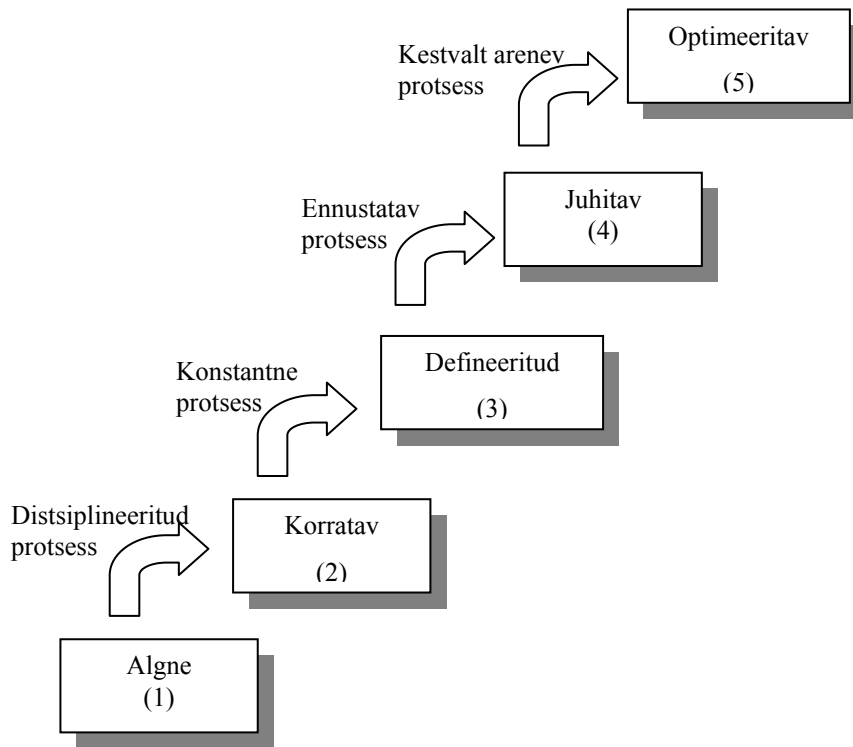
Tase 1. Protsess on juhuslik ja käitub ennustamatult.

Tase 2. Protsess on erinevates projektides korratav. Sarnaste projektide tase on suhteliselt konstantne.

Tase 3. Protsess on defineeritud, projektid käituvad aja möödudes hinna, ajagraafiku või tarkvarakvaliteedi suhtes vähehaaval järjest paremini.

Tase 4. Protsessi juhitakse. Aja möödudes projekti käitumine paraneb kas ühe näitaja suhtes oluliselt või mitme näitaja suhtes korraga (näiteks hinna ja tarkvara kvaliteedi suhtes).

Tase 5. Protsessi optimeeritakse. Iga järgnev projekt käitub oluliselt paremini igas mõttes. Viienda taseme organisatsioonid saavad tõenäoliselt tegutseda väga kitsas nišis.



Joonis 2. CMM'i viis taset.

Kuigi pole välistatud, et paljud organisatsioonid suudaksid tekitada endale näo, mis võimaldab seda organisatsiooni hinnata kolmanda taseme vääriliseks, sealjuures protsess tegelikkuses ei pruugi hea olla, peaks tõesti hea protsess kolmanda taseme hinde kergesti saama. Walker Royce väidab, et omades tosinate protsesside hindamise kogemustest on ta selgeks saanud veel mõned indikaatorid, mis tõeliselt vastavad küpsele protsessile:

- Objektivne arusaamine protsessi praegusest küpsusest.
- Objektivne arusaamine projekti kvantitatiivsest käitumisest hinna ja kvaliteedi suhtes.
- Tegelik projekti käitumise paranemine.
- Aeg, mida vajatakse hindamise ettevalmistamiseks, on minimaalne.

(Walker Royce, "Software Project Management: A Unified Framework", 1998, lk. 366)

Tasemel organisatsioon ja tasemel projektid teavad protsessi ja järgivad seda. Nad ei vaja aega auditi ettevalmistamiseks. Kui te arvate, et teie organisatsioon on kolmandal tasemel, siis vastake küsimusele: Kas Te taluksite üllatusauditit?

Vastavus CMM-ile ei too tingimata kaasa kvaliteetsete toodete arendamist. Küll aga protsess, mille käigus valmivad kvaliteetsed tooted, hinnatakse küpseks. Enamike protsesside paikapandud raamistike peamiseks puuduseks on see, et nad määravad kvaliteedikontrolli programmi staatiliselt kellegi eraldi tööülesandeks selle asemel, et integreerida kvaliteedikontroll dünaamiliselt igapäevaste tööülesannete hulka.

Vale arusaam on see, et küps protsess on kallis - küps protsess ei ole kallis. Pikemas perspektiivis see hoopis säästab raha. Parandades ebaküpset protsessi, organisatsioon sageli tajub muutustega seotud kulutusi – protsessi juurutamine juba jooksva projektil, mida valitsevad lühiajalise perspektiiviga kulukaalutlused, on väga raske. Samas on protsessi juurutamine pikaajaliste äriolude püüdlustega organisatsioonis ja pikaajalistes projektides, mis on alles planeerimise staadiumis, kergem.

Tarkvaraprotsessi küpsuse hindamine – kellel seda vaja ja kes seda läbi viivad

Organisatsiooni tarkvaraprotsessi hindamisest võivad huvitatud olla mitu osapoolt – nii firma ise kui ka tellija. Firma jaoks näitab see muidugi ära mitmendale tasemele ta kuulub CMM'i skaalal, kuid see ei ole minu arust isegi see kõige tähtsam osa. Detailne audit näitab ära organisatsiooni tarkvaraprotsessi tugevad ja nõrgad kohad, mille järgi saab organisatsioon kavandada järgmisi samme oma protsessi parendamiseks. Tellija jaoks aga ongi tähtis just teadmine, mitmendale tasemele vastav ettevõtte kuulub, kuna tal puudub selle ettevõtte siseelu puudutatav informatsioon.

Et audit oleks võimalikult objektiivne ja täpne, viivad seda läbi aastatepikkuse kogemusega sõltumatud spetsialistid. Kuna tõenäosus, et esimesel katsel kogemuste puudumisel võib selline audit ebaõnnestuda, on väga suur, on soovitatav alguses tellida eelaudit, mis on pisut kergekaalulisem ning mille tulemusel antakse soovitusi, kuidas olukorda parandada. Seejärel tuleb tegelik audit. USA's ja ka mujal maailmas viivad neid auditeid läbi mitmed institutsioonid, millest üks on ka SEI. Eestis üritab seda teenust pakkuma hakata minu teada Cybernetica AS ja tõenäoliselt ka KPMG.

CMM'i põhiprintsiibid

1. CMM on pidevalt täiustuva tarkvaraprotsessi kirjeldus.
2. CMM asetab rõhu võimekuse suurendamisele.
3. CMM järgib defineeritud arhitektuurilisi ja süntaktilisi juhtnööre.
4. CMM on mudel, mis ei põhine ainult teoorial, vaid ka praktilistel kogemustel.
5. 5 ei ole maagiline number. Kuigi CMM koosneb viiest tasemest, võib selle kohandamisel või isegi uue dimensiooni lisamisel antud struktuur muutuda ebasobivaks.
6. CMM ei ütle, kuidas midagi teha, vaid mida oodatakse lõpptulemuselt.
7. CMM ei käsi kasutada spetsiifilisi vahendeid (arendusvahendid, meetodikad, tehnoloogiad...).
8. Ole arukas! CMM'i soovitatakse kasutada asjatundjatel, kellel on aastatepikkune praktiline kogemus. CMM jagab tegutsemiseks juhiseid, kuid neid ei tohiks 1:1 järgida, vaid kohandada vastavalt olukorrale, organisatsioonile jne.
9. CMM on refleksiivne. Tase 5 ei ole mitte optimeeritud, vaid optimeeriv.
10. CMM kaitseb investeeringuid. CMM keskendub astmelisele arendusele, täiustamisele, eeldusel, et madalamatel tasemetel olevatesse tegevustesse tehtud investeeringud ei tekita vasturääkivusi kõrgematel tasemetel.

CMM'i tulevikusuunad

Järgmise CMM'i versiooni, versioon 2.0 jaoks vaatab SEI üle terve senise CMM'i. Kuigi muutusi võidakse sisse viia kõikidesse tasemetesse, on põhirõhk siiski pandud 4. ja 5. tasemele. 2. ja 3. taseme KPA'd on kõige täpsemalt defineeritud. Kuna 4. ja 5. taseme vääriliseks on hinnatud väga vähesed organisatsioonid, on ka nende karakteristikate kohta vähem teada. Nende tasemete tegevusi täiustatakse koostöös 4. ja 5. tasemele kuuluvate ja nende poole pürgivate firmade abiga. CMM võib muutuda ka multidimensionaalseks adresseerides nii tehnoloogiaga kui ka inimressurssidega seonduvaid küsimusi. (Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles W. Weber, "Capability Maturity Model for Software, version 1.1", 1993, lk. 69)

II CMM'i võrdlus ISO ja XP metoodikatega

CMM ei ole ainuke mudel, mille järgi hakata arendama organisatsiooni tarkvaraprotsessi. Valikuvariante peale CMM'i on hulgaliselt: ISO 900X ja 15XXX seeriad, Six Sigma, Malcolm Baldrige, BootStrap, Trillium jne. Erinevusi neil jagub, kuid üldine eesmärk on siiski sama. Kahjuks ei ole aga enamuse nendest mudelitest tasuta saadaval. Järgnevalt esitan CMM'i võrdluse ISO ja XP metoodikatega.

CMM vs. ISO 9000

Järgnevalt toon välja SW-CMM ja ISO 9000 põhilised erinevused. Erinevused ja sarnasused võrreldes ISO 9001'ga on enam-vähem samad.

- Tasemed. ISO 9000 ei oma tasemeid, CMM'il on aga 5 hästi defineeritud taset.
- Kirjeldus. ISO 9000 on nõuete kogumi spetsifikatsioon (standardid, ettekirjutused), seevastu suurema osa CMM'ist võtab enda alla protsessi kirjeldus.
- Orientatsioon. ISO 9000 on organisatsioonikeskne, eesmärgiks on tagada organisatsiooni kvaliteetne toimimine. CMM on aga rohkem protsessi- ja projektikeskne, eesmärgiks on arendusprotsessi pidev parendamine.
- Rõhuasetus. ISO 9000 nõuab, et defineeritud oleks organisatsioon, mis toimib vastavalt ettekirjutatud nõuetele (kvaliteedi planeerimine ja kontroll – Quality Planning & Control). CMM keskendub pidevale arendustööle.
- Tegevuspiirkond, ulatus. CMM'i huvi on asetunud tarkvaraarendusele, kaudselt ka haldamine. ISO 9000 muretseb võrdselt nii arenduse, varustuse kui ka halduse pärast. ISO 9000 ei ole aga siiski nii konkreetne ja praktiline kui CMM.
- Kohustused, osapooltevahelised suhted. ISO 9000 asetab rõhu tellija-varustaja suhtele, organisatsioonivälisele suhtlemisele kliendi ja firma vahel. CMM hindab jälle organisatsioonisisest suhtlemist, näiteks erinevate divisjonide vahel.
- Protsessi/tehnoloogia vaade. Mõlemad metoodikad katavad tarkvaraarenduses nii protsessi kui ka tehnoloogia kontrolli osa. ISO 9000 puhul pole aga need kaks eraldatud. CMM kindlalt eraldab mõlemad vaated, kuid protsessi osa katab siiski oluliselt põhjalikumalt kui tehnoloogia oma.

CMM'i ja ISO 9000 võrdlemisel tulevad välja ka järgnevad suhted:

- CMM'i skaalal 3. tasemele kuuluvale organisatsioonile ei tohiks olla mingi probleem ISO 9000 sertifikaadi omandamine.
- Kuna ISO sertifikaate jagavad inimesed pole tavaliselt tarkvara osas just kuigi 'kodus', siis ei saa väita, et ISO sertifikaati omav firma CMM'i skaalal 3. tasemeni küündiks.

(Marc De Smet, "Software Process Improvement and the Capability Maturity Model", lk. 9)

CMM vs. 15504 (SPICE)

15504 on üheksakümnendate esimesel poolel tarkvaraprotsessi hindamiseks välja töötatud rahvusvaheline standard. Selle koostamisel on inspiratsiooni saadud mitmetelt teistelt mudelitelt; nende hulgas on CMM, BootStrap, Trillium, SAM jt. 15504 arendamise juures on olnud võtmefiguurideks ka mõned samad inimesed, kes tegelesid CMM'i välja töötamisega. SPICE'i kohta leiab lisainformatsiooni järgmistelt aadressidelt: <http://www.sei.cmu.edu/iso-15504/resources/v1baseline.html>, <http://www.scpm.salford.ac.uk/spice/index.html>.

SEI CMM ja 15504 omavad erinevusi nii protsessi arenduse, kui ka abstraktsiooni tasemete osas. Abstraktsiooni tasemete erinevus on seotud nende kahe mudeli eesmärgiga. 15504 kui rahvusvaheline standard on siiski piisavalt normatiivne, et hõlmata erinevaid protsessi parendamise viise, mis sisaldavad ka CMM'i vaateid protsessi arendamise osas. CMM jagab kasutajatele informatiivset teadet mitte just kõige detailsemalt, kuid üritades siiski tekitada kasutajate jaoks mingi visiooni sellest, millised peaksid välja nägema kõrgema tasemega protsessi tegevused.

Struktuursed erinevused mõnede CMM'ide (n. SE-CMM) ja 15504 vahel on suhteliselt otsesed, kuid samas teiste CMM'ide (n. SW-CMM) ja 15504 vahel on erinevused rohkem kaudsed. SW-CMM'i areng peaks nende kahe mudeli vahelisi seoseid veelgi nähtavamaks tegema, mis võib viia isegi selleni, et 15504 auditeid on võimalik teha ka kasutades standardseid SEI meetodikaid – muidugi sõltub see ka sellest, mis suunas areneb edasi 15504 järgmine versioon. (Suzanne M. Garcia, "Evolving Maturity Paradigms: Capability Maturity Models & ISO/IEC 15504 (PDTR)", lk. 10)

Põhilised sarnasused CMM'i ja 15504 vahel on järgmised:

- Fookus pideval protsessi arendamisel.
- Mõlemad kasutavad tasemeid protsessi küpsusastme esitamiseks. SPICE'il on 6 taset [0..5].
- Tegu on avalike ja tasuta saadaval olevate mudelitega.
- Arusaam, et kompetentsed hindajad on elulise tähtsusega.

Põhilised erinevused:

- 15504 puhul on põhimõtteliselt tegu raamsüsteemiga (*framework*). Näide: raamsüsteem – mudel => mingi asja spetsifikatsioon – mingi asi.
- Arhitektuurilised erinevused. 15504 ja CMM jagavad protsessi domeeni osadeks erinevate kriteeriumite järgi. 15504 vaatab neid osasid kui eraldi protsesse. CMM aga vaatab neid kui tihedalt omavahel seotud osi.
- Protsessi ulatus.
- Orienteeritus. CMM on loodud suurte organisatsioonide ja projektide jaoks, 15504 seevastu on aga universaalne, ehk seda saab edukalt rakendada ka väikeprojektide, -firmade korral.

CMM vs. XP

Kuigi esmapilgul võivad need kaks tunduda vägagi erinevad, on nad siiski kohati ka väga sarnased. Pelgalt nime järgi seda siiski ei julgeks väita: Capability Maturity Model ja eXtreme Programming. Tunduvad tõesti nagu kaks täiesti erinevat asja, kuid vähemalt CMM'i esimese kolme taseme juures sarnasusi jagub, nii et organisatsiooni küpsustaset on võimalik tõsta ka kasutades XP'd, et CMM'i skaalal tõusta. XP kohta leiab lisainformatsiooni aadressilt <http://www.extremeprogramming.org>.

XP on nagu süsteemne vaade programmeerimisele, SW-CMM aga süsteemne vaade organisatsiooni protsessi parendamiseks. Organisatsioonid, mis tahavad oma võimekust tõsta, võiksid valida mõlema variandi hulgast enda jaoks sobivaid ideid ja neid arukalt rakendada.

Kas organisatsioonid peaksid kasutama XP'd suurte kriitiliste reaalajasüsteemide arendamisel? Kindlasti mitte, kuna XP ei aseta erilist rõhku disaini dokumentatsioonile ja arhitektuurile. Kuid samas XP üks tõhusaim külg on see, et seda on lihtne kohandada

vastavalt keskkonnale jne. (Mark C. Paulk, “Extreme Programming from a CMM Perspective”, 2001, lk. 4) Paindlikkus ei ole jälle näitaja, mis iseloomustab CMM’i.

Järgnevalt toon välja tabeli, kus on tasemeti kirjas CMM’i KPA’ d ning kas XP neid toetab või mitte.

Tase	KPA	Toetus XP osas
2	Nõuete haldus	++
2	Projekti planeerimine	++
2	Tarkvaraprojekti jälgimine ja ülevaadete tegemine	++
2	Alltöövõtude haldus	--
2	Tarkvara kvaliteedikontroll	+
2	Tarkvara konfiguratsioonihaldus	+
3	Organisatsiooni fookus protsessil kui tervikul	+
3	Organisatsioonipoolne protsessi defineerimine	+
3	Koolitusprogrammid	--
3	Terviklik tarkvaraarenduse juhtimine	--
3	Tarkvaratoote arendus	++
3	Gruppidevaheline koordineerimine	++
3	Vastastikused läbivaatused	++
4	Kvantitatiivne protsessi juhtimine	--
4	Kvaliteedijuhtimine	--
5	Defektide ärahoidmine	+
5	Tehnoloogia muutuste juhtimine	--
5	Protsessi muutuste juhtimine	--
+ osaliselt toetatud ++ täielikult toetatud -- XP-poolne toetus puudub		

Tabel 2. XP-poolne toetus CMM’i KPA’dele eeldusel, et eksisteerib seda soosiv keskkond. (Mark C. Paulk, “Extreme Programming from a CMM Perspective”, 2001, lk. 6)

Nagu eelnevast tabelist näha, adresseerib XP vaid ühte KPA’ d neljanda ja viienda taseme peale kokku. Mitmed KPA’ d, mida XP kas täielikult või siis osaliselt ignoreerib, leiavad tõenäoliselt oma osa siiski tegelikes projektides. XP vajab organisatsiooni ja infrastruktuuri toetust isegi kui see esmapilgul seda välja ei näita.

Üks suur erinevus, mis veel nende kahe mudeli vahel välja tuleb on nende orienteeritus. XP on mõeldud väiksemate projektide ja meeskondade jaoks (reeglina kuni 20 inimest), kus arendustöö on jagatud väga lühikesteks tsükliteks; CMM on seevastu just suurte

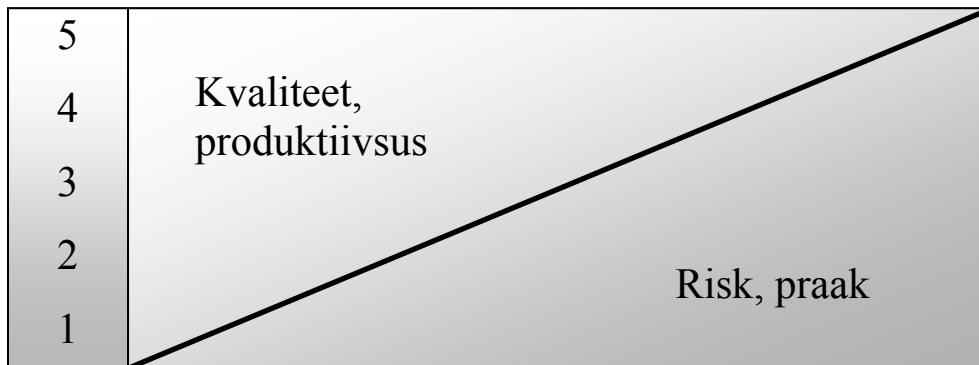
organisatsioonide ja projektide jaoks, kus meeskonnaliikmete arv võib olla mitmeid kordi suurem, kuna CMM'i puhul tulevad mängu ka alltöövõtjad, mida XP omakorda ei tunnista.

Ka XP puhul on tasemed välja toodud, kuid nende ja CMM'i tasemete vahel paralleele tõmmata siiski ei saa. XP loojate sõnul on tasemed järgmised:

1. Esimene tase on see, kui XP'd järgitakse 1:1.
2. Teine tase on siis, kui XP on modifitseeritud vastavalt organisatsiooni, projekti või tiimi vajadustele, omadustele.
3. Kolmas tase on see, kui enam ei hoolita sellest, kas nö. 'tehakse' XP'd või mitte.

III SPI rakendamise tulemused

Mida tähendab organisatsiooni jaoks küps protsess? Mis muutub, kui organisatsioon hakkab tegelema protsessi arendamisega? Nendele küsimustele annab põgusa vastuse järgmine joonis.



Joonis 3. Kvaliteedi ja produktiivsuse kasv ning riski ja praagi osakaalu vähenemine liikudes CMM'i madalamatelt tasemetelt kõrgemate suunas.

Kuigi eelnev joonis kajastab CMM'i järgimisel tekkivat kvaliteedi ja produktiivsuse kasvutrendi, on samad näitajad omased ka teiste SPI mudelite kasutamisele.

Küpse protsessi hind ei ole midagi kolossaalset, küps protsess ei maksa rohkem raha. Otse vastupidi, pikas perspektiivis see säästab raha. Kypse protsessi juurutamine juba käimasoleva projekti juures on äärmiselt keeruline ja ebaotstarbekas. Samas on protsessi juurutamine pikaajaliste äriliste püüdlustega organisatsioonis ja pikaajalistes projektides, mis on alles planeerimise staadiumis, kergem.

Küpset protsessi juurutades suureneb ka produktiivsus. Tarkvaratootjate puhul on võimalik seda mõõta mitmeti, näiteks kirjutatud koodiridade arv (LOC) mingi ajaühiku kohta või varajases staadiumis avastatud defektide arv. Varajases staadiumis avastatud defektide parandamine on palju odavam, kui nende parandamine näiteks testimisfaasis või üldse peale toote valmimist ja kliendile üleandmist. Erinevate uurimuste ja hinnangute järgi küündib hinnavahe isegi mitme tuhande dollarini. Mida varasemas staadiumis defekt avastatakse, seda odavam tuleb ka selle parandamine.

Toote valmimise aeg väheneb. Kui SPI'd rakendada organisatsioonis, siis tavaliselt esimese aasta jooksul võib toote valmimise aeg olla isegi pikem, kuna protsess pole veel piisavalt omaks võetud või siis pole protsess piisavalt optimeeritud. Peale kahte-kolme aastat see aeg aga väheneb olulisel määral võrreldes ajaga, mis kulus enne SPI rakendamist.

Sarnaste projektide puhul see aeg väheneb aastatega kuni teatud maani, mille järel muutub see enam-vähem konstantseks ehk on saavutatud optimaalne tase antud tingimuste juures.

Kvaliteet suureneb. Kvaliteet viitab tarkvara seisundile, kui see tellijale üle anti või turule toodi. Üks näitaja, mille abil kvaliteeti hinnata saab on defektide avastamine peale toote üleandmist/väljastamist.

Rentaablus. See tähendab põhimõtteliselt seda, mitu krooni teenitakse tagasi iga investeeritud krooni pealt. Sageli öeldakse selle kohta ka ROI (return on investment). Ühe uurimuse põhjal oli see näitaja 5, mis on väga hea tulemus, kuna on vähe investeerimisvõimalusi, mis pakuvad samasugust tulusust. Rentaablus on aga üks näitaja, mida on väga raske ennustada enne SPI rakendamist.

Järgnev tabel toob välja ühe uurimuse tulemused, mis viidi läbi SEI poolt 13 firma seas, mis olid rakendanud erinevaid SPI meetodikaid. Antud firmad ei kuulu ainult CMM'i 4. ja 5. tasemele, vaid nende hulgas on ka madalamate tasemete esindajaid. Peale protsessi parendamise alustamist on olulisel määral paranenud järgmised näitajad: produktiivsus, defektide avastamine varases staadiumis, toote valmimise aeg ja kvaliteet.

Kategooria	Hinnavahe	Mediaan
SPI rakendamise maksumus aastas	\$49000 - \$1202000	\$245000
SPI'ga tegeldud aastate arv	1-9	3,5
SPI maksumus ühe inimese kohta	\$490 - \$2004	\$1375
Produktiivsuse kasv aastas	9% - 67%	35%
Varases staadiumis avastatud defektide arv	6% - 25%	22%
Toote valmimise aja vähenemine aastas	15% - 23%	19%
Peale toote valmimist ilmnenu defektide arv	10% - 94%	39%
Rentaablus	4,0 – 8,8	5,0

Tabel 3. Kokkuvõtte SPI rakendamise tulemustest. (James Herbsleb, Anita Carleton, James Rozum, Jane Siegel, David Zubrow, "Benefits of CMM-Based Software Process Improvement: Initial Results", 1994, lk. 26)

Toon veel rea positiivseid näitajaid, mida on SPI rakendamine organisatsioonis endaga kaasa toonud.

- Kodeerimisele ja testimisele kuluv aeg lüheneb.
- Toetudes olemasolevale faktide ja muu info kogumile on võimalik projektide õnnestumist ette ennustada.
- Meeskonna vaim, uhkus ja moraal on tõusnud.

- Uut tehnoloogiat on lihtsam kasutusele võtta.
- Töötingimused on paranenud: vähem ületunde, stressi ja närvesöövaid probleeme.
- Kriisiolukordi tuleb väga harva ette.

Kokkuvõte tööst

Tarkvarafirmad üle maailma on hakanud lõpuks aru saama, et asi ei saa samamoodi enam edasi kesta. Statistikud näitavad meile numbreid, mida on lausa valus vaadata. 1998. aastal USA's läbi viidud uurimuse tulemusena selgus, et 175000 infotehnoloogiaprojektist, mille eelarve oli kokku 250 miljardit dollarit, täideti edukalt tähtjaks ja eelarve piirides vaid 26%; paar aastat varem oli see näitaja umbes 10%. Et sellele lõppu teha hakati välja töötama mudeleid, mis aitaksid arendada organisatsioonide tarkvaraprotsessi.

Üks tuntumaid ja levinumaid on SEI poolt välja töötatud suutvusküpsuse mudel ehk CMM. Selle rakendamine on end enamikel juhtudel ka õigustanud, kuna asjaga on tegelenud kompetentsed ja pühendunud inimesed. SPI rakendamine on pikk protsess, mis nõuab pühendumist ka kõige kõrgemal organisatsiooni tasemel – managementi täielik toetus on ülioluline. Nagu juba öeldud, protsessi muutmine võtab aega ja selle tegelike tulemuste nägemine ei pruugigi kohe 'silma karata'. Protsessi muutus mõjutab inimesi ja organisatsiooni rohkem sügavuti, kui tehnoloogia või vahendite muutmine. Seda tuleb plaanida ja juhtida hoolikalt. Tuleb tuvastada muudatuse väljavaated ja oodatav kasu, teha see huvitatud osalistele selgeks, tõsta nende teadlikkuse taset ja siis järkjärgult liikuda seniselt tööpraktikalt uuele. Kui lihtsalt üks hetk otsustada, et alates tänasest hakkame asju tegema nii, siis head nahka sellest ei tule.

Need küpsusmudelid võivad ka tekitada sellise illusiooni, et kui nüüd seda rakendada, siis on kõik varsti OK. Samuti järgitakse tihti seda, et juhised saaksid 100% 1:1 täidetud. See ongi üks tüüpiline viga, mida tehakse. Tegelikuses on aga asi nii, et neid ei ole üldsegi otstarbekas 1:1 kasutusele võtta, vaid neid tuleb kohandada vastavalt organisatsiooni eripäradele. Kui õige lahendus kohe silma ei paista, siis tuleb katsetada – vigadest ju õpitaksegi! Kui vaid õpitaks ka teiste vigadest...

Oleks vale öelda, et mõni mudel on halb ja mõni hea ning et mõni ei kõlba üldse kusagile. Kõik mudelid on head eeldusel, et neid rakendatakse õiges keskkonnas, õigete inimeste poolt ja kindla eesmärgiga. Näiteks XP rakendamine suures ettevõttes suure projekti korral on praktiliselt võimatu, seevastu CMM'i jaoks sobib see kui rusikas silmaauku. Kuigi CMM'i on võimalik kasutada ka väikefirmas, võib see siiski kujuneda ebaotstarbekaks. Veelgi parema efekti saavutamiseks on otstarbekas kasutada mitut mudelit paralleelselt, kuna mõlemad täiendavad teineteist. Üks CMM'i autoritest, M. C. Paulk on isegi öelnud, et väikeste firmade puhul tuleks CMM'i asemel kasutada mingit teist mudelit, näiteks TSP'd koos PSP'ga.

Inimeste harjumusi ja organisatsiooni kultuuri on väga raske muuta. Kui aga inimesele on võimalik selgeks teha, et uus viis on parem, siis ootab meid ees helge tulevik.

Lisad

Lisa 1. Akronüümid

CMM – Capability Maturity Model

SW-CMM – Software Capability Maturity Model / Capability Maturity Model for Software

ISO – International Organization for Standardization

KPA – Key Process Area

LOC – Lines of code

PSP – Personal Software Process

ROI – Return on investment

SCE – Software Capability Evaluation

SEI – Software Engineering Institute

SPI – Software Process Improvement

SPICE – Software Process Improvement and Capability dEtermination

TSP – Team Software Process

XP – eXtreme Programming

Lisa 2. Sõnaseletused

Defekt – defect - toote või pooltoote toimimise kõrvalekalle või puudujääk. Defektid on näiteks elutsükli varastes faasides leitud tegematajätmised ja puudused ning testimiseks või kasutamiseks valmis oleva tarkvara vigade sümptomid.

Faas – phase - kahe peamise tähtpunkti vaheline aeg, mille jooksul saavutatakse hulk hästidefineeritud eesmärke.

Protsess – process - Hulk osaliselt järjestatud samme mingi eesmärgi saavutamiseks. Tarkvaratehnikas on eesmärgiks valmistada uus tarkvaratoode või muuta olemasolev paremaks.

Metodoloogia – Methodology – Kõrge distsipliiniga kindlate tegevuste ja nõuete kogum, mis ütlevad mida teha ja kuidas seda teha.

Mudel – Model – Kindla struktuuri ja eesmärgiga süsteemi detailne kirjeldus.

Kasutatud kirjandus

1. Walker Royce, "Software Project Management: a Unified Framework," The Addison-Wesley object technology series. 1998, Addison Wesley Longman, Inc.
2. James Herbsleb, Anita Carleton, James Rozum, Jane Siegel, David Zubrow, "Benefits of CMM-Based Software Process Improvement: Initial Results", <http://www.sei.cmu.edu/publications/documents/94.reports/94.tr.013.html>, 1994, CMU/SEI.
3. Mark C. Paulk, "Practices of High Maturity Organizations in 2001", 2001, CMU/SEI.
4. Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles W. Weber, "Capability Maturity Model for Software, version 1.1", <http://www.sei.cmu.edu/publications/documents/93.reports/93.tr.024.html>, 1993, CMU/SEI.
5. Mark C. Paulk, "A Comparison of ISO 9001 and the Capability Maturity Model for Software", <http://www.sei.cmu.edu/pub/documents/94.reports/pdf/tr12.94.pdf>, 1994, CMU/SEI.
6. Suzanne M. Garcia, "Evolving Maturity Paradigms: Capability Maturity Models & ISO/IEC 15504 (PDTR)", CMU/SEI.
7. Mark C. Paulk, "Using the Software CMM With Good Judgement", 1999, CMU/SEI.
8. Mark C. Paulk, "Extreme Programming from a CMM Perspective", <http://www.sei.cmu.edu/cmm/papers/xp-cmm-paper.pdf>, 2001, CMU/SEI.
9. Mark C. Paulk, "Using the Software CMM in Small Organizations", <http://www.sei.cmu.edu/cmm/papers/cmm-small.pdf>, CMU/SEI.
10. Kai A. Simon, "The Capability Maturity Model as an Approach to Improvement", IMG.
11. Asko Seeba, "Unifitseeritud tarkvaraarendamise protsess ja selle rakendamise juhtumianalüüs", 2001, TÜ.
12. Peeter Normak, "Projektijuhtimine: loengumaterjal", 2001, TPÜ.
13. Marc De Smet, "Software Process Improvement and the Capability Maturity Model", Management Information Systems in Zoersel (Belgia).
14. "SPICE: Software Process Improvement – Part 1: Concepts and introductory guide, version 1.00", <http://www.sei.cmu.edu/iso-15504/resources/v1baseline.html>.