

# CMM - metode za osiguranje kvalitete softvera

---

Šugić, Josip

Master's thesis / Diplomski rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:862454>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-29**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli  
Fakultet ekonomije i turizma  
«Dr. Mijo Mirković»

**JOSIP ŠUGIĆ**

**CMM – METODA ZA OSIGURANJE  
KVALITETE SOFTVERA**

Diplomski rad

Pula, 2015.

Sveučilište Jurja Dobrile u Puli  
Fakultet ekonomije i turizma  
«Dr. Mijo Mirković»

**JOSIP ŠUGIĆ**

**CMM – METODA ZA OSIGURANJE  
KVALITETE SOFTVERA**

Diplomski rad

**JMBAG: 0303004718, redoviti student**

**Studijski smjer: Poslovna informatika**

**Predmet: Management kvalitete**

**Mentor: Izv.prof.dr.sc. Giorgio Sinković**

Pula, rujan 2015.

## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani \_\_\_\_\_, kandidat za magistra \_\_\_\_\_ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student:

U Puli, 09. 09. 2015.

\_\_\_\_\_

# Sadržaj

UVOD .....	2
1. RAZVOJ SOFTVERA .....	4
1.1. Model životnog ciklusa softvera .....	5
1.2. Metode i alati za razvoj softvera .....	9
2. KVALITETA SOFTVERA.....	11
2.1. Pojam kvalitete .....	11
2.1.1. Kvaliteta kao tržišni čimbenik .....	12
2.1.2. Kvaliteta sa stajališta kupca, proizvođača, tržišta i društva .....	13
2.2. ISO 25000; SquaRE (Software product Quality Requirements and Evaluation).....	14
2.3. Upravljanje kvalitetom .....	18
2.2.1. Planiranje kvalitete .....	18
2.2.2. Kontrola kvalitete .....	19
2.2.3. Osiguranje kvalitete.....	20
3. CMM (Capability Maturity Model) – METODA ZA OSIGURANJE KVALITETE SOFTVERA .....	21
3.1. CMM pojam .....	21
3.2. Povijest CMM-a .....	22
3.3. Razine zrelosti CMM-a .....	23
3.3.1. Inicijalna razina .....	28
3.3.2. Ponavljajuća razina.....	28
3.3.3. Definirana razina .....	30
3.3.4. Upravljana razina .....	33
3.3.5. Optimizirana razina.....	35
3.3. Uobičajene značajke i ključni postupci/procedure CMM strukture .....	38
3.4. Korištenje CMM-a .....	41
3.5. CMMI (Capability Maturity Model Integration).....	42
3.5.1. CMMI okvir (eng. Framework) .....	45
3.5.2. CMMI za Razvoj (CMMI-DEV).....	46
3.5.3. CMMI za Usluge (CMMI-SVC).....	47
3.5.4. CMMI za Nabavu (CMMI-ACQ).....	47
3.6. Ostali CMM modeli.....	49
3.6.1. P-CMM – People Capability Maturity Model .....	49
3.6.2. SSE-CMM – Model sigurnosti (eng.System Security Engineering Capability Maturity Model) .....	49
3.7. CMM certificiranje.....	50

3.7.1. <i>Ustanove za certifikaciju</i> .....	53
4.    OSTALI MODELI ZA OSIGURANJE KVALITETE SOFTVERA.....	55
4.1.    ISO 9001.....	55
4.1.1. <i>TickIT</i> .....	56
4.2.    ISO/IEC 90003 .....	57
4.3.    Bootstrap metoda.....	60
ZAKLJUČAK .....	62
LITERATURA.....	64

## UVOD

Razvoj informacijskih tehnologija danas obuhvaća svaki kutak života modernog čovjeka u 21. stoljeću. Čovjek kao društveno biće proširio je svoju sposobnost komuniciranja na razne informacijske tehnologije. Informacijske tehnologije tako danas nalazimo u školskim ustanovama, poslovnim poduzećima, svjetskim vladama čime direktno i indirektno utječu na živote ljudi na svijetu. Danas većina ljudi barem jednom na dan koristi jedan tip informacijske tehnologije, bilo to računalo, mobitel ili tablet. Korisnici tih uređaja stupaju u kontakt s različitim softverima koji su napravljeni za milijun različitih funkcija a sve u svrhu lakšeg korištenja tih elektronskih uređaja.

Od samih početaka razvoja računala, hardver i softver su uz čovjekovu prisutnost obuhvaćali nekakav začarani krug potrebnih dijelova za pravilno funkcioniranje računala. Razvojem novih tipova računala, hardverski troškovi su tijekom godina postepeno opadali dok su troškovi softvera iz godine u godinu polagano rasli te danas obuhvaćaju veliki dio troškova u IT sektoru. Tijekom razvoja softvera postepeno se razvijala i potreba za softverskom kvalitetom koja je preuzela glavnu ulogu u razvoju novih informacijskih tehnologija. Kako je kvaliteta softvera postala važan dio razvoja i održavanja softvera razvili su se i modeli za osiguranje njegove kvalitete od kojih su danas najpoznatiji i najkorišteniji CMM - Capability Maturity Model i ISO 9001.

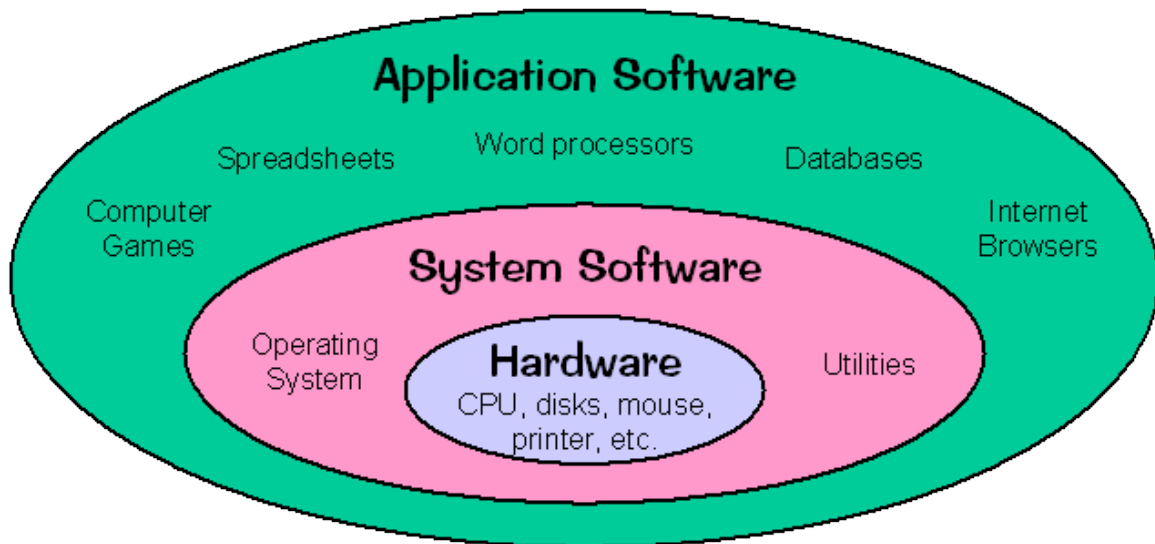
U ovom radu će biti govora o razvoju prvenstveno CMM metode za osiguranje kvalitete softvera, CMM razinama zrelosti i njezinim vrstama kao i drugim metodama za osiguranje kvalitete softvera poput norme ISO 9001. Osim norme ISO 9001 ovaj rad će obuhvatiti i usko povezane norme koje se bave razvojem softvera i osiguranjem njegove kvalitete. Te norme su ISO 9001: 2008 koji obuhvaća točke za konstanto osiguranje kvalitete softvera od strane poduzeća za korisnike softvera, normu ISO 25000 koja obuhvaća zahtjeve potrebne za kvalitetu softvera i njegovu evaluaciju te ISO 90003 koji služi kao vodič za primjenu ISO 9001:2008 standarda na računalni softver. Uz opis metode CMM za osiguranje kvalitete softvera bit će govora i o njezinim razinama tj. stupnjevima razvoja softvera. Bit će govora i o CMM certificiranju, ustanovama za certificiranje te kako one djeluju. Također ovaj rad će obuhvatiti i razvoj CMMI - Capability Maturity Model Integration modela kao novu verziju CMM modela nastalu krajem 1990-tih godina, njegovu efektivnost te njezine razne vrste CMMI-DEV Capability Maturity Model For Development koji obuhvaća razvoj softvera i

softverskih projekata, CMMI-SVC koji se bavi uspostavljanjem, upravljanjem i pružanjem usluga, CMMI-ACQ koji se bavi objedinjavanjem proizvoda i usluga, PCMM – People Capability Maturity Model koji obuhvaća prilagođavanje zaposlenika poduzeća razvoju novog softvera i izgradnji starog, SSE-CMM model koji se bavi sigurnošću softvera i IT-a. Na kraju ovaj rad će obuhvatiti i Bootstrap metodu koja analizira, redizajnira i unaprijeđuje poslovne procese razvoja softvera.



# 1. RAZVOJ SOFTVERA

Prije nego što se krene u tematiku CMM metode potrebno je objasniti ono na što se CMM nadovezuje a to je softver i njegova kvaliteta. Za softver (*eng. Software*) se može reći da je neopipljivi dio računala tj. računalni program koji služi prvenstveno za upravljanje hardverom (*eng. Hardware*) kojeg čine mehanički i elektronički dijelovi samog računala. Pod softver se nalaze aplikacijski programi, operacijski sustav te programi i podaci koji se nalaze na samom računalu.



Slika 1. Koncept softvera (Izvor : <http://mcqsets.com/s/fundamentals/computer-software-concept/>)

Slika 1. Prikazuje koncept softvera u računalu. Na slici se može vidjeti da je softver podijeljen na dva dijela, Aplikacijski i Sistemski softver koji se nadograđuju na hardver računala. Pod Aplikacijski softver pripadaju na primjer internetski pretraživači (*eng. Internet browsers*), baze podataka (*eng. Databases*), aplikacije za obradu teksta (*eng. Word processors*), proračunske tablice (*eng. Spreadsheets*) i računalne igre (*eng. Computer games*). Aplikacijski softver se koristi za ostvarivanje određenih računalnih zadataka koji nisu odgovorni za pokretanje računala i njegovog sistema te on može sadržavati jedan računalni program npr. za slušanje glazbe ili skupinu programa (softverski paket) koji imaju zajednički zadatak. Za razliku od Aplikacijskog softvera Sistemski softver je odgovoran za kontrolu, integraciju i

upravljanje hardvera unutar računalnog sistema. Pod Sistemskim softverom se nalazi operacijski sistem (*eng. Operating system*) poput Microsoft Windows-a. Windows-i su danas jedan on najpoznatijih i najrasprostranjenijih operacijskih sistema na svijetu. Uz operacijski sistem također se nalaze i brojni dodatni uslužni programi (*eng. Utilities*) poput programa za formatiranje tvrdog diska (*eng. Hard disk*), programa za upravljanje datotekama, uređivači teksta, mrežni softver itd.

### **1.1. Model životnog ciklusa softvera**

Kao i svaki proizvod kojeg koriste ljudi bio on opipljiv ili neopipljiv poput softvera, tijekom svoje izgradnje prolazi kroz svoj životni ciklus. Životni ciklus predstavlja jedan vremenski tijek izgradnje, implementacije, korištenja i gašenja softverskog proizvoda. U današnje vrijeme na računalno tržište svake godine iziđe stotine različitih vrsta softvera prilagođenih za razne namjene. U sjeni tih nekoliko stotina nalaze se i oni softverski proizvodi koji nisu vidjeli svjetlo dana tj. njihov životni vijek je završio prije same implementacije proizvoda. To se dogodilo možda tijekom izrade i testiranja prototipa ili je bio odbačen kao ideja na samom početku. Kroz ovaj tekst se može uočiti da životni ciklus proizvoda čini nekoliko usko povezanih aktivnosti koje su:

1. Inicijalizacija sustava je aktivnost u kojoj se navodi podrijetlo softvera.
2. Analiza i specificiranje zahtjeva je aktivnost u kojoj se identificiraju problemi koje je potrebno riješiti novim softverom.
3. Specifikacija funkcija je aktivnost u kojoj se identificiraju i formaliziraju podaktivnosti definiranja predmeta obrade, identificiranje atributa i veza objekata i operacija.
4. Strukturiranje i izbor dijelova su aktivnosti kojima se na osnovi identificiranih zahtjeva i specifikacije funkcija strukturira softver na takve dijelove kojima se može upravljati, a koji predstavljaju logičke cjeline.
5. Specifikacija strukture je aktivnost u kojoj se definiraju međusobne veze između dijelova strukture i sučelje između modula sustava.
6. Specifikacija detaljnih komponenti dizajna je aktivnost u kojoj se definiraju procedure putem kojih se izvori podataka svakog pojedinog modula transformiraju iz potrebnih ulaza u zahtijevane izlaze.

7. Implementacija komponenti i otklanjanje nedostataka je aktivnost u kojoj se kodiraju dizajnirane procedure i procesi i pretvaraju u izvorni kod.
8. Integracija i testiranje softvera je aktivnost koja potvrđuje i održava cjelokupnu integralnost komponenti softvera putem verifikacije konzistentnosti i kompletnosti uvedenih modula.
9. Provjera dokumentacije i uvođenje softvera je aktivnost koja obuhvaća izradu systemske dokumentacije i uputa za korisnika.
10. Obuka i upotreba je aktivnost koja osigurava korisnicima softvera instrukcije i upute za razumijevanje mogućnosti i ograničenja u cilju uspješne upotrebe sustava.
11. Održavanje softvera je aktivnost koja podržava operacije sustava u ciljnom okruženju na način da osigura potrebna unapređenja, proširenja, popravke, zamjene i dr.
12. Gašenje softvera (povlačenje iz primjene) je posljednja aktivnost u životnom ciklusu.<sup>1</sup>

Tijekom godina razvoja i evolucije softvera razvili su se razni procesi za njegov razvoj koji se nazivaju modeli životnog ciklusa softvera. Najpoznatiji model za prikazivanje životnog ciklusa softvera je vodopadni model. Vodopadni model prikazuje životni ciklus softvera kroz sedam faza koje su:

1. Zahtjevi i specifikacije. Ova metoda bavi se definiranjem zahtjeva, specifikacija zahtjeva koje treba riješiti određenim programskim proizvodom.
2. Projektiranje (dizajn) programskog proizvoda. U ovoj fazi se specificira cjelokupna konfiguracija sustava, programski jezik, glavni moduli i njihove veze, strukture podataka te plan testiranja.
3. Detaljni dizajn. Detaljnije se specificiraju moduli, njihova neophodna međusobna komunikacija, algoritmi, strukture podataka i interne kontrolne strukture.
4. Programiranje/kodiranje. U ovoj fazi se izgrađuju moduli. Istovremeno se testiraju pojedine jedinice modula kao i testiranje pojedinih podsistema.
5. Integriranje sustava. U ovoj fazi se testiraju pojedini moduli u međusobnom interaktivnom radu, te se testira se cjelokupni sustav.
6. Instaliranje/prihvatanje sistema. Programski proizvod isporučuje se korisniku na testiranje. Također se isporučuje korisnička dokumentacija, izvodi se obučavanje

---

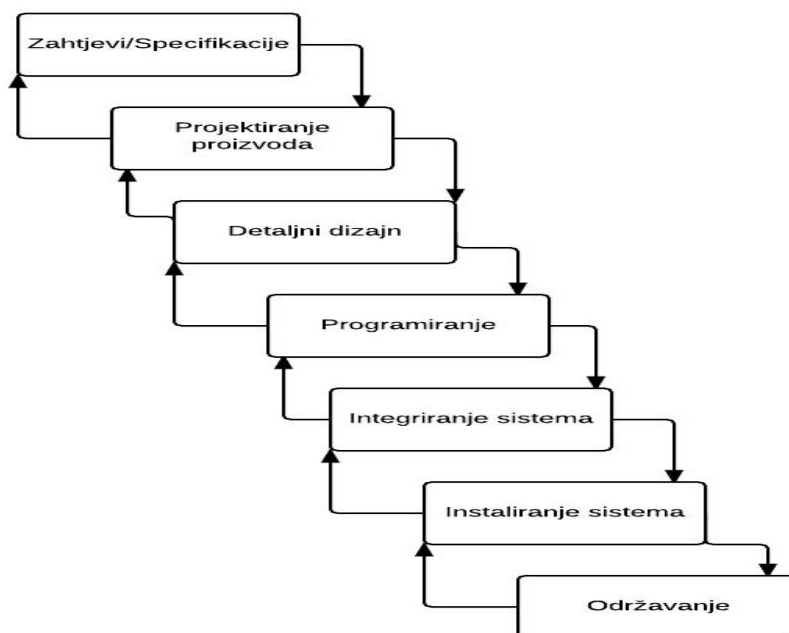
<sup>1</sup> [hrcak.srce.hr/file/151922](http://hrcak.srce.hr/file/151922) (preuzeto 20.05.2015.)

korisnika. Među ostalim u ovoj fazi se ispravljaju eventualne greške prije prihvaćanja programskog proizvoda.

7. Održavanje. Zadnja faza koja je usko povezana s osiguranjem kvalitete softvera i CMM-om. Ovo je dugotrajna faza ispravljanja naknadno uočenih grešaka, izmjena koda, priručnika itd.<sup>2</sup>

Ovih sedam faza se grupira u četiri kategorije. Prva kategorija je dizajn u kojoj se nalaze prve tri faze životnog ciklusa softvera tj. zahtjevi i specifikacije, projektiranje i detaljni dizajn. Druga kategorija predstavlja kodiranje softvera gdje se softver razvija, dok je treća kategorija testiranje softvera gdje se nalaze integriranje sistema i instaliranje/prihvaćanje sistema. Zadnja kategorija je održavanje pod kojoj se nalazi sedma istoimena faza. Kod održavanja se proizvod koristi, nadograđuje tj. razvija dok sve korisnik softvera ima koristi od njega, dok ne zastari ili bude zamijenjen novijom verzijom.

Zanimljivo je navesti da programiranje i kodiranje softvera zauzima najveći postotak utrošenog vremena za razvoj softvera što je 40%, dok dizajniranje zauzima 25%, integriranje sistema 20% a projektiranje 15% radnog vremena.<sup>3</sup>



Slika 2. Vodopadni model životnog ciklusa softvera

(Izvor: hrcak.srce.hr/file/151922; datum pristupa 20.05.2015.)

<sup>2</sup> hrcak.srce.hr/file/118917 (preuzeto 20.05.2015)

<sup>3</sup> hrcak.srce.hr/file/151922 (preuzeto 20.05.2015)

Slika 3. Prikazuje sve faze vodopadnog modela životnog ciklusa softvera. Na slici se mogu primijetiti strelice koje definiraju da sljedeća faza životnog ciklusa može početi tek kad je prethodna završila. Može se vratiti na prethodnu fazu ali jako teško jer se time ponekad gubi napredak koji je postignut na sljedećoj fazi. Vrijedi napomenuti da se vodopadni model primjenjuje samo u projektima kod kojih su programeri detaljno definirali kako stići od prve do zadnje faze. Ako su jednoj od faza vodopadnog modela dogodi bilo kakva pogreška koja je trebala biti ispravljena u prethodnoj fazi ili se neki zahtjevi promjene dolazi do problema koji je skupa prilagodba novim zahtjevima i popravcima softvera.

Postoje dvije vrste modela, sekvencijalni koji se naziva još i vodopadni te iterativni npr. evolucijski. Evolucijski model programeri koriste kada su nakon razgovora s korisnikom tog softvera došli do zaključka da zahtjevi nisu u potpunosti definirani. Može se tako dogoditi da tijekom razvijanja softvera korisnik odluči promijeniti svoje zahtjeve ili dodati nove. U tom slučaju je najbolje koristiti evolucijski model. On se najčešće koristi kod malih projekata. Najvažnije značajke evolucijskog modela su dakle da od ideje koju programeri dobiju od korisnika razvijaju prvu verziju softvera, nakon koje slijedi proces razvoja koji u sebi uključuje specificiranje, razvijanje i validiranje softvera. Nakon što su programeri napravili softver koji odgovara svim zahtjevima korisnika oni ga šalju korisniku na testiranje. Nakon toga korisnik ga pokreće i pregledava te ako ima bilo kakve primjedbe ili želi nekakve izmjene unutar softvera, korisnik daje primjedbe programerima koji dodatno prilagođavaju softver potrebama korisnika. Za razliku od definirane kontrole procesa kojom se služi vodopadni model, gdje je predvidljivost ishoda visoka, evolucijski tj. iterativni model se služi empirijskom kontrolom procesa prilagođenom za neizvjesne okolnosti dok se vodopadni služi definiranom kontrolom procesa. Vodopadni model je tradicionalan te se oslanja na detaljno dokumentiranje izlaza svih faza kao ulaza u sljedeću fazu dok se evolucijski model oslanja na fleksibilnost te je prilagodljiv na promjene u razvoju softverskog proizvoda.

## 1.2. Metode i alati za razvoj softvera

Tijekom godina razvoj softvera je postao skup process. Danas je kvalitetan softver ponekad i skuplji od hardverskih dijelova računala. Kako su rastom troškova softvera porasle i cijene softvera uvode se nove tehnike kao što je softversko inženjerstvo s namjerom da se cijena koštanja softvera smanji te poveća kvaliteta. Razvoj softvera je proces koji obuhvaća sve korake od samog formiranja ideje do realizacije te ideje u obliku finalnog proizvoda. Tako se može reći da je razvoj softvera složen posao koji se sastoji od velikog broja koraka, počevši od početnog zahtjeva korisnika, preko analize i izgradnje, pa do implementacije i korištenja. A da bi se softver razvio potrebno je koristiti razne metode, alate, tehnologije.

Metode pokrivaju širok spektar zadataka među kojima su: planiranje i procjenjivanje projekata, analiza sistemskih i softverskih zahtjeva, projektiranje strukture podataka, definiranje arhitekture programa, kodiranje, testiranje i održavanje. Izabrati odgovarajuću metodu koja će se primijeniti u razvoju nije jednostavno, s obzirom na to da su raspoložive brojne mogućnosti. Metode se mogu međusobno kombinirati nezavisno od toga koja je metoda najpogodnija za rješavanje danog problema. Najznačajnija težnja prilikom pravilnog izbora metoda je minimalizacija troškova razvoja i osiguranje visoko kvalitetnog proizvoda.<sup>4</sup>

Alati osiguravaju automatiziranu ili poluautomatiziranu podršku u primjeni metoda. Oni predstavljaju pomoć neophodnu da bi se sistematizirale i što je više moguće automatizirale aktivnosti razvoja softvera kao: upravljanje projektom odnosno planiranje, procjenjivanje, raspoređivanje, modeliranje, analiza, projektiranje, kodiranje, dokumentiranje, testiranje, integracija elemenata sa sustavom, upravljanje konfiguracijom, kontrola kvalitete softvera, upravljanje podacima i dr. Danas gotovo svaka metoda posjeduje određeno pomoćno sredstvo, instrument, alat. Kada su alati na takav način integrirani oni se nazivaju Computer Assisted Software Engineering ili skraćeno CASE alati. CASE tehnologije pokrivaju područje od pojedinačnih alata za automatizaciju određenih zadataka u razvoju softvera do cjelovitih rješenja za automatizaciju izrade većine koraka u razvoju softvera kao cjeline. CASE tehnologije ne predstavljaju zamjenu za bilo koju metodu ili tehniku razvoja, već samo dodatak metodi ili tehnici u generiranju kvalitetnog proizvoda. Njihovo korištenje je interaktivno, prilagođeno korisniku (razvojnem inženjeru) uz naglasak na upotrebi grafike.<sup>5</sup>

---

<sup>4</sup> hrcak.srce.hr/file/151922 (preuzeto 20.05.2015.)

<sup>5</sup> hrcak.srce.hr/file/151922 (preuzeto 20.05.2015.)

CASE alati su tehnologije koje ne zamjenjuju metode razvoja softvera nego pružaju podršku samim metodama u razvoju softvera. Oni su namijenjeni automatizaciji procesa razvoja softverskog proizvoda.

CASE tehnologija	Metodologije i razvojni alati	Sustavi upravljanja bazama podataka
Promod PLUS ( <a href="http://www.gesys.com/pro-mod.htm">http://www.gesys.com/pro-mod.htm</a> )	Yourdon, de Marco, Hatley/Pirbhai; Uniface Six, Source Pilot, C	Sybase, Oracle, Informix, Ingres
Orade Designer ( <a href="http://www.orafaq.com/wiki/Designer">http://www.orafaq.com/wiki/Designer</a> )	J. Martin, de Marco, Ernst & Young; Uniface Six, CASE Generator SQL, Forms	Oracle, DB2
Westmount I-CASE ( <a href="http://www.unlcsi.cuny.edu/faqs/software-engineering/blurb/Westmount.html">http://www.unlcsi.cuny.edu/faqs/software-engineering/blurb/Westmount.html</a> )	Ward-Mellor, de Marco, Chen, SSADM; Ingres 4GL, Informix 4GL	Ingres, Informix, SQL
PTECH ( <a href="http://en.wikipedia.org/wiki/Ptech">http://en.wikipedia.org/wiki/Ptech</a> )	Martin-Odell, OOAD; ugradeni C++ generator koda	OODBMS
Paradigm Plus ( <a href="http://www.encyclo.co.uk/define/Paradigm%20Plus">http://www.encyclo.co.uk/define/Paradigm%20Plus</a> )	Rumbaugh OMT, Martin-Odell, OOIE, Booch OOAD, Coad, Yourdon; ProtoScript, C, C++, Ada, SmallTalk, PowerBuilder, SQL, JAVA, Corba IDL, Visual Basic, Visual Studio	ORACLE 7, dBase, DB2, uniSQL, Access, Centura, SQLBase, Sybase/SQL, objectStore, gemStone
Rational CASE family ( <a href="http://www-01.ibm.com/software/rational/">http://www-01.ibm.com/software/rational/</a> )	Rumbaugh OMT, Booch OOAD, Jacobson Objectory Use Case, UML;C, C++, Forte, Java, SmallTalk, PowerBuilder, Gupta, SQLWindows, VisualBasic	Oracle 7, Sybase, SQLBase, SQLServer, Watkom SQL, Ansi SQL
CA ERwin Modeling ( <a href="http://erwin.com/">http://erwin.com/</a> )	IDEF1X, IE, DM	DB2, IDS (Informix), MySQL, Oracle, Progress,SQL Server, Sybase, Teradata
PowerDesigner ( <a href="http://www.sybase.com/products/modelingdevelopment/powerdesigner">http://www.sybase.com/products/modelingdevelopment/powerdesigner</a> )	BPEL4WS, BPMN, DTD, ebXML, IDEF, RDBMS, Rich Text Format (RTF), UML 2.0 diagrams, XML, XML Schema	DB2, Informix SQL, Ingres, Interbase, MS Access, MS SQL, MySQL, NonStop SQL, Oracle, PostgreSQL, Sybase, Teradata
Enterprise Architect ( <a href="http://www.sparxsystems.com.au/">http://www.sparxsystems.com.au/</a> )	UML, DDS, MDA, UPDM, Corba IDL, SysML, BPEL, ArchiMate, WSDL, DoDAF, BPMN, TOGAF, SoaML, MODAF, SPEM	DB2, Firebird/InterBase, Informix, Ingres, MS Access MS SQL Server, MySQL, Oracle, PostgreSQL, Sybase

Tablica 1. Najpoznatije CASE tehnologije

(Izvor: hrcak.srce.hr/file/151922; datum pristupa 20.05.2015.)

Tablica 1. Prikazuje tablicu s imenima i web linkovima najpoznatijih CASE tehnologija.

Tablica 1. prikazuje najpoznatije CASE tehnologije koje se danas koriste u svijetu te internet stranice proizvođača CASE tehnologija. Uz ime CASE tehnologija prikazane su njihove metodologije i razvojni alati te sustavi upavljanja bazama podataka gdje dominiraju Oracle i MS SQL alati i sustavi upravljanja bazama podataka.

## 2. KVALITETA SOFTVERA

Kvaliteta je jedan od najopširnijih pojmova kojim se u osnovi uvijek podrazumjeva neka pozitivna svojstva proizvoda. Kvaliteta u informatičkom svijetu podrazumjeva pravilno funkcioniranje računalnog sustava ili nekakav pogled od strane korisnika sustava kako bi taj sustav trebao „pravilno“ raditi. Tako se može reći da ako sustav ili softver kojeg koriste pojedinci ili organizacije odgovara korisnikovom viđenju „pravilnog“ rada sustava onda je on njima kvalitetan.

### 2.1. Pojam kvalitete

Koncept kvalitete bilo kojeg proizvoda, pa tako i softverskog ima u načelu dva aspekta. Ti aspekti su da proizvod mora posjedovati neka pozitivna svojstva i biti bez nedostataka. Poduzeća putem svojih marketinških kampanja osiguravaju kupce da kupuju njihove proizvode jer su kvalitetni dok sam kupac ili korisnik tog proizvoda može imati potpunu drugačiju percepciju kvalitete. Navedena pozitivna svojstva često mogu predstavljati individualna očekivanja pojedinaca. Prema definiciji ISO 9000 “kvaliteta je stupanj u kojem značajke proizvoda zadovoljavaju zahtjeve”. U ovoj definiciji su uključeni objektivni i subjektivni pogledi. Objektivni je “stupanj zadovoljavanja zahtjeva” a subjektivan “ što su to zahtjevi i tko ih postavlja?”

Na Zemlji se nalazi mnoštvo ljudi od kojih svako ima svoju definiciju kvalitete. Općenito se može reći da kvaliteta označava vrijednost, valjanost neke stvari, njenu primjerenost određenim uzorima, zahtjevima, normama (kvaliteta prirodnih materijala, kvaliteta industrijskih proizvoda, kvaliteta trgovačke robe te kvaliteta tehničkih i umjetničkih radova). Dr. Juran riječ „kvaliteta“ definira na dva načina, i to:

1. „Kvaliteta je zadovoljstvo kupca“,
2. „Kvaliteta je prikladnost za upotrebu“<sup>6</sup>

Kvaliteta se tako može gledati kao nekakva nevidljiva razina tolerancije prednosti i mana određenog proizvoda ili usluge od strane korisnika tj. kupca. Kada se ta razina prođe tada

---

<sup>6</sup> KONDIĆ, Ž. (2002.) Kvaliteta i ISO 9000 – primjena -. Varaždin: TIVA



korisnik gubi zanimanje jer mu se proizvod više ne sviđa tj. više mu nije kvalitetan. Ovo se može prikazati na primjeru softvera kojeg korisnik koristi. Dokle god korisnik koristi određeni softver za svoje potrebe on mu je kvalitetan, a kada dođe neka superiornija verzija ili drugi softver s bojim funkcijama, korisnik ga mijenja jer mu je postao nekvalitetan.

### *2.1.1. Kvaliteta kao tržišni čimbenik*

Kada potencijalni kupac određenog proizvoda ili usluge odluči kupiti proizvod ili uslugu gledati će uvijek kako da prođe najpovoljnije. Kod tog odabira kupcu se javljaju tri čimbenika pomoću kojih on odlučuje kupiti specifičan proizvod ili uslugu. Ta tri čimbenika su kvaliteta, cijena i rok isporuke proizvoda ili datum korištenja usluge tj. vrijeme.

Kod ova tri čimbenika kvaliteta je jedini čimbenik koji se nalazi u mraku te kojeg kupac treba otkriti. Prvenstveno pri odabiru proizvoda ili usluge kupac odlučuje što želi kupiti. Imajući na umu svoje zahtjeve i želje kupac selekcijom odabire proizvod kojeg on percipira kvalitetnijim nego ostali. Prilikom tog odabira često je kvaliteta najvažniji od sva tri čimbenika. Ponekad nije važno da li je proizvod skup ili jeftin ili gdje se nalazi. Ako ga kupac percipira kao nešto što mu je potrebno ili ga jednostavno mora imati on njemu stvara vlastitu kvalitetu kojom ga razlikuje od ostalih proizvoda ili usluga. Može se reći da kupac ima neku percepciju kvalitete ali on ne odlučuje o njoj. Na primjer mnogi kupci kupuju Citroen a ne Mercedes, ali zbog toga ne možemo reći da je Citroen kvalitetniji.

Kod razvoja softvera jedino se mogu u potpunosti zadovoljiti dva od tri navedena čimbenika. Ako se informatičari fokusiraju na kvalitetu i unutar određenog proračuna kojeg je korisnik izdvojio za softver, duže će vremena trebati za izradu softvera. Ako je vremenski rok postavljen i proračun definiran tada će radi vremenskog ograničenja kvaliteta softvera biti lošija.

### 2.1.2. *Kvaliteta sa stajališta kupca, proizvođača, tržišta i društva*

Kvaliteta se može gledati s različitih strana a ne samo s strane kupca. Osim kupca proizvoda ili usluge na kvalitetu gledaju i proizvođači/pružatelji proizvoda/usluga, tržište na kojem se nalazi proizvod ili usluga te se kvaliteta gleda sa stajališta samog društva u kojem kupac i proizvođač žive.

Kvaliteta sa stajališta kupca/potrošača se može gledati kao nekakva razina koju kupac percipira u svojoj glavi. Ta razina predstavlja uporabnu vrijednost proizvoda ili usluge do koje ona zadovoljava određenu potrebu (zahtjev ili funkciju). Tako možemo reći da je kvaliteta sa stajališta kupca neki stupanj vrijednosti proizvoda ili usluge koji zadovoljava određenu potrebu. Važno je naglasiti da kupci razlikuju kvalitetne proizvode i usluge, ali kupuju ono što mogu platiti.

Proizvođači na kvalitetu gledaju s drugačijeg stajališta od kupaca. S njihovog stajališta kvaliteta pokazuje koliko je određeni proizvod uspio na tržištu. Tako se kvaliteta s stajališta proizvođača može podijeliti na:

1. **Kvalitetu koncepcije.** Ona se gleda kao polazna točka u razvoju novog proizvoda ili usluge koja obuhvaća sve potrebno za razvoj kvalitetnog proizvoda ili usluge. Kvaliteta koncepcije pokazuje koliko su proizvođači idejnim projektom pogodili zahtjeve i želje kupca. Kvaliteta koncepcije se može povezati sa životnim ciklusom razvoja softvera gdje programeri skupljaju sve potrebne informacije o ukusima i željama korisnika njihovog softvera. Vrijedi napomenuti da promašena koncepcija u većini slučajeva biva propali proizvod.
2. **Kvaliteta konstrukcije.** Ova kvaliteta predstavlja odnos uporabnih vrijednosti dvaju konceptijskih jednako kvalitetnih proizvoda. Npr. postoje dva laptopa sa istom konfiguracijom ali su od drugačijih proizvođača te samim time neki njihovi dijelovi rade brže ili sporije od istih u drugom laptopu. Kod njihovih performansi se može uspoređivati jednostavnost izvedbe, pouzdanost konstrukcije, jednostavnost održavanja, jednostavnost montaže i demontaže itd.
3. **Kvaliteta izrade.** Ona se gleda kao razina do koje je proizvođač kadar realizirati kvalitetu izrade svih potrebnih dijelova potrebnih za izradu glavnog proizvoda. Pod potrebne dijelove spadaju svi poluproizvodi i gotovi proizvodi. Unaprijeđenjem

sustava proizvodnje proizvoda ili nadogradnja u novije verzije softvera omogućuju da se kod kvalitete izrade kvaliteta može lako, brzo i jednostavno poboljšati.

Kvaliteta sa stajališta tržišta predstavlja stupanj do kojeg proizvod ili usluga više (ne)zadovoljava određenog kupca u odnosu na proizvode ili usluge konkurencije. Ako postoji samo jedna vrsta robe na tržištu ona je najkvalitetnija.

Kvaliteta sa stajališta društva u kojem se nalaze kupac i proizvođač prokazuje stupanj do kojeg su određeni proizvodi ili usluge prošli zakone, norme države u kojoj su se potvrdili kao roba, gdje su ostvarili dobit.

## **2.2. ISO 25000; SquaRE (Software product Quality Requirements and Evaluation)**

Prije nego što se krene u objašnjavanje CMM-a kao metode za osiguranje kvalitete softvera vrijedi napomenuti važnu ISO normu koja definira sam pojam kvalitete softverskog proizvoda. ISO/IEC 25000: Software engineering – Software product Quality Requirements and Evaluation (SquaRE) je ISO standard objavljen godine 2005. Koji je nasljedio odrednice norme ISO 9126 koja je bila objavljena ranije. ISO 25000 je norma koja je orjentirana na proizvod. Standardi ISO 2500 podijeljeni su u pet podgrupa i to:

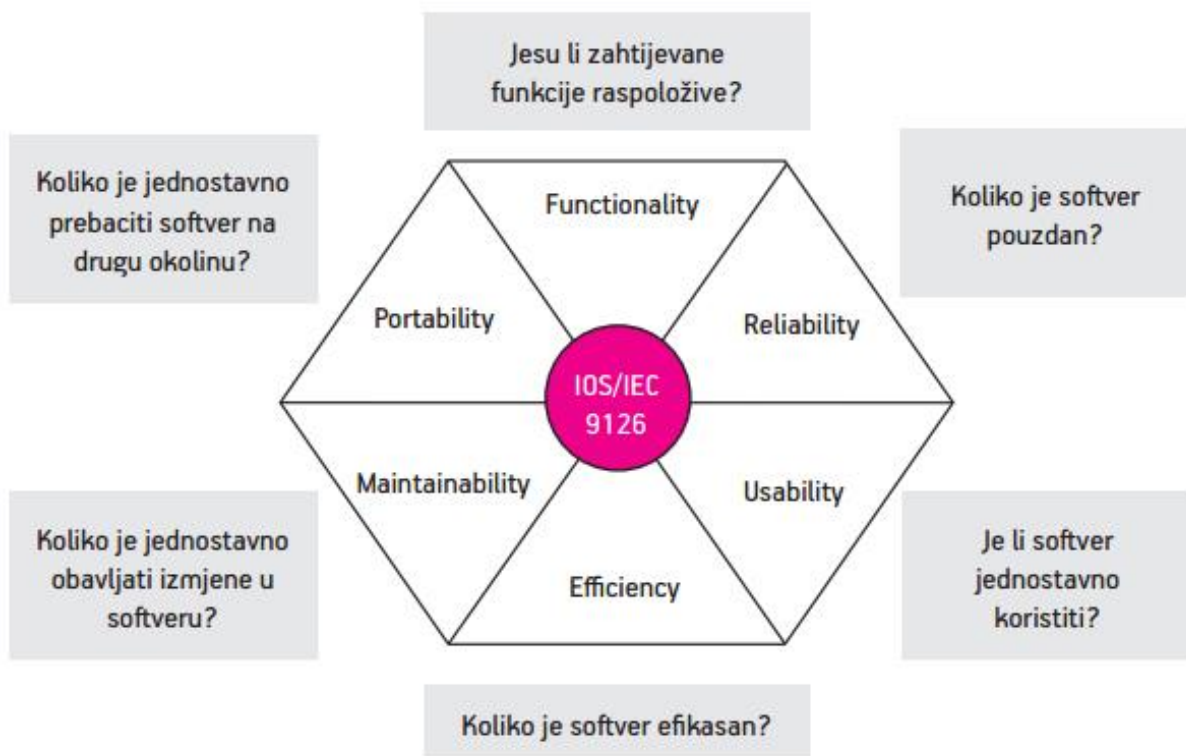
- ISO 2500n: Upravljanje kvalitetom
- ISO 2501n: Modeli kvalitete
- ISO 2502n : Mjerenje kvalitete
- ISO 2503n: Zahtjevi za kvalitetom
- ISO 2504n: Vrednovanje kvalitete

Ove skupine standarda ISO 25000 su kako je prije spomenuto prije nastavak norme ISO 9126. Prva verzija ISO 9126 objavljena je 1991. godine a zadnja 2001. godine te je bila objavljena u četiri dijela ISO/IEC 9126-1 do 9126-4:

- ISO 9126-1: Software engineering - Product quality – Part 1. Quality model (*hr. Model kvalitete*)

- ISO 9126-2: Software engineering - Product quality – Part 2. External metrics (*hr. Vanjska metrika*)
- ISO 9126-3: Software engineering - Product quality – Part 3. Internal metrics (*hr. Unutarnja metrika*)
- ISO 9126-3: Software engineering - Product quality – Part 4. Quality in use metrics (*hr. Metrike u korištenju kvalitete*)

Iz ISO 9126:2001 dijelova i samog naslova ISO 25000 norme vidi se da norma ISO 25000 definira karakteristike softverskog proizvoda i daje smjernice za njihovu upotrebu. Smjernice unutar ISO 25000 standarda utječu na osiguranje i razvoj kvalitete softvera te pružaju managerima odjela za proizvodnju softvera uvid kako softver učiniti kvalitetnim a kupcima softvera obris kvalitete softvera. Po modelu kojeg je norma ISO 25000 naslijedila od ISO 9126, norma predlaže zahtjeve koje softver mora zadovoljiti. Ti zahtjevi su podijeljeni u 6 glavnih karakteristika i 21 pod karakteristiku raspodijeljenih na temelju vanjskih (orijentiranih na kupca tj. korisnika softvera) i unutarnjih (orijentiranih na poboljšanja i održavanje softvera) karakteristika.



Slika 3. Karakteristike norme ISO 25000

(Izvor: [http://www.politehnika-pula.hr/download/repository/Techne\\_br13\\_svibanj\\_2012%5B1%5D.pdf](http://www.politehnika-pula.hr/download/repository/Techne_br13_svibanj_2012%5B1%5D.pdf), datum pristupa 1.7.2015.)

Slika 3. Prikazuje karakteristike norme ISO 25000 koje su raspodijeljene po sljedećem modelu:

1. **Funkcionalnost** (*eng. Functionality*) - sposobnost softvera da zadovolji zahtjevima korisnika.
  - Prilagodljivost - sposobnost realizacije specifičnih zadataka
  - Točnost - preciznost rezultata
  - Interoperativnost - sposobnost interakcije sa raznim aplikacijama
  - Sigurnost - sposobnost zaštite od manipulacije podacima i programima od strane neautoriziranih korisnika.
  - Sukladnost - softvera prema raznim normama i pravilima zahtijevanih od strane zakona i sl.
2. **Pouzdanost** (*eng. Reliability*) - sposobnost softvera da održi nivo performansi sistema kada se koristi pod specifičnim okolnostima.
  - Zrelost - sposobnost softvera da spriječi otkaz sistema uslijed greške u softveru.
  - Tolerantnost na greške - sposobnost softvera da održi nivo operativnosti uslijed greške softvera.
  - Oporavljivost - sposobnost softvera povratka u prvobitno stanje nakon ispravka greške softvera.
3. **Upotrebljivost** (*eng. Usability*) - osobina softvera koja zahtijeva da bude lagan za upotrebu od strane korisnika.
  - Shvatljivost - sposobnost softvera da smanji napore korisnika u prepoznavanju logičkih koncepata i aplikacija softvera.
  - Pogodnost za učenje - atribut softvera koji utječe na smanjenje napora korisnika za shvaćanje osnovnih aplikacija.
  - Operativnost - atribut softvera koji utječe na smanjenje napora korisnika pri upotrebi i kontroli softvera.
4. **Efikasnost** (*eng. Efficiency*) - sposobnost softvera da pruži adekvatne performanse.
  - Vremenski odziv - karakteristika softvera koji utječe na brzinu odziva i izvršenje zadatka s obzirom na vanjske naredbe od strane korisnika.

- Korištenje resursa - sposobnost softvera da primjereno koristi vlastite resurse
- 5. **Pogodnost za održavanje** (*eng. Maintainability*) - sposobnost softvera da se promjeni. Promjene mogu obuhvatiti korekcije, poboljšanja ili prilagodbe promjenama u okolini softvera i u zahtjevima funkcionalnosti.
  - Mogućnost analize - sposobnost softvera da se lakoćom dijagnosticiraju uzroci grešaka ili identificiraju dijelovi softvera koje treba mijenjati.
  - Izmjenjivost - sposobnost softvera da se smanje naponi pri promjeni softvera, uklanjanju grešaka ili izmjeni okoline.
  - Stabilnost - ograničavanje neželjenih efekata uslijed promjene softvera ili neočekivanih događaja.
  - Mogućnost testiranja - lakoća testiranja novih promjena.
- 6. **Prenosivost** (*eng. Portability*) - osobina softvera koja omogućava njegovo premještanje iz jednog okruženja u drugo.
  - Prilagodljivost - sposobnost softvera da se s lakoćom prilagodi raznim mogućim okruženjima.
  - Mogućnost instalacije - sposobnost softvera da se s lakoćom instalira u neko specifično okruženje.
  - Koegzistencija - odlika softvera da funkcionira u interakciji sa raznim aplikacijama koje se nalaze u njegovom okruženju.
  - Promjenjivost - odlika softvera da zamjeni druge namjenske softvere sličnih karakteristika u njihovim okruženjima.<sup>7</sup>

Prema navedenim karakteristikama i pod karakteristikama može se vidjeti da su one fokusirane na osiguranje kvalitete softvera. To znači da softver mora biti prilagodljiv na promjene u svojem okruženju i unutar svojih performansi čime se unaprijeđuje i osigurava njegova kvaliteta. Također su karakteristike fokusirane na kupca tj. korisnika softvera tako da softver bude lagan i jednostavan za korištenje korisniku te da zadovoljava njegovim zahtjevima koji utječu na percepciju kvalitete softvera kod korisnika. Vrijedi napomenuti da se svaka karakteristika ispituje putem specijalno zato napravljenih upitnika i tabela. Treba naglasiti da ovisno o vrsti softvera navedeni atributi kvalitete mogu imati različitu težinu. Na

---

<sup>7</sup> SINKOVIĆ, G. (2012.) Upravljanje kvalitetom informacijskih sustava. TECHNE 13. [Online] (Kolovoz). str.13 Dostupno na - [http://www.politehnika-pula.hr/\\_download/repository/Techne\\_br13\\_svibanj\\_2012%5B1%5D.pdf](http://www.politehnika-pula.hr/_download/repository/Techne_br13_svibanj_2012%5B1%5D.pdf) [Pristupljeno: 14.06.2015.]

primjer kod softvera za upravljanje zrakoplovima, o kojem ovise ljudski životi, pouzdanost zasigurno ima puno veću težinu nego kod softvera za obradu teksta.

### **2.3. Upravljanje kvalitetom**

Upravljanje kvalitetom (engl. quality management) je jedan od najvažnijih zadataka suvremenog menadžmenta, koji zbog sve oštrije konkurencije proizvoda na tržištu dobiva još veće značenje. Pojam se donedavno primarno odnosio na kvalitetu proizvoda, ali se proširio na cjelokupnu organizaciju, na ukupno poslovanje. U posljednje vrijeme se za tu orijentaciju poduzeća rabi izraz totalno upravljanje kvalitetom. Upravljanje kvalitetom u najužoj je vezi s zadovoljstvom kupaca, ali i zaposlenih, što su tri bitne odrednice modernog menadžmenta. Samo poduzeća koja sustavno njeguju i razvijaju dobru radnu klimu i odnose prema svojim zaposlenima i suradnicima mogu očekivati visoku kvalitetu svojih proizvoda i usluga, čime osiguravaju i zadovoljstvo kupaca i trajnu sigurnu budućnost. Upravljanje kreativnošću nekad se nazivalo program nulte greške.<sup>8</sup>

#### *2.2.1. Planiranje kvalitete*

Planiranje kvalitete predstavlja način na koji želimo upravljati kvalitetom. Planiranje se sastoji od niza aktivnosti kojima se utvrđuju ciljevi i uvjeti koji se odnose na kvalitetu visokog obrazovanja i primjenu mehanizama sustava kvalitete. Ono predstavlja dio upravljanja kvalitetom usmjeren na određivanje ciljeva kvalitete i utvrđivanje potrebnih provedbenih procesa te odgovarajućih resursa.<sup>9</sup> Pod planiranje kvalitete se nalazi planiranje proizvoda gdje se nalaze utvrđivanje, klasifikacija, određivanje kvalitete i postavljanje ciljeva te uvjeti koji se odnose na kvalitetu i njezina ograničenja. Osim planiranja proizvoda pod planiranje kvalitete se nalazi planiranje upravljanja i rada, elaboracija planova kvalitete te osiguravanje mjera poboljšanja kvalitete.

---

<sup>8</sup> <http://limun.hr/main.aspx?id=13738&Page=2> (datum pristupa 02.08.2015)

<sup>9</sup> <http://struna.ihj.hr/naziv/planiranje-kvalitete/18270/> (datum pristupa 02.08.2015)

### 2.2.2. Kontrola kvalitete

Kontrola kvalitete je postupak ili skup postupaka čiji je cilj provjeriti unaprijed postavljene kriterije kvalitete i udovoljiti zahtjevima klijenata ili kupaca. U svi proizvodnim procesima potrebno je pratiti u kojoj mjeri proizvodi zadovoljavaju specifikacije.<sup>10</sup>

Prema gore navedenoj definiciji može se vidjeti da je kontrola kvalitete važan postupak koji omogućuje da proizvođači i potrošači putem postavljenih kriterija dobiju što žele. Proizvođači dobivaju klijente i dobit dok potrošači dobivaju ispunjenje svojih zahtjeva i želja putem kvalitetnog proizvoda. Tradicionalno za kontrolu kvalitete se može reći da je to skup metoda i postupaka kojima se na osnovi kriterija kvalitete utvrđuje zadovoljavanje postavljenih zahtjeva i želja potrošača. Kontrola kvalitete ima zadatak da uklanja loše proizvode te se provodi nakon nastanka pogreške i ne utječe na uzorke nekvalitete. Danas se na kontrolu kvalitete gleda kao nekakav skup operacija kojima je cilj poboljšati kvalitetu proizvoda i usluga. Kontrola kvalitete obuhvaća monitoring, evaluaciju i održavanje proizvoda.

Prema načinu provođenja kvaliteta se može podijeliti na dvije vrste:

1. **Unutarnja kontrola kvalitete** – kontrolu kvalitete koju provodi proizvođač. Do industrijske revolucije proizvođači su sami kontrolirali kvalitetu svojih proizvoda dok danas u većini slučajeva to obavljaju kontrolori (posebna skupina zaposlenika). Nakon 1987. godine pojavom normi ISO 9000 unutarnja kontrola kvalitete je prerasla u integralni sustav osiguranja i upravljanja kvalitetom.
2. **Vanjska kontrola kvalitete** – je ona kontrola kvalitete koju obavlja okruženje poduzeća a sastoji se od korisnika, tržišta i društva. Vanjska kontrola kvalitete je stara koliko i ljudska civilizacija, gdje se u starim zapisima poput Hamurabijevog zakonika mogu naći kazne za nekvalitetnu gradnju kuća. Danas se vanjska kontrola kvalitete može podijeliti na indirektnu koju čini konkurencija i tržišni udio te na direktnu koju čine norme, zakoni te zahtjevi.

---

<sup>10</sup> ŠIŠKO KULIŠ M. I GRUBIŠIĆ D. (2010.) Upravljanje kvalitetom. Split: Sveučilište u Splitu, Ekonomski fakultet



### 2.2.3. Osiguranje kvalitete

Osiguranje kvalitete predstavlja aktivni pristup planiranju i razvoju kvalitete. Postiže se kada se kontrola kvalitete proširi u bavljenje kvalitetom u svim fazama razvoja, nastanka, proizvodnje i korištenja proizvoda ili usluga.<sup>11</sup>

Osiguranje kvalitete je skup alata i postupaka koji služe za pronalaženje grešaka i nesklada u proizvodnji što na kraju utječe na kvalitetu proizvoda ili usluge. Osiguranje kvalitete je uključeno u sve faze poslovanja.

Osiguranje kvalitete softvera obuhvaća cijeli proces razvoja softvera, koji uključuje procese kao što su definiranje zahtjeva korisnika, softverski dizajn, kodiranje, kontrola source code-a<sup>12</sup>, upravljanje konfiguracijom softvera, testiranje softvera, kontrola datuma izlaska softvera i integracija softvera. Osiguranje kvalitete osigurava da se izrađuje i pohranjuje dokumentacija tijekom razvoja softvera kao podrška održavanju i usavršavanju softvera. Također je važno da se u određenim vremenskim intervalima provode pregledi kontrole osiguranja softvera poput auditiranja. Najvažnije je potrebno da se za razvoj i osiguranje softvera koriste definirane procedure te da se kontrola osiguranja kvalitete provodi prema normama i metodama osiguranja softvera. Najpoznatije norme i metode za osiguranje kvalitete softvera su ISO norme i CMMI metode o kojima će sljedeće biti govora.

---

<sup>11</sup> ŠIŠKO KULIŠ M. I GRUBIŠIĆ D. (2010.) Upravljanje kvalitetom. Split: Sveučilište u Splitu, Ekonomski fakultet

<sup>12</sup> Izvorni kod (eng. source code) nekog programa obično je skup tekstualnih datoteka, ali može uključivati i druge sporedne dijelove, poput grafičkih elemenata, zvukova i sl. Tekstualne datoteke sadrže skup naredbi (instrukcija) napisanih u nekom programskom jeziku.

### **3. CMM (Capability Maturity Model) – METODA ZA OSIGURANJE KVALITETE SOFTVERA**

U današnjem svijetu informacijskih tehnologija osiguranje kvalitete softvera zauzima važno mjesto u životnom vijeku samog softvera. Putem rada na različitim poslovima poput održavanja softvera programeri se trude držati svoj proizvod kvalitetnim i sigurnim. Danas postoji tisuće različitih softvera namijenjenih za različite zadaće. Oni su svi proizvodi te za njihovo optimalno funkcioniranje potrebno je koristiti nekakve smjernice za osiguranje kvalitete. Od samih početaka razvoja softvera postojala je želja za pronalaženjem cilja uspješne kontrole osiguranja kvalitete softvera koja se može naći u raznim metodama osiguranje kvalitete poput CMM-a, ISO normi te ostalih metoda poput Bootstrapa. U ovom poglavlju biti će govora o CMM metodi za osiguranje kvalitete softvera.

#### **3.1.CMM pojam**

Od prvih inačica softvera pa sve do danas programeri su bili suočeni s mnoštvom softverskih problema koje su trebali riješiti. Kod otkrivanja softverskog problema uvijek se javljalo pitanje kako taj problem riješiti. Bez komunikacije i dogovora između programera i njihovih managera oko popravka tog otkrivenog problema softver ostaje zapušten i s greškama. Da bi se riješio taj problem u komunikaciji i ucrtao put ka riješenju softverskog problema nastao je CMM. Vrijedi napomenuti da je CMM nastao da daje smjernice u rješavanju softverskih greški i problema u softverskoj organizaciji a ne da popravljaju navedene greške.

CMM (Capability Maturity Model) ili model za procjenu zrelosti procesa razvoja softvera služi za identifikaciju problema dijelova softvera i aktivnosti u organizaciji koje se mogu poboljšati. Tijekom rada na softveru programeri primjećuju da kvaliteta razvojnog procesa snažno i pozitivno utječe na kvalitetu softvera, te je važno identificirati sve ključne dijelove softvera za daljnje unaprijeđivanje i osiguravanje kvalitete. Tako CMM pruža softverskim organizacijama smjernice kako kontrolirati procese razvoja softvera. On je napravljen da pomaže softverskim organizacijama u odabiranju pravih strategija za unaprijeđenje procesa tako da određuje trenutnu zrelost procesa i identificira najkritičnije probleme kvalitete

softvera. Tijekom rada na rješavanju i otkivanju softverskih problema organizacija polagano podiže kvalitetu softverskog procesa što dovodi do povećanja sposobnosti softvera. Najveći problemi koji se mogu naći u softverskoj organizaciji su troškovi razvoja i održavanja softvera, razni bugovi<sup>13</sup>, datumi za isporuku finalne verzije softvera itd. Može se reći da greške u softveru i organizaciji u većini slučajeva plaća korisnik jer kupuje softver kojemu isporuka kasni te čije je održavanje teško i skupo.

### **3.2.Povijest CMM-a**

CMM kakvog danas poznajemo imao je dugu povijest koja ide unazad do 1930-tih godina kada je Walter Shewart američki fizičar, inženjer i statističar naveo principe statističke kontrole kvalitete. Na njegovu prvobitnu ideju kontrole kvalitete nadovezali su se poznati gurui kvalitete Edwards Deming i Joseph Juran koji su je dodatno razvili i uspješno primjenili u svojim radovima. Knjiga još jednog poznatog gurua kvalitete Philip-a Crosby-a „Quality is Free“ (hr. Kvaliteta je besplatna) inspirirala je okvir za određivanje zrelosti softverskog procesa, kojega je uspješno razvio Watts Humphrey na SEI-u (Software Engineering Institute) u Pittsburgu 1987. godine. Godinu prije javile su se prvobitne verzije okvira zrelosti softvera u konceptu razina zrelosti koje su postale osnova CMM-a za korištenje u softverskoj industriji. Početak razvoja CMM-a bilo je istraživanje američke vlade u problem razvoja softvera za vojne svrhe, kod čijeg je procesa cijena razvoja uvijek išla iznad zadanog budžeta. Tako je istraživanje na SEI-u dovelo do razvoja CMM modela. Taj model je bio financiran od strane američke vlade kojoj je glavni cilj tog istraživanja bila analiza rada proizvođača softvera koji su razvijali softver za potrebe Ministarstva odbrane SAD-a. CMM je tada služio ministarstvu obrane SAD-a da uspješnije pronalazi kvalitetnije partnere za razne projekte.

Od 1990. godine kada CMM počinje biti aktivan uz pomoć američke vlade i industrije. Razvija se u rafiniran model koji je sve do 1997. godine savršen model za unaprijeđenje softverskog procesa. Do 2002. godine nije bilo većih promjena u CMM modelu te je iste godine bio zamijenjen sa CMMI (Capability Maturity Model Integration) od strane Carnegie Mellon Sveučilišta gdje djeluje SEI. Navedeno je da CMMI dodatno unaprijeđuje proces razvoja softvera ne samo u softverskom projektu nego i u cijeloj organizaciji tijekom svih

---

<sup>13</sup> Bug je izraz koji se koristi za opisivanje greške, pada softvera, ili neispravnosti u radu nekog softvera na računalnom sustavu. Većina bugova nastaju iz pogreške u izvornom kodu, dizajnu softvera itd.

razina zrelosti. Daljnji razvoj rezultirao je novim verzijama CMMI-a te je tako 2002. godine izašla CMMI 1.1 verzija te pet godina kasnije u kolovozu 2006. godine CMMI 1.2 verzija.

Specijalizacijom CMM modela je nastalo više njegovih vrsta namenjenih različitim dijelovima razvoja softvera. Tako danas postoji više vrsta CMMI-a od kojih su poznati:

1. CMMI for Development (CMMI-DEV) - Razvoj proizvoda i usluga
2. PCMM (People) – Razvoj managementa i organizacije
3. CSCMM (Cloud Service) – Razvoj softvera u cloud service okruženju
4. CMMI for Acquisition (CMMI-ACQ) - Objedinjavanje proizvoda i usluga
5. CMMI for Services (CMMI-SVC) - Uspostavljanje, upravljanje i pružanje usluga

Vrijedi napomenuti da nije pametno mješati razne verzije CMMI-a u organizaciji poduzeća jer može doći do problema u funkcioniranju razvoja i osiguranja kvalitete softvera i ostalih proizvoda i usluga.

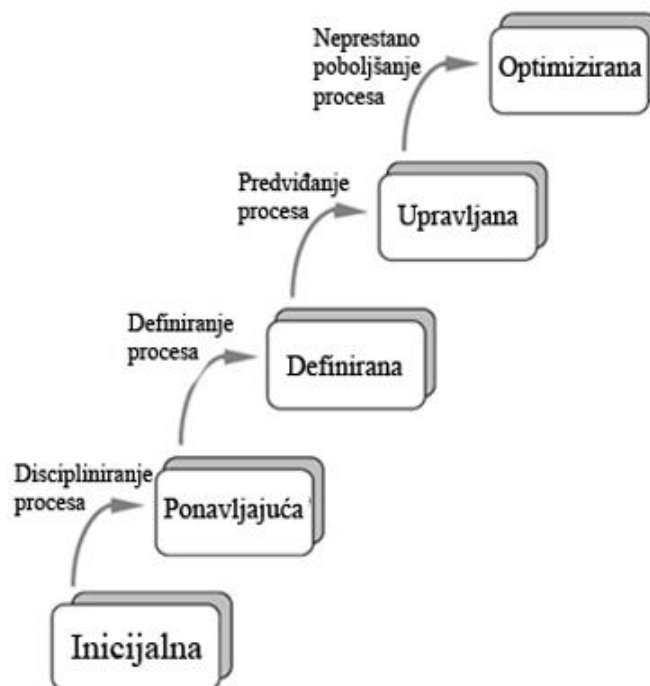
### **3.3.Razine zrelosti CMM-a**

Razvijanje procesa softvera je neprekidan proces koji se sastoji od malenih koraka kojima se unaprijeđuje rad softvera i njegova kvaliteta. To može biti naporan posao za programere odnosno softverske inženjere i organizaciju u kojoj rade ako su aktivnosti rada na softveru neusklađene. CMM je napravljen da pruža okvir za organiziranje tih koraka u razvoju softvera te ih dijeli u pet razina zrelosti koje pomažu softverskim inženjerima u neprekidnom poboljšanju osiguranja kvalitete softvera. Tih pet razina zrelosti pomažu u definiranju ljestvice zrelosti softvera gdje se procjenjuje sposobnost softverskog procesa. Također ovih pet razina pomažu prioritzirati organizaciji rad na daljnjem poboljšanju.

Za razinu zrelosti može se reći da je definirana platforma za postizanje zrelosti softverskog procesa te svaka razina pruža temelje za kontinuiran napredak softverskog procesa. Svaka razina se sastoji od ciljeva koji služe za određivanje stabilnosti softverskog procesa. Također prelaskom na svaku razinu zrelosti softvera utvrđuje se nova komponenta u razvoju softvera što povećava sposobnost procesa organizacije. Prva razina zrelosti služi za uspostavljanje temelja za usporedbu poboljšanja procesa na višim razinama zrelosti. Od druge do pete razine zrelosti nalaze se razine koje je moguće opisati nizom aktivnosti koje organizacija provodi

radi poboljšanja softverskog, ali i svakog drugog procesa, te aktivnostima koje se provode u okviru svakog projekta organizacije.

Sve razine zrelosti mogu se podijeliti na nekoliko ključnih procesa koji označavaju ona područja na koja bi se organizacija trebala fokusirati da bi unaprijedila svoj softverski proces. Područja ključnih procesa identificiraju probleme s kojima se organizacija mora suočiti kako bi ostvarila novu razinu zrelosti. Također svaki dio razina zrelosti se sastoji od niza aktivnosti koje, kada su jednom izvršene, ostvaruju ciljeve potrebne za poboljšanje sposobnosti procesa.



Slika 4. Razine zrelosti CMM-a

Slika 4. Prikazuje pet razina zrelosti CMM-a koje će biti objašnjene u sljedećem tekstu. (Izvor: <http://ibiblio.org/gferg/ldp/SCM-OpenSource/cmm.png>; datum pristupa 15.07.2015)

Razine zrelosti CMM-a prikazane u slici 4. podjeljene su na:

1. **Inicijalna razina:** Procesni na ovoj razini su obično nedokumentirani i u stanju stalne promjene. Na ovoj razini softverskim procesima se upravlja neorganizirano te se neplanirano i nekontrolirano ažuriraju od strane korisnika ili nekih događaja što daje kaotičnu ili nestabilnu okolinu.

2. **Ponavljajuća razina:** Na ovoj razini postoje neki procesi koji se ponavljaju i to s konzistentnim rezultatima. Također na ovoj razini procesna disciplina nije stroga te pomaže da se postojeći procesi održavaju u vrijeme stresa.
3. **Definirana razina:** Na ovoj razini postoji skup standardnih i dobro dokumentiranih procesa koji se s vremenom i poboljšavaju.
4. **Upravljana razina:** Na ovoj razini organizacija je uspostavila opsežna mjerenja i analize procesa. Također na ovoj razini, mjerenje procesa može identificirati načine podešavanja i prilagodbu procesa za pojedine projekte, bez gubitaka kvalitete ili odstupanja od specifikacije softvera.
5. **Optimizirana razina:** Zadnja razina gdje je poduzeće ostvarilo sve gore navedene korake te se sada može posvetiti praćenju performansi, te podešavati svoje procese kako bi se poboljšala produktivnost i smanjile nepravilnosti kroz cijelu organizaciju.

Razina	Obilježja	Ključni procesi
Inicijalna razina	<b>Kaos</b>	*preživljavanje
Ponavljajuća	<b>Pojedinačna kontrola</b>	*upravljanje konfiguracijom *osiguranje kvalitete *upravljanje kupljenim soft. *praćenje projekta razvoja softvera *planiranje razvoja softvera *upravljanje korisničkim zahtjevima
Definirana	<b>Institucionalizacija procesa</b>	*pregledi, revizije softvera *koordinacija među odjelima *integralno upravljanje softverom *plan i program treninga *definicija poslovnih procesa, fokus

Upravljana	<b>Mjerenje procesa</b>	*upravljanje kvalitetom *mjerenje i analiza poslovnih procesa
Optimizirana	<b>Unapređenje</b>	*upravljanje promjenama *praćenje tehnoloških inovacija *prevencija problema, ispada

Tablica 2. Obilježja i ključni procesi razina zrelosti<sup>14</sup>

Tablica 2. Prikazuje obilježja i ključne procese razina zrelosti softvera, kroz koje se može vidjeti evolucija i razvijanje organizacije i softvera kojeg ona proizvodi. Također vrijedi napomenuti da kroz ove razine CMM ne garantira organizacijama uspjeh te im samo daje smjernice kako bi bilo moguće osiguravati kvalitetu softvera i njegov razvoj.

Ciljevi razina zrelosti CMM-a prikazani na tablici 2. predstavljaju ključne procedure koje se mogu upotrijebiti za određivanje uspješnosti implementacije područja ključnih procesa od organizacije ili projekta. Ciljevi predstavljaju domet do kuda organizacija može ići te samim time i ograničenja jer ciljevi predstavljaju nekakvu granicu. Uz domet i ograničenja ciljevi također predstavljaju i svrhu svakog područja ključnih procesa. Ciljevi u CMM-u kod korištenja ključnih procedura u određenim situacijama određuju uspješnost softverskih projekata. Kod implementacije područja ključnih procesa ciljevi imaju svrhu i alata za odabir kako bolje organizacija može upravljati softverskim projektima.

Uz ciljeve razina zrelosti CMM-a, postoje i ključni procesi koji opisuju aktivnosti i infrastrukturu područja ključnih procesa koja pridonosi uspješnoj implementaciji i institucionalizaciji područja ključnih procesa. Specifične procedure koje je potrebno izvršiti u okviru svakog područja ključnih procesa razvijat će se sazrijevanjem organizacije. Ključne procedure opisuju sve ono što mora biti izvršeno, no ne ističu na koji način bi koji proces trebao biti implementiran. Svrha ovih ključnih procedura je davanje uvida u ključne elemente uspješnog procesa. Svaka ključna procedura u svakom području ključnih procesa opisana je

<sup>14</sup> CINDRIĆ J. (2009.) ZRELOST ORGANIZACIJE; stručni članak

pomoću skupa tzv. zajedničkih obilježja (common features). Zajednička obilježja su atributi koji nam pokazuju da li je implementacija i institucionalizacija područja ključnih procesa uspješna, ponovljiva i trajna. Postoji pet zajedničkih obilježja:

1. **Obveza izvođenja:** niz aktivnosti koje organizacija mora izvršiti kako bi osigurala provedbu procesa i njegovo trajanje (npr. uspostavljanje organizacijske politike).
2. **Sposobnost izvođenja:** preduvjeti koji moraju postojati u projektu ili organizaciji da bi se softverski proces kompetentno implementirao. To obično podrazumijeva sredstva, organizacijsku strukturu i obuku.
3. **Izvršene aktivnosti:** procedure potrebne za implementaciju područja ključnih procesa. To obično uključuje uspostavu plana i procedura, izvođenja poslova, praćenje poslova i provođenje korektivnih zahvata.
4. **Mjerenja i analiza:** potreba za izvršenjem mjerenja procesa i analize rezultata tih mjerenja. To obično podrazumijeva primjere mjerenja koja bi mogla biti izvršena da bi se ustanovio status i uspješnost izvršenih aktivnosti.
5. **Procjena implementacije:** osiguravanje da su potrebne aktivnosti izvršene u skladu s utvrđenim procesom. To obično podrazumijeva ocjene i reviziju koju provodi menadžment i osiguranje kvalitete proizvoda ili usluge.<sup>15</sup>

Ciljevi i ključni procesi se vežu za područja ključnih procesa koja postoje u razinama zrelosti CMM-a. Područja ključnih procesa se nalaze u svakoj razini zrelosti osim inicijalne te oni označavaju područja na koja se poduzeće i organizacija trebaju fokusirati da bi unaprijedili softverski proces. Za područja ključnih procesa može se reći da identificiraju probleme s kojima se organizacija mora suočiti kako bi ostvarila novu razinu zrelosti. Na svakoj razini područja ključnih procesa predstavljaju aktivnosti koje ostvaruju ciljeve potrebne za poboljšanje sposobnosti procesa. Kako postoji na milijune različitih poduzeća i organizacija koje imaju različite ciljeve te se samim time kod svakog poduzeća i organizacije razlikuju područja ključnih procesa, ciljevi te samo funkcioniranje softverskih procesa i projekata. CMM ne opisuje detaljno sve procese potrebne u razvijanju i održavanju softvera, odnosno poslovnog procesa, nego samo one koji su najkorisniji ili najuspješniji u unapređenju sposobnosti procesa u organizaciji. Kako bi se ostvarila određena razina zrelosti prvo moraju biti ostvarena i zadovoljena područja ključnih procesa. Da bi se zadovoljilo područje ključnih

---

<sup>15</sup> CINDRIĆ J. (2009.) ZRELOST ORGANIZACIJE; stručni članak



procesa, potrebno je ostvariti njegove pripadajuće ciljeve.

### *3.3.1. Inicijalna razina*

Inicijalna razina predstavlja prvu razinu zrelosti softvera koja se naziva i kaotičnom. Razlog toga je prvenstveno preživljavanje organizacije gdje su procesi neorganizirani. Na ovoj razini poslovni i softverski proces organiziran je kao ad hoc<sup>16</sup>. U inicijalnoj razini nema ključnih procesa te se sve svodi na preživljavanje a prelazak na drugu razinu ovisi o trudu i željom za uspjehom. Također se na na inicijalnoj razini poduzeće mora boriti u nestabilnom okruženju za razvoj softvera. U većini slučajeva na inicijalnoj razini planirani rokovi se ne ispunjavaju te nastaju problemi koji mogu uzrokovati prestanak razvoja softvera. Ova razina predstavlja razdoblje u kojem se nalaze milijuni ideja i ranih verzija softvera od kojih samo nekolicina nastavi svoj put u drugu razinu. Najviše odgovornosti na ovoj razini ima management i razvojni tim jer su oni kotač koji se mora okretati da bi organizacija i razvoj softvera funkcionirali. Management se treba sastojati od ljudi koji znaju što rade kako bi se prošlo kroz razinu bez neplaniranih poteškoća.

Ova razina se uz ad hoc i kaotičnu naziva i herojskom radi toga jer su potrebni snažni, utjecajni i vješti ljudi koji bi vodili organizaciju kroz proces razvoja softvera koji može biti nepredvidljiv. Na primjer projekt može dobiti manju svotu novca od planirane za razvoj softverskog proizvoda. Potrebno je imati odlučne i inteligentne individualce koji se ne boje modificirati razvoj softvera. Ukratko tijekom razvoja softverskog procesa na ovoj razini ovisi sve o sposobnostima tih individualaca tj. njihovim vještinama, znanju i motivaciji da pokreću kotač razvoja softvera kroz razne nepredvidljive rasporede, promjenjive budžete itd. Samo ustrajnim radom i željom za uspjehom.

### *3.3.2. Ponavljajuća razina*

Ponavljajuća razina je druga razina zrelosti koja je za razliku od inicijalne razine stabilnija. Na ovoj razini se postavljaju načini upravljanja procesima razvoja radi čega se donosi red nakon kaotične prve razine te se uspostavljaju načela za upravljanje i

---

<sup>16</sup> Ad hoc označava neko rješenje za neki specifični problem ili zadaću koji se ne generalizira, te koji se ne smije primjenjivati u druge svrhe.

implementaciju projektom. Na ovoj razini troškovi se kontroliraju te nema prevelikih odstupanja u budžetu kao u inicijalnoj razini. Prate se vremenski rokovi i funkcioniranje samih procesa. Za razliku od inicijalne razine ponavljajuća razina ima iskustva s upravljanjem projektima te sadrži više ljudi koji znaju svoj posao. Cilj ove razine je dokumentiranje i ponavljanje koraka koji su doveli do uspješnog završetka zadanih projekata organizacije. Ovdje manageri prate troškove, vremenske rokove i rad na projektu te ispravljaju probleme u trenutku njihovog nastajanja. Tako se može reći da ponavljajuću razinu karakteriziraju procesi koji se ponavljaju s konzistentnim rezultatima te su procesi softverskog projekta pod kontrolom managementa kojemu su realni planovi za projekt zasnovani na performansama prijašnjih projekata.

Za razliku od inicijalne razine, ponavljajuća razina koja je stabilnija sadrži i područja ključnih procesa i ciljeve. Područja ključnih procesa ponavljajuće razine su:

1. **Upravljanje konfiguracijom** kod koje se određuje konfiguracija softvera u određenom vremenskom roku uz neprekidno praćenje promjena konfiguracije i održavanje integriteta konfiguracije kroz životni ciklus softvera. Pod softverskom konfiguracijom se nalaze svi proizvodi koji se dostavljaju naručitelju poput zapisa softverskih zahtjeva i sve ostale dijelove potrebne za razvijanje tih softverskih proizvoda. Ciljevi upravljanja konfiguracijom su planiranje aktivnosti managementa, identifikacija, kreiranje i kontrola odabranih radnih procesa, kontrola promjena poslovnih procesa, informiranje grupa i pojedinaca koji rade na softverskom projektu.
2. **Osiguranje kvalitete softvera** kod koje se osiguranje provodi kako bi managementu pružilo jasan uvid u procese koji se koriste u softverskom projektu. Osiguranje kvalitete se provodi ocjenjivanjem softverskog projekta i njegovih aktivnosti kako bi se utvrdilo da li su u skladu sa procedurama, standardima i zahtjevima te kako bi pružilo projektu i managementu dokumentaciju potrebnu za buduće slične projekte. Tako se iz ovoga mogu vidjeti i ciljevi koji su planiranje aktivnosti osiguranja kvalitete proizvoda, pridržavanje svih aktivnosti standardima, procedurama i zahtjevima, informiranje uključenih grupa i pojedinaca softverskog procesa o aktivnostima i rezultatima osiguranja kvalitete proizvoda.
3. **Upravljanje kupljenim softverom** sadrži sve aktivnosti koje softverski inženjeri i pojedinci koji rade na softverskom projektu obavljaju. Upravljanje kupljenim

softverom se dokumentira radi uspješnog rada softvera i pronalaženja grešaka u softveru. Također upravljanje kupljenim softverom obuhvaća i dokumentaciju proizvođača tog softvera. Ciljevi su tako dokumentirati proces rada softvera i uspješno detektirati greške u sustavu softvera.

4. **Praćenje projekta razvoja softvera** pruža uvid u napredak projekta kako bi management poduzeo korektivne mjere u slučaju da projekt počne odstupati od planiranog. Pod praćenje se nalazi usporedba rezultata sa dokumentiranim procjenama, obvezama i planovima. Tako se iz ovoga mogu vidjeti ciljevi koji su praćenje i usporedba stvarnih rezultata projekta sa planiranima, provođenje korektivnih postupaka ukoliko stvarna dostignuća odstupaju od planiranog.
5. **Planiranje razvoja softvera** služi za razvijanje planova za softverski projekt, te za upravljanje samim projektom. Također planiranje razvoja projekta uključuje stvaranje proračuna za poslove koji se trebaju izvršiti, utvrđivanje obaveza te definiranje plana izvršavanja poslova. Kroz planiranje razvoja softvera se procjenjuje zahtjevnost projekta te potrebna sredstva za sam projekt te se nakon toga pravi vremenski raspored te identificiraju rizici i utvrđuju obaveze. Vrijedi napomenuti da je rezultat planiranja razvoja softvera Plan projekta. Ciljevi planiranja razvoja softvera su dokumentiranje proračuna radi praćenja projekta te dokumentiranje aktivnosti i obveza projekta.
6. **Upravljanje korisničkim zahtjevima** kao zadnje područje ključnih procesa podrazumjeva dokumentiranje korisničkih prijedloga i pritužbi radi uspješnog funkcioniranja softvera.

### 3.3.3. *Definirana razina*

Na definiranoj razini softverski procesi organizacije za stvaranje i održavanje softvera dokumentirani su što uključuje softversku i procesnu dokumentaciju projekta. Tako se može reći da su poslovni procesi organizacije za inženjerske aktivnosti dokumentirani, standardizirani i integrirani u procese organizacije. Na ovoj razini postoje definirani i standardizirani softverski procesi koji su odobreni i provjereni te ih organizacija može koristiti bez ikakvih briga. Softverski procesi na ovoj razini pomažu managementu i tehničkom osoblju kod postizanja bolje efektivnosti svojih projekata. Na ovoj razini se nalazi grupa koja je odgovorna za procesne aktivnosti na projektu. Ta grupa se zove softversko inženjerska procesna grupa (eng. Software engineering process group ili SEPG). Također na ovoj razini se

implementira program za treniranje i učenje osoblja i managera organizacije tako da imaju znanje i vještine potrebne za dodjeljene uloge na softverskom procesu unutar organizacije. Projekti na ovoj razini definiraju organizacijski standardni softverski proces da bi se iz toga stvorio organizacijski vlastiti softverski proces od čega se grade jedinstvene karakteristike projekta organizacije. Ti vlastiti softverski procesi u CMM-u se zovu definirani softverski projekti koji se sastoje od dosljednog, integriranog i definiranog softverskog inženjerstva i managementskog procesa. Kod dobro definiranog softverskog procesa mogu se karakterizirati kriteriji spremnosti, razni inputi, standardi i postupci za obavljanje posla, verifikacijski mehanizmi, outputa i kriterija za završetak procesa. Radi toga jer je softverski proces dobro definiran na ovoj razini, management ima dobar uvid u napredak na projektu.

Za definiranu razinu može se reći da je standardizirana i konzistentna zato jer su softversko inženjerstvo i managerske aktivnosti unutar organizacije stabilni i mogu se ponavljati. Sa uspostavljenim aktivnostima koje se potpuno kontroliraju poput proizvodnih linija, troškova, vremenski raspored i funkcionalnost uspješno se prati kvaliteta softvera. Funkcionalnost softverskih procesa definirane razine tako se može pohvaliti time da organizacija poznaje svoje aktivnosti, da djelatnici znaju što rade pomoću definiranih uloga te su odgovorni u definiranju softverskog procesa. Općenito se smatra da ova razina odgovara certificiranom sustavu prema ISO 9001 uz primjenu smjernica ISO 90003.

Kod definirane razine područja ključnih procesa se odnose na probleme organizacije koja upravlja softverskim projektom. Područja ključnih procesa su:

1. **Pregledi, revizije softvera** koje uključuju učinkovito uklanjanje defekata i pogrešaka verzija poslovnih procesa u najranijim fazama. Također uključuje pregled poslovnih procesa konkurenata kako bi se identificirali defekti i područja na kojima je potrebno izvršiti promjene i korekcije. Ciljevi ovog područja tako su planiranje aktivnosti revizije konkurenata te identifikacija i uklanjanje defekata radnih verzija poslovnih procesa.
2. **Koordinacija među odjelima** omogućava komunikaciju i zajednički rad raznih odjela organizacije kako bi projekt zadovoljio potrebe managementa. Ciljevi koordinacije među odjelima su tako dogovori svih odjela unutar organizacije o zahtjevima

managementa i pojedinih odjela te identifikacija, praćenje i rješavanje problema koji se javljaju unutar tih odjela.

3. **Integralno upravljanje softverom** ima svrhu integracije aktivnosti softverskog inženjerstva i managementa u cjelovit, definiran poslovni proces, izrađen prema standardu definiranog poslovnog procesa. Pod integralno upravljanje softverom se uključuje razvoj definiranog poslovnog procesa softvera i upravljanje projektom koristeći definirani poslovni proces. Novi i trenutni projekti softverskog projekta mogu međusobno izmjenjivati podatke o procesu i naučena znanja radi izbjegavanja mogućih problema koji bi se javljali. Ciljevi intergralnog upravljanja softverom su izrada kompletnog definiranog poslovnog procesa prema organizacijskom standardu poslovnih procesa te planiranje i upravljanje softverskim projektom na temelju definiranog poslovnog procesa.
4. **Plan i program treninga** ima za svrhu potporu razvoju vještina i znanja zaposlenika poduzeća kako bi mogli što bolje i učinkovitije obavljati svoj posao. Da bi se upješno provodio potrebno je prvo identificirati program obuke koji je potreban organizaciji poduzeća, softverskom projektu i pojedincima, a tek onda prema identificiranim potrebama razvoj i provođenje obuke. Ciljevi ovog područja ključnog procesa tako su planiranje aktivnosti obuke, obuka, razvoj vještina i znanja potrebnih za obavljanje potrebnih poslova unutar poduzeća Cilj toga je da zaposlenici dobiju znanje potrebno za izvršavanje njihovih obaveza i poslova.
5. **Fokus i definicija poslovnih procesa.** Fokus predstavlja razvijanje i održavanje poslovnih procesa organizacije i projekta te koordinaciju aktivnosti koje su potrebne za izvršenje, razvoj, održavanje i unapređenje tih procesa. Dok definicija poslovnih procesa predstavlja razvoj i održavanje organizacijskog standarda poslovnih procesa te elemenata procesa kao što su opis životnog ciklusa, smjernice i kriteriji razvoja procesa, baza podataka o poslovnim procesima, te cjelokupna dokumentacija vezana za poslovne procese. Ciljevi fokusa i definicije poslovnih procesa su koordinacija aktivnosti razvoja i unapređenja poslovnih procesa unutar cijele organizacije, identifikacija prednosti i slabosti poslovnih procesa, te razvoj i održavanje procesa.

CMM stupanj zrelosti	Trajanje projekta (mjeseci)	Broj angažiranih osoba*mjeseci	Isporučeni broj grešaka	Prosječni ostvareni troškovi
1	30	600	61	5,5
2	18,5	143	12	1,3
3	15	80	7	0,73

Tablica 3. CMM statistika za projekte od 200.000 LOC<sup>17</sup>

(Izvor: [http://www.politehnika-pula.hr/download/repository/Techne\\_br13\\_svibanj\\_2012%5B1%5D.pdf](http://www.politehnika-pula.hr/download/repository/Techne_br13_svibanj_2012%5B1%5D.pdf); datum pristupa 14.06.2015.)

Tablica 3. Prikazuje softverske proizvode iste veličine ali kroz različite CMM stupnjeve zrelosti od prve inicijalne razine do treće definirane. Tako se može vidjeti što je poduzeće zrelije ono je iskusnije u proizvodnji softver i osiguravanju njegove kvalitete. Na tablici 3. može se vidjeti da se tijekom rasta poduzeća kroz razine zrelosti vrijeme projekta skraćuje, smanjuje se broj angažiranih osoba na projektu. Među ostalim smanjuje se broj grešaka samog softvera što pridonosi većoj kvaliteti samog te prosječni troškovi padaju.

#### 3.3.4. Upravljana razina

Na upravljanjoj razini organizacija postavlja kvantitativne ciljeve kvalitete za softverske proizvode i procese. Produktivnost i kvaliteta su mjerljivi po važnosti aktivnosti softverskog procesa preko svih projekata unutar organizacije kao dio organizacijskog programa za mjerenje. Baza podataka organizacijskog softverskog procesa koristi se da bi prikupljala i analizirala dostupne podatke koji se mogu dobiti iz projekata softversog procesa. Softverski procesi se kontroliraju sa definiranim i konstantnim mjerama. Te mjere uspostavljaju kvantitativnu osnovu za evaluaciju softverskih procesa i proizvoda softverskog procesa. Sužavanjem varijancija u procesnim performansama projekti dobivaju kontrolu nad svojim proizvodima i procesima. Time te performanse dolaze u prihvatljive kvantitativne granice. Proces razvoja poslovnih procesa na upravljanjoj razini može se prikazati kao kvantificiran i predvidljiv jer se kontrolira i provodi unutar mjerljivih granica. Kada se primjeti da bi se zadane granice mogle preći, pristupa se korektivnim akcijama kako bi se

<sup>17</sup> LOC (Lines of Code) – broj naredbi u softveru.

osigurala maksimalna kvaliteta procesa. Time se smanjuje varijabilnost procesa i omogućava bolje predviđanje rezultata i trendova kvalitete procesa.

Ukratko sposobnost softverskog procesa na upravljanoj razini može se prikazati kao predvidljiva jer je sam proces mjerljiv i funkcionira unutar mjerljivih granica. Upravljana razina omogućuje organizaciji da predviđa trendove u procesnoj i proizvodnoj kvaliteti unutar kvantitativnih granica, Kada se te granice pređu poduzimaju se korektivne mjere za ispravljanje situacije. Kod ove razine softverski proizvodi su visoke kvalitete.

Područja ključnih procesa upravljane razine fokusiraju se na uspostavu sustava kvantitativnog softverskog procesa i softverskog proizvoda. Područja ključnih procesa su:

1. **Upravljanje kvalitetom.** Upravljanje kvalitetom se provodi sa svrhom uspostave kvantitativnih pokazatelja kvalitete poslovnih procesa i softvera te postizanja određenih ciljeva kvalitete. Pod upravljanje kvalitetom nalazi se definiranje ciljeva kvalitete, izrada planova za ostvarivanje tih ciljeva te praćenje i prilagođavanje planova, poslovnih procesa i programskog rješenja. Cilj toga je zadovoljavanje potrebe managementa i poslovnog sustava. Također se prati i kvaliteta vlastitog proizvoda ili usluge prema kupcima. Ciljevi upravljanja kvalitetom su planiranje aktivnosti managementa kvalitete, definiranje ciljeva i prioriteta.
2. **Mjerenje i analiza poslovnih procesa.** Kod ovog područja ključnih procesa kvantitativna kontrola performansi procesa uključuje se utvrđivanjem ciljeva performansi definiranog poslovnog procesa. Mjerenje se odvija na osnovi ranijih iskustava i unaprijed definiranih metrika. Mjerenjem performansi procesa i analizom rezultata tih mjerenja izvršavaju se prilagodbe poslovnog procesa kako bi se performanse samog procesa održale unutar prihvatljivih granica. Tako su ciljevi mjerenja i analize poslovnih procesa kvantitativna kontrola performansi definiranog poslovnog procesa te planiranje aktivnosti managementa kvantitativnih procesa.

### 3.3.5. *Optimizirana razina*

Kod optimizirane razine cijela se organizacija fokusira na neprestano unaprjeđivanje poslovnog procesa. Organizacija ima sposobnosti da identificira slabosti i snage procesa sa ciljom prevencije budućih defekata. Podaci o efektivnosti softverskog procesa se koriste za obavljanje analize troškova i koristi novih tehnologija te predlažu promjene organizacijskog softverskog procesa. Kod optimizirane razine se identificiraju inovacije koje su najbolje za softversko inženjerstvo unutar poduzeća. Timovi i grupe koje rade na softverskim projektima na optimiziranoj razini analiziraju defekte softverskog proizvoda u cilju pronalaska uzroka samih defekata. Vršiti se evaluacija softverskih procesa u svrhu prevencije ponavljanja tih defekata. Softverski procesi na optimiziranoj razini se konstantno unaprjeđuju zato jer se organizacije na ovoj razini trude unapređivati i širiti opseg svojih procesnih sposobnosti te samim time unapređivati performanse procesa na projektima. Unapređenja se događaju postepeno uz polagan napredak u postojećim procesima i preko inovacija korištenjem novih tehnologija i metoda.

Kao i kod prethodnih razina optimizirana razina se sastoji od nekoliko područja ključnih procesa koja su:

1. **Upravljanje promjenama.** Upravljanje promjenama se fokusira na kontinuirano poboljšanje poslovnih procesa koji se koriste u organizaciji s ciljem poboljšanja kvalitete procesa te povećanja produktivnosti i smanjenja vremenskog perioda potrebnog za razvoj novih procesa i proizvoda ili usluga. Upravljanje promjenama postepeno preuzima poboljšanja ostalih područja ključnih procesa, praćenje tehnoloških inovacija i prevencija problema te ih čini upotrebljivima za cijelu organizaciju. Ciljevi upravljanja promjenama tako su planiranje kontinuiranog poboljšanja procesa, sudjelovanje cijele organizacije u aktivnostima poboljšanja poslovnih procesa organizacije, neprestano poboljšanje organizacijskog standardnog poslovnog procesa i definiranog poslovnog sustava.
2. **Praćenje tehnoloških inovacija.** Kod praćenja tehnoloških inovacija se vrši identifikacija novih tehnologija. Tako praćenje tehnoloških inovacija uz identifikaciju, sadrži odabir i procjenu novih tehnologija te njihovo uključivanje i uvođenje u organizaciju. Svrha svega toga je poboljšanje kvalitete poslovnih procesa i programskog rješenja koje ga prati te povećanje produktivnosti i smanjenje cikličkog



perioda razvoja proizvoda ili usluge. Ciljevi ovog područja ključnih procesa su planiranje primjene i uvođenje novih tehnologija, procjena novih tehnologija radi utvrđivanja njihovog potencijalnog doprinosa kvalitete i produktivnosti organizacije te implementacija novih tehnologija u čitavoj organizaciji.

3. **Prevenција problema, ispada.** Ima svrhu utvrđivanja uzroka defekata i prevencije njihovog ponovnog pojavljivanja. Prevenција problema analizira defekte koji su se pojavljivali u prošlosti te poduzima specifične akcije za spriječavanje ponavljanja istih pogrešaka u budućnosti. Također se ti defekti mogu identificirati tijekom rada na drugim projektima u ranijim fazama. Prevenција defekata predstavlja stvaranje iskustva za rad na novim projektima te širenje znanja među grupama koje rade na trenutim i budućim projektima. Ciljevi ovog područja ključnih procesa su planiranje aktivnosti sprečavanja pogrešaka, identifikacija i proučavanje najčešćih uzroka defekata, te neprekidno uklanjanje najčešćih uzroka defekata.

Optimizirana razina predstavlja zadnju razinu zrelosti CMM modela koji se razvijao tijekom devedesetih. Vrijedi napomenuti da se uočavanjem nedostataka te pojavljivanjem novih tehnologija razvio CMMI koji će uz svoje vrste i razvoj biti objašnjen u sljedećim poglavljima.

U raznim člancima može se naći puno zanimljivosti vezanih za CMM razine zrelosti softvera. Tako se u časopisu za politehničku obrazovnu teoriju i praksu *TECHNE 13* nalazi da je između 1997. i 2001. godine 1018 poduzeća u SAD ispitano po CMM modelu, s prosječnom zaposlenošću od 100 stručnjaka dalo je sljedeće rezultate: 27% poduzeća bilo je na razini 1, 39% - razina 2, 23% - razina 3, 6% - razina 4 i 5% - razina 5.<sup>18</sup> CMM model je proizveden u SAD-u gdje se i najviše koristi tako da nema puno podataka koji bi govorili na kojoj se razini zrelosti nalaze Europska poduzeća. Ali ima jedan zanimljiv podatak iz Italije 2001. godine procijenjeno je da barem 85% proizvođača softvera ne prelazi prvu razinu<sup>19</sup>, što ukazuje na kaotično okruženje u većini poduzeća Italije. Godine 2005. prema postojećim podacima SEI-a

---

<sup>18</sup> SINKOVIĆ, G. (2012.) Upravljanje kvalitetom informacijskih sustava. *TECHNE 13*. [Online] (Kolovoz). str.15 Dostupno na - [http://www.politehnika-pula.hr/\\_download/repository/Techne\\_br13\\_svibanj\\_2012%5B1%5D.pdf](http://www.politehnika-pula.hr/_download/repository/Techne_br13_svibanj_2012%5B1%5D.pdf) [Pristupljeno: 14.06.2015.]

<sup>19</sup> SINKOVIĆ, G. (2007.) Izvorni znanstveni rad. Str 36. Dostupno na [http://oet.unipu.hr/uploads/media/Economic\\_research\\_Vol.20\\_No.2\\_december\\_2007\\_03.pdf](http://oet.unipu.hr/uploads/media/Economic_research_Vol.20_No.2_december_2007_03.pdf) [Pristupljeno: 14.06.2015.]

u svijetu je postojalo 207 poduzeća sa petom razinom zrelosti od kojih je većina smještena u Indiji.



Slika 5. Certifikat pete razine zrelosti

(Izvor: [http://www.masfak.ni.ac.rs/milan.zdravkovic/files/articles/2005\\_CMM.pdf](http://www.masfak.ni.ac.rs/milan.zdravkovic/files/articles/2005_CMM.pdf); datum pristupa 10.07.2015)

Slika 5. Prikazuje certifikat za uspješno postizanje pete razine zrelosti poduzeća L3 Communications, integrated Systems. Certifikat opisuje da je poduzeće uspješno demonstriralo osobine pete razine zrelosti CMM-a što je rezultat CMM procjene za unaprijeđenje unutrašnjih procesa (eng. CMM Based Assessment for Internal Process Improvement).

Vrijedi napomenuti interesantnu informaciju da je SEI izdao vremensku razliku koja je potrebna da se pomakne iz jedne razine zrelosti u drugu:

1. Od prve do druge razine zrelosti – 24 mjeseca
2. Od druge do treće razine zrelosti – 22 mjeseca

3. Od treće do četvrte razine zrelosti – 32 mjeseca
4. Od četvrte razine do pete razine zrelosti – 16 mjeseci<sup>20</sup>

Država	Razina zrelosti 4	Razina zrelosti 5	Ukupno
India	27	50	77
USA	39	20	59
China	0	2	2
Australia	2	0	2
Canada	0	1	1
Russia	0	1	1
France	1	0	1
Ireland	1	0	1
Israel	1	0	1
Singapore	1	0	1

Tablica 4. Razine zrelosti 4 i 5 u svijetu 2004. godine<sup>21</sup>

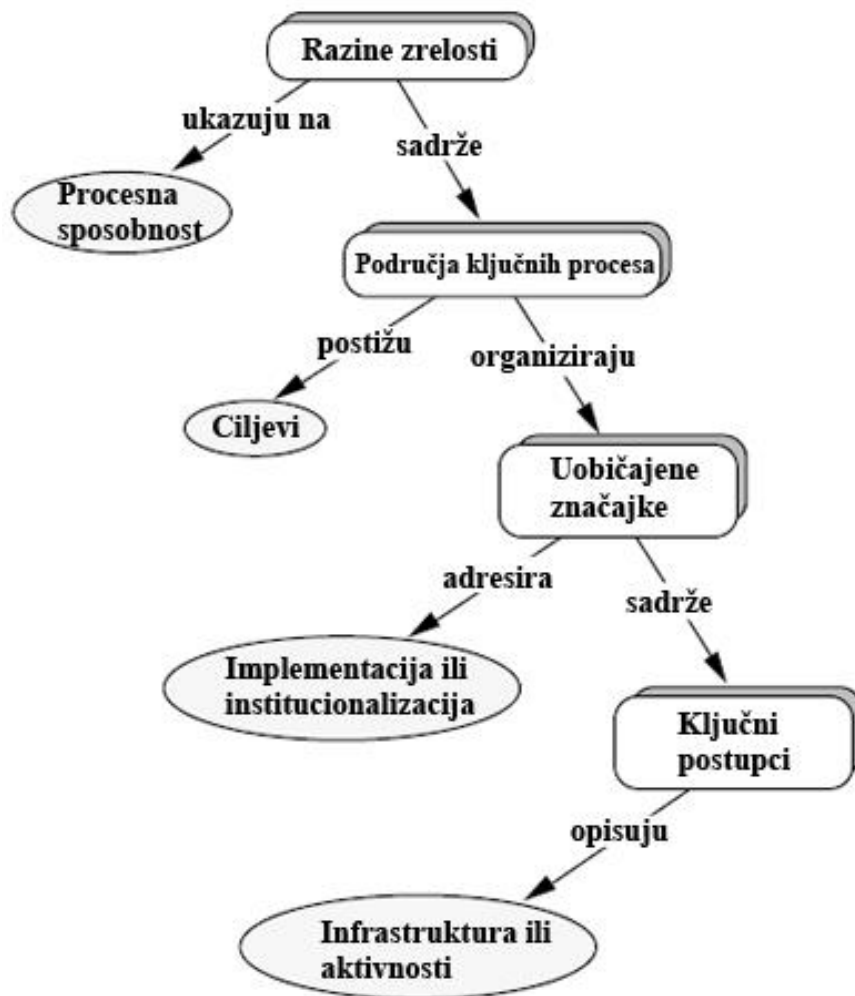
Tablica 4. Prikazuje broj poduzeća u državama navedenim u tablici koja su postigla upravljaju (razinu 4) i optimiziranu (razinu 5) razinu zrelosti CMM-a 2004. godine. Prema prikazanim podacima može se vidjeti da je Indija vodeća zemlja u ukupnoj količini poduzeća koja su postigla razine 4 i 5 sa 77 poduzeća dok je SAD drugi sa 59 poduzeća. SAD ima veći broj poduzeća od Indije u upravljanoj razini zrelosti sa 12 poduzeća više od Indije. Prema ovoj tablici može se jasno vidjeti da je Indija jaka zemlja u razvoju ICT-a te da poduzeća u njoj imaju visok stupanj kvalitete što se odražava na samu proizvodnju softvera. Za najnovije podatke o broju poduzeća koja koriste CMM i njihovim razinama zrelosti može se posjetiti internet stranica : <https://sas.cmmiinstitute.com/pars/pars.aspx>

### 3.3. Uobičajene značajke i ključni postupci/procedure CMM strukture

U prošlom poglavlju govorilo se o razinama zrelosti CMM-a te njihovim područjima ključnih procesa. U ovom poglavlju će se govoriti o uobičajenim značajkama i ključnim postupcima CMM strukture. CMM struktura se kako je prikazano na slici 6. sastoji od razina zrelosti, područja ključnih procesa te uobičajenih značajki i ključnih postupaka CMM-a.

<sup>20</sup> GRIGGS G. (2004.), *Quality Management of the Software Industry*; str.5

<sup>21</sup> GRIGGS G. (2004.), *Quality Management of the Software Industry*



Slika 6. CMM struktura<sup>22</sup>

Slika 6. prikazuje CMM strukturu u kojoj se nalaze razine zrelosti (eng.maturity levels), područja ključnih procesa (eng. key process areas), uobičajene značajke (eng. Common features) i ključni postupci (eng. Key practices). Na slici se može vidjeti da sve kreće od razina zrelosti tj. načina rada poduzeća gdje se definiraju područja ključnih procesa s ciljevima. Kako je navedeno u prošlom poglavlju kod razina zrelosti se prikazuje procesna sposobnost organizacije te kako oraganizacija reagira u određenim situacijama. Na slici se također vidi da područja ključnih procesa organiziraju uobičajene značajke koje se bave implementacijom ili instutucionalizacijom<sup>23</sup> područja ključnih procesa te na kraju uobičajene

<sup>22</sup> PAULK M. (1993.), Capability Maturity Model for Software, Version 1.1, Carnegie Mellon University, Pittsburgh, Pennsylvania

<sup>23</sup> proces nastajanja strukturiranih društvenih praksa koje služe održavanju društvenih vrijednosti i zadovoljavanju ljudskih potreba

značajke sadrže ključne postupke koji opisuju onu infrastrukturu i aktivnosti organizacije koje najviše doprinose implementaciji i institucionalizaciji područja ključnih procesa.

Uobičajene značajke CMM strukture su atributi koji prikazuju jesu li implementacija i institucionalizacija područja ključnih procesa efektivni, ponavljajući i trajni. Postoji pet uobičajenih značajki koje su:

1. **Posvećenost izvršenju posla** – opisuje koje radnje treba poduzeti organizacija da bi osigurala da uspostavljeni poslovi traju. Posvećenost izvršenju posla sadrži organizacijske politike i sponzorstvo višeg managementa.
2. **Sposobnost za obavljanje** – opisuje koji preduvjeti moraju postojati u projektu i organizaciji da bi se u potpunosti uspješno implementirao softverski proces. Ova uobičajena značajka uključuje resurse, organizacijske strukture i trening.
3. **Obavljene aktivnosti** – opisuje uloge i procedure potrebne za implementaciju područja ključnih procesa. To se uobičajeno odnosi na uspostavljanje planova i procedura, obavljanje i praćenje posla i poduzimanje korektivnih radnji po potrebi ako nešto ne valja kod uspostavljanja planova i procedura i obavljanja posla.
4. **Mjerenje i analiza** – opisuje potrebu za mjerenje procesa i analiziranje mjera. Mjerenje i analiza uključuju primjere mjera koje se moraju uzeti u obzir kod određivanje statusa i efektivnosti obavljenih aktivnosti.
5. **Potvrđivanje implementacije** – opisuje korake koji su potrebni za osiguravanje da se aktivnosti izvode po tome kako je postavljen proces. Potvrđivanje obuhvaća recenzije i audite od strane managementa te uz to i osiguranje kvalitete softvera.

Svaka područja ključnih procesa opisana su kao nekakvi ključni postupci koji doprinose zadovoljavanju ciljeva područja ključnih procesa. Ključni postupci opisuju infrastrukturu i aktivnosti koje najviše doprinose efektivnoj implementaciji i institucionalizaciji područja ključnih procesa. Svaki se ključni postupak sastoji od jedne rečenice koju sljedi detaljan opis te koji sadrži primjere i obradu samog postupka. Ključni postupci se također nazivaju i postupci najviše razine koji navode temeljne politike, procedure i aktivnosti za područja ključnih procesa. Dijelovi detaljnog opisa ključnog postupka često se nazivaju kao podpostupci. Važno je naglasiti da ključni postupci opisuju što se treba napraviti a ne kako ostvariti željene ciljeve tj. ključni postupci opisuju sve ono što mora biti izvršeno, no ne ističu

na koji način bi koji proces trebao biti implementiran. Ključni postupci trebaju se interpretirati racionalno radi toga da bi se moglo zaključiti mogu li se ciljevi područja ključnih procesa drugačije postići. Svrha ključnih postupaka je davanje uvida u ključne elemente uspješnog procesa. Također svaki ključni postupak u svakom području ključnih procesa opisan je pomoću skupa zajedničkih obilježja.

### **3.4.Korištenje CMM-a**

Iz prijašnjih poglavlja može se zaključiti da CMM uspostavlja kriterije za opisivanje karakteristika zrelih softverskih organizacija. Ti se kriteriji tako koriste od strane organizacija kod unaprijeđivanja njihovih procesa za razvoj i održavanje softvera. Također se mogu koristiti od strane vlade ili komercijalnih organizacija za procjenu rizika prije nego što budu krenuli raditi na softverskom projektu sa određenim poduzećima.

Kod korištenja CMM-a postoje dvije metode razvijene od strane SEI-a za ocjenjivanje zrelosti softverskog procesa organizacije. Te dvije metode su procjena softverskog projekta (eng. Software process assessment) i procjena sposobnosti softvera (eng. Software capability evaluation). Procjena softverskog projekta se koristi kod utvrđivanja trenutnog stanja softverskog procesa organizacije da bi se pronašli i utvrdili kritični i visoko prioritetni problemi i greške s kojima se suočava organizacija. Također se procjena softverskog projekta koristi da se dobije organizacijska podrška za unaprijeđenje softverskog procesa. Procjena sposobnosti softvera se koristi za identificiranje onih ljudi koji su kvalificirani za rad na softveru te prati stanje softverskog procesa. CMM je uobičajen temelj za procjenu softverskog projekta i procjenu sposobnosti softvera dok se njihove svrhe uveliko razlikuju.

Uz različite svrhe procjena softverskog projekta i procjena sposobnosti softvera imaju par sličnosti koje su te:

- Da koriste upitnik za zrelost svaki put kad se metode krenu koristiti.
- Da koriste CMM kao mapu koja pomaže voditi procjene na licu mjesta.
- Razvijaju rezultate koji identificiraju snage i slabosti softverskog procesa s obzirom na područja ključnih procesa u CMM-u.

- Dobivaju okvir zasnovan na analizi zadovoljstva s ciljevima unutar područja ključnih procesa
- Prikazuju svoje rezultate prikladnoj publici s obzirom na rezultate istraživanja i okvir područja ključnih procesa.

Uz sličnosti koje su navedene procjena softverskog projekta i procjena sposobnosti softvera imaju i svoje razlike. Jedan od razloga njihovih različitosti je opseg same procjene softverskog projekta i procjene sposobnosti softvera. Procjene softverskog procesa se izvode u otvorenom, suradničkom okruženju a procjena sposobnosti softvera se izvodi u audit orijentiranom okruženju. Uspjeh procjene softverskog procesa ovisi o odlučnosti managementa i profesionalnog osoblja za unaprijeđenjem organizacije. Cilj je naći probleme i pomoći managerima i inženjerima unaprijediti organizaciju u kojoj rade. Dok je cilj procjene sposobnosti softvera vezan za novčane okolnosti jer preporuke tima za procjenu pomažu odabrati izvođače ili postaviti naknade za nagrade za uspješnu procjenu sposobnosti softvera. Kod procjene softverskog procesa identificiraju se problemi softverskog procesa s kojima se suočava organizacija, organizacijski fokus na proces te motivaciju i entuzijazam kod izvođenja akcijskog plana. To su ovo najvažniji i najpoželjniji rezultati kod same procjene softverskog procesa.

### **3.5.CMMI (Capability Maturity Model Integration)**

Povijest CMMI-a počinje razvojem CMM-a (CMM v1.0) koji je prvi put objavljen 1990. Nakon što su ga mnoga američka i svjetska poduzeća usvojila i počela uspješno primjenjivati u mnogim domenama, počele su se razvijati druge CMM inačice za druge discipline i funkcije, kao što su inženjerstvo sustava, ljudi, integrirani proizvod razvoja, nabavka softvera itd. Korištenjem CMM-a mnoge su organizacije ostvarivale korist. Zajedničko im je bila borba s problemima zbog preklapanja, nedosljednosti i integracije. CMMI je zamišljen kao inicijativa za integriranje raznih CMM-ovih modela te se je razvio kao nasljednik CMM-a. Prema SEI-u CMMI pomaže „integrirati odvojene organizacijske funkcije, namjestiti ciljeve za unaprijeđenje procesa i prioriteta, pružiti smjernice za kvalitetnije procese i pružiti referentnu točku za ocjenjivanje trenutnih procesa“. CMMI se razvija i danas te je prošao kroz tri verzije. CMMI verzija 1.1 je izašla 2002. godine, verzija 1.2 je izašla u Kolovozu 2006. godine, a CMMI verzija 1.3 u Studenom 2010. Godine. CMMI

kao i CMM ima pet razina zrelosti. Za razliku od CMM-a CMMI sadrži dodatno procesno područje, nazvano "Mjerenje i analiza" (eng. Measurement and Analysis) koje je definirano za drugu razinu zrelosti. Osnovni smisao definicije dodatnog procesnog područja je definiranje i uspostava mehanizma upravljanja informacijama koje su neophodne za praćenje i nadzor svakog definiranog procesnog područja na jedinstven način.

CMMI model nije proces nego opisuje karakteristike efektivnih procesa. Koncept ovog modela ujedinjuje sistemsko i softversko inženjerstvo, razvoj proizvoda i usluga, te poslovnih procesa i podršku korisnicima. Na osnovama tradicionalnog CMM koncepta procesnog i projektnog managementa ovaj je model razvijen kako bi se koristio u interakciji s drugim CMM modelima, fokusirajući se na ostvarenje zadatka određene razine zrelosti pri razvijanju ili vrednovanju određenih procesa.

Kod CMMI-a postoji način na koji se CMMI može koristiti u organizaciji te se on dijeli na dva potencijalna pravca:

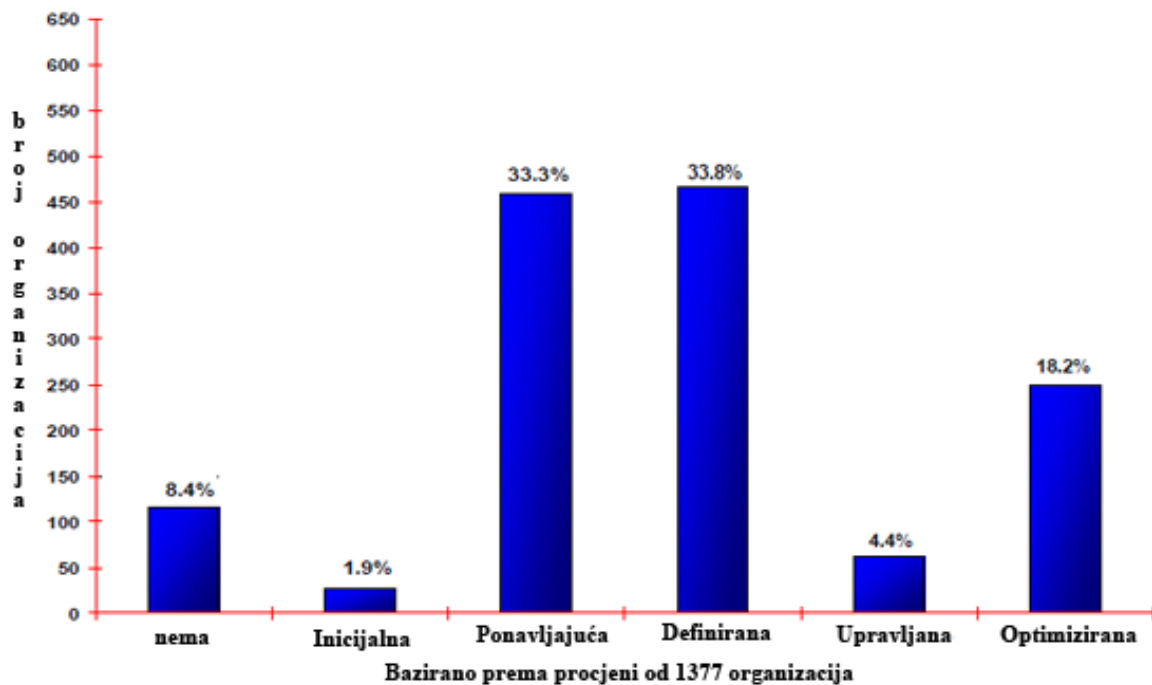
1. **Kontinuirani pristup** (eng. *Continuous Representation*) – Procesna područja se biraju na osnovu prioriteta. Osnovni cilj je unaprijediti samo procesno područje do određenog nivoa zrelosti, u skladu s drugim procesnim područjima s kojima je usko povezana.
2. **Pristup po nivoima** (eng. *Staged Representation*) – Procesna područja se biraju na osnovu nivoa zrelosti na kome se organizacija nalazi, pa se samim tim pažnja usmjerava isključivo na procesna područja u okviru konkretnog nivoa zrelosti.

Kako je navedeno u prijašnjem tekstu da zbog pojave višestrukih modela organizacijske zrelosti softverskih procesa nastaje CMMI (CMM Integration). CMMI okvir (eng. Framework) dizajniran je s posebnim ciljem da integrira aktivnosti poboljšanja procesa razvoja iz različitih disciplina u organizaciji. CMMI za razliku od CMM modela unosi dodatnu zasebnu disciplinu o mjerenjima koja je temeljena na SPICE normi. SPICE (eng. Software Process Improvement and Capability dEtermination; ISO/IEC 15504) norma je referentni model modela zrelosti (koji se sastoji od svojstva procesa i generičnih postupaka) preko kojih procjenitelji mogu postaviti dokazne materijale njihove procjene tj. da procjenitelji mogu dati svoju potupunu procjenu organizacijskih sposobnosti za isporuku proizvoda (softvera itd.). O SPICE-u više u sljedećem poglavlju. Zanimljivo je da u zadnjem



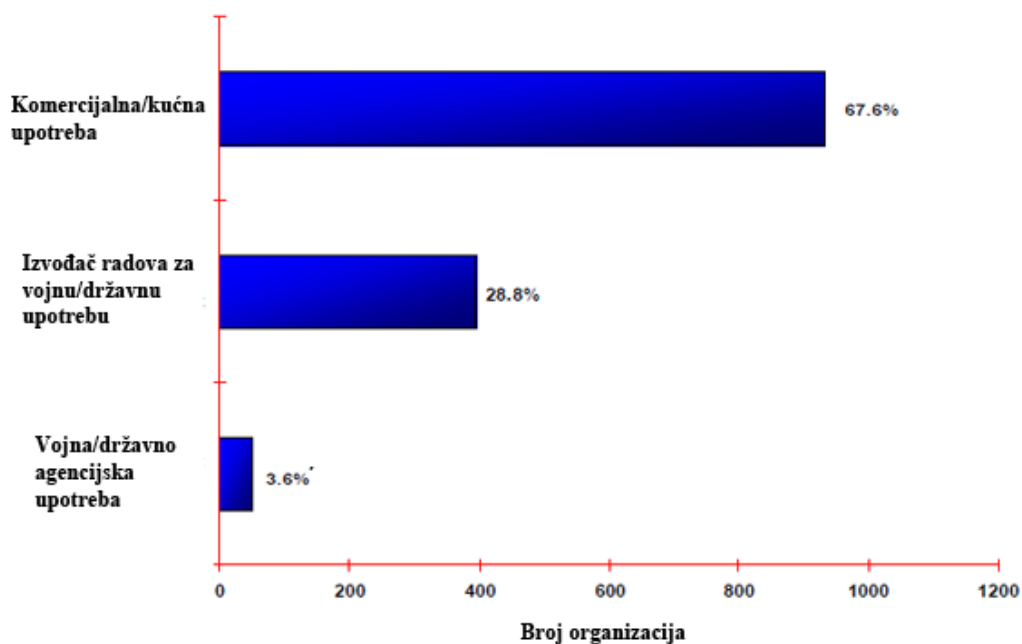
desetljeću CMM i CMMI modeli postaju popularniji od međunarodno priznatih ISO normi a razlog toga je što su uglavnom besplatni i jednostavno dostupni svima za razliku od ISO normi. Kod trenutne verzije 1.3 CMMI-a postoje tri Vrste CMMI modela:

1. **CMMI for Development (CMMI-DEV)** – Koji predstavlja razvoj proizvoda i usluga.
2. **CMMI for Services (CMMI-SVC)** – Koji predstavlja uspostavljanje, upravljanje i pružanje usluga.
3. **CMMI for Acquisition (CMMI-ACQ)** – Koji predstavlja objedinjavanje proizvoda i usluga.



Graf 1. CMMI Organizacije prema njihovim razina zrelosti (Izvor: <http://resources.sei.cmu.edu/library/>; datum pristupa 15.06.2015.)

Graf 1. Prikazuje broj organizacija koje imaju CMMI model na datum 30. Lipnja 2006. godine. Tada je broj organizacija koje koriste CMMI bio 1377. Na grafu se može jasno vidjeti da je tada bilo najviše organizacija koje se nalaze na Ponavljajućoj i Definiranoj razini zrelosti a najmanje na prvoj tj. inicijalnoj razini. Zanimljivo je istaknuti da je od 1377 organizacija njih 8.4% tada još bilo bez definirane razine zrelosti.



Graf 2. Kategorije CMMI organizacija (Izvor: <http://resources.sei.cmu.edu/library/>; datum pristupa 15.06.2015.)

Graf 2. Prikazuje kategorije CMMI organizacija tj. u kojim se svim područjima koristi CMMI. Tako se može vidjeti da u komercijalnoj/kućnoj upotrebi postoji najviše organizacija koje koriste CMMI sa 67.6%. Nakon toga su izvođači radova za vojnu/državnu upotrebu kojih ima 28.8% te najmanje vojnih/državno agencijskih organizacija koje koriste CMMI sa 3.6% od ukupno 1377 organizacija na dan 30. Lipnja 2006. Godine.

### 3.5.1. CMMI okvir (eng. Framework)

CMMI okvir je kolekcija komponenti koji se koriste za izgradnju CMMI modela, CMMI materijala za trening i CMMI materijala za procjenu. Komponente za izgradnju CMMI modela sadrže područja procesa, ciljeve, procedure i informativni materijal za korištenje modela i njegovih komponenti. Materijali za trening sadrže vodiče za implementaciju modela i video-e za učenje kako koristiti model. Materijali za procjenu opisuju proces procjenjivanja organizacijskih procesa prema ciljevima i procedurama modela. Oni također uključuju trening za provođenje procesa procjenjivanja.

Svrha okvira CMMI arhitekture je kontrola odabiranja i korištenja komponenti modela za izgradnju CMMI modela za razna područja koja zanimaju organizaciju. Kada se gradi novi

CMMI model, programeri koriste postojeće isprobane komponente koje odgovoraju potrebama novih područja interesa. Korištenjem tih postojećih komponenti smanjuje se vrijeme potrebno za trening i ako je potrebno prilagođava se postojećim procesima. CMMI okvir također omogućava modelima koji su novi da zadrže terminologiju i strukturu od drugih CMMI modela u okviru tako da učenje novog modela bude bazirano na poznatim konceptima modela.

Kako je navedeno u prošlom tekstu CMMI okvir je kolekcija svih komponenti modela, materijala za trening i komponenti za procjenjivanje. Te navedene komponente su organizirane u grupe koje se zovu konstelacije, koje olakšavaju izgradnju odobrenih modela i čuvaju nasljeđe postojećih CMM i CMMI modela. U CMMI okviru postoje također i konstelacije komponenti koje se koriste za izgradnju modela, trening i materijala za procjenu u područjima interesa (npr. nabava, razvoj, usluge itd.). Tako se može reći da konstelacije nabave podržavaju organizaciju ili projekt u nabavi proizvoda ili usluga od dobavljača van organizacije ili projekta. Konstelacije za razvoj podržavaju organizaciju ili projekt koji razvija proizvode ili usluge, dok konstelacija usluga podržava organizaciju ili projekt koji isporučuje usluge. U sljedećim poglavljima će biti malo govora o konstelacijama za razvoj, usluge i nabavu.

### 3.5.2. *CMMI za Razvoj (CMMI-DEV)*

CMMI za razvoj je referentni model koji pokriva aktivnosti za razvoj proizvoda i usluga. Organizacije koje posluju u različitim industrijama, uključujući zrakoplovstvo, bankarstvo, računalni hardver i softver, proizvodnja automobila i telekomunikacije, sve one mogu koristiti CMMI za Razvoj. CMMI za Razvoj sadrži procedure koje obuhvaćaju projektni management, procesni management, sistemsko inženjerstvo, hardversko inženjerstvo, softversko inženjerstvo i druge pomoćne procese koji se koriste u razvoju i održavanju.

CMMI za Razvoj pomaže organizacijama i managerima da koristeći svoju profesionalnu prosudbu i zdrav razum intepretiraju model za svoje potrebe i potrebe organizacije. CMMI za Razvoj priručnik navodi da iako procesna područja u CMMI-DEV modelu opisuju korištenje najboljih procedura za većinu korisnika, procesna područja i procedure se trebaju intepretirati

korištenjem opširnog znanja o CMMI-DEV, određenih organizacijskih ograničenja i poslovnog okruženja same organizacije koja ga koristi.

### 3.5.3. CMMI za Usluge (CMMI-SVC)

CMMI-SVC se oslanja na koncepte i procedure od CMMI i drugih standarda i modela koji se fokusiraju na usluge. CMMI-SVC priručnik navodi da poznavanje drugih modela i standarda može pružiti bolje razumjevanje CMMI-SVC-a ali ih nije nužno poznavati. CMMI-SVC model obuhvaća potrebne aktivnosti za uspostavljanje, pružanje i održavanje usluga. Prema CMMI-u usluga je proizvod koji je neopipljiv i koji se nemože pohraniti. CMMI-SVC model je razvijen da se prilagođava toj definiciji usluge. Ciljevi i procedure CMMI-SVC-a su važne samo onim organizacijama koje se bave pružanjem usluga. Vrijedi napomenuti da je CMMI-SVC sličan drugim CMMI modelima u ostalim konstelacijama tako da oni koji su upoznati s drugim CMMI modelima neće imati problema s CMMI-SVC. Pri korištenju ovog modela za organizaciju je također pametno koristiti profesionalnu prosudbu i zdrav razum za njegovu intepretaciju. Unatoč sličnostima za razliku od CMMI-DEV, CMMI-SCV pruža alternativni pojednostavljen pristup za procjenu i unaprijeđenje razvoja uslužnih sistema.

### 3.5.4. CMMI za Nabavu (CMMI-ACQ)

Prije nastanka CMMI-ACQ postojala je mala uvodna kolekcija za nabavu koja je u sebi sadržavala nabolje procedure za nabavu a zvala se Nabavni modul (eng. Acquisition Module (CMMI-AM)). On je bio baziran na CMMI okviru (Framework). Uz to što je opisivao najbolje procedure modul nije bio namjenjen da bude model za procjenjivanje ili model za unaprijeđenje procesa. Danas postoji noviji CMMI dokument za nabavu zvan CMMI Acquisition Primer te ga se ne smije zamjenjivati sa CMM-ACQ. Prvi koraci koji su doveli do nastanka CMMI-ACQ su počeli kada su se udružili SEI i General Motors u cilju da naprave prvi inicijalni model za nabavu koji je bio skica današnjeg modela. Danas CMMI-ACQ predstavlja rad mnogih organizacija i individualaca iz različitih industrija, vlada i SEI-a. Današnja verzija CMMI-ACQ je zajednički model različitih organizacija koji služi za pružanje uvida kako nabavljati materijale za razvoj proizvoda.

Loše upravljanje, nemogućnost za shvaćanjem korisničkih potreba, loš dobavljač i ugovorni procesi, nekontrolirani zahtjevi za promjene su jedni od faktora koji dovode do propasti projekta. Odgovornost je jednako raspoređena na dobavljača i primatelja. Većina propalih projekata mogu se izbjeći tako da se primatelj pripremi i pravilno nauči kako upravljati dobavljačima.

Zanimljiva informacija vezana za CMMI za Razvoj, Usluge i Nabavu nalazi se u CMMI verziji 1.2 te najnovijoj 1.3. Pregledom u oba dvije verzije CMMI-a može se vidjeti da CMMI za Razvoj, Usluge i Nabavu imaju 16 zajedničkih procesnih područja koja su:

1. Organizational Process Focus (*eng. Fokusiranje organizacijskih procesa*)
2. Organizational Process Definition +IPPD (*eng. Definiranje organizacijskih procesa + IPPD*)
3. Organizational Training (*eng. Organizacijski trening*)
4. Organizational Process Performance (*eng. Organizacija učinkovitosti procesa*)
5. Organizational Innovation and Deployment (*eng. Organizacija inovacija i razvoja*)
6. Project Planning (*eng. Planiranje projekta*)
7. Project Monitoring and Control (*eng. Planiranje nadzora i praćenja*)
8. Integrated Project Management +IPPD (*eng. Integrirano upravljanje projektima*)
9. Risk Management (*eng. Upravljanje rizicima*)
10. Quantitative Project Management (*eng. Upravljanje kvantitativnim projektima*)
11. Requirements Management (*eng. Upravljanje zahtjevima*)
12. Configuration Management (*eng. Upravljanje konfiguracijom*)
13. Process and Product Quality Assurance (*eng. Procesi i Osiguranje kvalitete proizvoda*)
14. Measurement and Analysis (*eng. Mjerenje i analiza*)
15. Decision Analysis and Resolution (*eng. Odluka analize i rezolucije*)
16. Causal Analysis and Resolution (*eng. Kauzalna analiza i rezolucija*)

### **3.6.Ostali CMM modeli**

#### *3.6.1. P-CMM – People Capability Maturity Model*

Osim CMMI modela postoje i mnoge druge verzije CMM prilagođene za određene situacije i radnje. Tako je nastao i P-CMM koji predstavlja model primijenjen procesima vezanim uz ljudske potencijale. Ovaj se model bavi ljudskim potencijalima, managementom znanja i poslovnim organizacijama te mu je njegov cilj razvoj i upravljanje ljudskim potencijalima. Kao ostali CMM modeli ovaj se model može koristiti skupa s drugim modelima čime se integrira razvoj ljudskih potencijala i poslovnih procesa. Glavni cilj P-CMM-a je usavršavanje sustava ljudskih potencijala kroz kontinuirano razvijanje individualnih kompetencija, razvoj timova i odjela, te uspostavljanje korporativne kulture.

#### *3.6.2. SSE-CMM – Model sigurnosti (eng.System Security Engineering Capability Maturity Model)*

Među raznim ostalim CMM modelima postoji i SSE-CMM model koji se bavi sigurnošću softvera i IT-a općenito te kao ostali modeli nastao je na klasičnom CMM modelu. Ovaj model specifičan je po tome da je u svojoj biti dvodimenzionalan. Prva dimenzija sastoji se od osnovne prakse koja definira inženjering sigurnosti a druga dimenzija odnosi se na generičku praksu koja ukazuje na razinu upravljanja procesima sigurnosti. Korištenjem ovog modela korisnici primjećuju razne probleme sigurnosti i zaštite u razvoju i primjeni informacijsko - komunikacijskih tehnologija, od sigurnosti aplikacija, sistemskih datoteka, sigurnosti u razvoju, pa do sigurnosti osoblja i same organizacije. Ovaj model daje smjernice i okvir za pravilno prepoznavanje svih aspekata sigurnosti i zaštite, kao i za njihovu implementaciju u praksi.

### 3.7.CMM certificiranje

Da bi se potvrdilo da su softverski proizvodi koje određeno poduzeće proizvodi kvalitetni i u zakonskim granicama te ne škode korisnicima proizvoda provodi se certifikacija. Certifikacija je postupak u kojem neovisna organizacija na temelju provedenog ocjenjivanja sukladnosti, utvrđuje zadovoljava li proizvod, proces, sustav upravljanja ili osoba kriterije sadržane u određenom normativnom dokumentu. Taj normativni dokument može biti međunarodna ili nacionalna norma, specifikacija, zakonski akt (pravilnik ili sl.) ili vlastito razvijena certifikacijska shema nekog certifikacijskog tijela. U slučaju ispunjenja postavljenih kriterija definiranih spomenutim normativnim dokumentom, certifikacijsko tijelo izdaje certifikat.

Važno je napomenuti da su rezultati procjene kvalitete (zrelosti) bilo kojeg proizvoda ili procesa nastajanja proizvoda (naravno i softvera) koji rezultiraju nekim certifikatom ili potvrdom iznimno važni i za kupca (korisnika) i za proizvođača, u ovom slučaju softverskog proizvoda. Kada se kupac odnosno korisnik odluči za kupnju tog certificiranog proizvoda, on kroz njegovu nabavu nastoji osigurati da nabavlja proizvod odgovarajuće razine zrelosti. Za razliku od kupca tog proizvoda certificiranje proizvođaču pomaže podizati kvalitetu procesa izrade softverskog proizvoda te tako samog proizvoda. Također mu certifikat osigurava određenu povoljniju poziciju na tržištu.

Godina	Procjene	Ukupne procjene	Organizacije	Postotak certifikata van SAD-a
87-93	338	338		16.0%
1994	130	468		18.2%
1995	123	591		20.7%
1996	182	773	616	20.5%
1997	220	993	782	24.1%
1998	245	1383	951	26.8%
1999	299	1537	1166	29.9%
2000	321	1858	1380	32.7%
2001	335	2193	1638	38.0%
2002	453	2646	1978	47.1%
2003	511	3157	2401	54.7%

Tablica 5. Broj certificiranih organizacija<sup>24</sup>

<sup>24</sup> GRIGGS G. (2004.), Quality Management of the Software Industry

Tablica 5. Prikazuje od godine 1987. do 2003. procjene tj. broj danih CMM certifikata od strane SEI-a. Ukupne procjene predstavljaju ukupan zbroj procjena prošlog i trenutnog vremenskog razdoblja. Na tablici se može vidjeti broj organizacija koje pripadaju pod razine 1-3 CMM-a, koje su se počele bilježiti od 1996. godine. Postotak certifikata van SAD-a prikazuje koliko je dano CMM certifikata van SAD-a tijekom svakog vremenskog perioda od broja procjena svake godine. Nakon 2003. godine može se vidjeti da je broj procjena tj. danih CMM certifikata van SAD-a prešao 50%.

Kod CMMI-a izraz certificiran (eng. certified) je zamjenjen sa izrazom procijenjen (eng. appraised). Ta procjena organizacije se se vrši na osnovu upoređivanja postignutog sa zahtjevima definiranim u dokumentu "Appraisal Requirements for CMMI (ARC)". Prema tom dokumentu definiraju se tri klase procjene koje su podjeljene u A, B i C metode:

1. **A metoda** – Predstavlja SCAMPI (Standard CMMI Appraisal Method for Process Improvement). Nju provode SEI-autorizirani vodeći procjenjivači u skladu sa SCAMPI A Method Definition Document-om (MDD). CMMI verzija 1.3. obuhvaća metodu SCAMPI. SCAMPI služi za procjenu kvalitete te se koristi za identificiranje i procjenjivanje snaga i slabosti postojećih procesa, otkrivanje rizika te utvrđivanje razine sposobnosti i zrelosti procesa.
2. **B metoda** – Predstavlja metodu koju provodi sama organizacija u cilju procjene nivoa zrelosti, ali sa značajno manjim troškovima. Kod ove metode organizacija sama dobiva detaljan uvid u snage i slabe točke svojih procesa i ima visoku svijest o tome na što treba obratiti pažnju u daljem razvoju.
3. **C metoda** – Predstavlja metodu koja je najmanje formalna od gore navedenih metoda. Ovom metodom organizacija dobiva u najkraćem roku izvještaj koliki je njen rejting po određenom nivou zrelosti i koji pokazuje koliko je organizacija spremna za formalnu SCAMPI A metodu procjene.

Zanimljivo je istaknuti da za procjenu razine zrelosti prema CMMI metodologiji se ne dodjeljuje certifikat kao kod ocjenjivanja po ISO normama, ali se dodjeljuje potvrda o postignutoj razini (1-5) koja ima vrijednost sličnu certifikatu i može se prikazivati kao oznaka postignute zrelosti.



Ako su poduzeća zainteresirana za CMM certifikat prvo se moraju informirati na stranicama SEI-a. Nakon što dobiju osnovne informacije kako krenuti u proces dobivanja CMM certifikata poduzeća se susreću s dva izbora.

Prvi izbor je kupovanje „CMM-a u kutiji“ i zapošljavanjem konzultanta. Kupovanje „CMM-a u kutiji“ tj. kupnja CMM predložaka je skupo i ne daje sa sigurnošću poduzećima certifikaciju. Tako da je najbolje i najsugrnije zaposliti konzultanta što je osobito važno za mala i srednja poduzeća. CMM konzultanti imaju zadatak da svu svoju pažnju fokusiraju na poduzeće. Također većina konzultanata su procjenitelji koji imaju dozvolu od SEI-a da se bave CMM konzultiranjem. Oni na kraju osiguravaju poduzećima da dođu do testiranja za CMM certifikat. Poduzećima je potrebno pola godine do tri godine maksimalono da se pripreme za CMM test koji se provodi pet do sedam dana od strane procjeniteljskog tima.

Drugi izbor je biranje između certifikacije područja procesa ili cjelokupne certifikacije. Svaka razina CMM certifikacije je podjeljena na povezana procesna područja npr. na definiranoj razini procesna područja su podjeljena na kontrolu, nabavu, osiguranje softverskog proizvoda itd. Cerifikacija procesnih produčja dopuštaju poduzećima procjenu određenih područja te fokusiranje napora na ta područja. Certifikacija područja procesa nije preporučljiva jer šteti sveukupnom cilju CMM-a koji je sveukupno unaprijeđivanje organizacije.

Accenture	Bank of America	BMW
Boeing	Bosch	Ericsson
Dyncorp	EDS	Fujitsu
FAA	Fannie Mae	Hitachi
General Dynamics	General Motors	Infosys
Honeywell	IBM Global Services	KPMG
Intel	J. P. Morgan	Motorola
L3 Communications	Lockheed Martin	NEC
NASA	NDIA	NRO
Nokia	Northrop Grumman	NTT DATA
Polaris	Raytheon	Reuters
SAIC	Samsung	Social Security Administration
Tata Consultancy Services	TRW	U.S. Air Force
U.S. Army	U.S. Navy	U.S. Treasury Department
Wipro	Zurich Financial Services	

Slika 7. Poznata poduzeća certificirana prema CMM-u

Slika 7. prikazuje neka poznata poduzeća certificirana prema CMM-u 2007. Godine objavljeno od strane SEI-a. Tu se mogu vidjeti razna poduzeća od proizvođača automobila BMW-a, preko proizvođača mobitela Nokie, Motorole proizvođača računalne opreme i softvera poput Intela i Hitachia te poduzeća i dijelova američkog sustava poput NASA-e, američke vojske, mornarice, zrakoplovstva i riznice. Na ovoj se stranici može vidjeti ukupan broj certificiranih poduzeća svake godine <https://sas.cmmiinstitute.com/pars/pars.aspx>.

### *3.7.1. Ustanove za certifikaciju*

Postoje različite vrste normi odnosno standarda (standard je viši oblik norme) koje donose različite međunarodne, regionalne, državne i druge organizacije i udruženja koja se bave normizacijom odnosno standardizacijom. Najvažnije ustanove koje se bave standardizacijom odnosno certificiranjem:

#### **1. Hrvatski zavod za norme**

Hrvatski zavod za norme (Zagreb, Ulica grada Vukovara 78.) obavlja upravne i druge stručne poslove koji se odnose na normizaciju, ovlašćivanje (akreditaciju), potvrđivanje (certificiranje), mjeriteljstvo i nadzor, izdavanje hrvatskih normi, uspostavljanje i vođenje središnjeg dokumentacijskog i informacijskog sustava o hrvatskim normama, uspostavljanje, opsluživanje i nadzor u djelatnostima normizacije i mjeriteljstva. Hrvatski zavod za norme član je niza međunarodnih organizacija za normizaciju i standardizaciju.

#### **2. Software Engineering Institute (SEI)**

Objašnjen u prijašnjim proglaavljenjima, institut vezan za razvoj CMM-a. Zadatak instituta je prvobitno bio stjecanje iskustva u softverskom inženjerstvu zbog važnosti kvalitete softvera unutar sustava obrane SAD-a (institut je sponzoriran od strane ministarstva obrane SAD-a). Osnovna područja djelovanja instituta su tehnički i managerski aspekti softverskog inženjerstva.

#### **3. European Software Institute (ESI)**

Unutar ovog instituta nalazi se Centar za poboljšanje kvalitete softvera osnovan 1993. godine. Institut je neprofitna organizacija sa sjedištem u gradu Zamudio (Španjolska). Usko je povezan s industrijom softvera, a zadatak mu je doprinijeti razvoju konkurentnosti u europskoj industriji softvera kroz promocije i konstantno poboljšanje informacijskih i komunikacijskih tehnologija.

#### **4. SPICE – Software Process Improvement and Capability dEtermination**

Organizacija istog imena osnovana 1993. godine sa zadatkom pružanja pomoći u razvijanju međunarodnog standarda za procjenu softverskih procesa. 1995. godine napravljena je skica standarda za procjenu softverskog procesa zvan SPICE a razvijena je kao odgovor na CMM. Ta skica je rezultirala razvojem ISO/IEC TR 15504:1998 - Software Process Assessment. SPICE je nastojalo uzeti najbolja iskustva iz raznih standarda i metoda ocjenjivanja PIS-a poput normi ISO 9001 i 90003 iz kojih su preuzeti koncepti sustava upravljanja kvalitetom i principi za definiranje i klasifikaciju softverskih procesa iz ISO 12207. Također su preuzeti i principi poboljšanja upravljanja kvalitete iz ISO 9004-45 te modeli ocjenjivanja sposobnosti softvera iz CMM-a.

## **4. OSTALI MODELI ZA OSIGURANJE KVALITETE SOFTVERA**

U ovom poglavlju biti će govora o ISO normama za osiguranje softvera poput ISO 9001, ISO/IEC 90003 te Bootstrap metodi. ISO je međunarodna nevladina organizacija osnovana 1947. godine, te se sastoji od tijela zaduženih za međunarodne norme u obliku federacije a trenutno se sastoji od preko 130 zemalja članica dok je Bootstrap institut koji se sastoji od mreže europskih organizacija koje surađuju s ciljem razmjene iskustava u svrhu poboljšanja kvalitete softvera.

Misija ISO organizacije je unaprijediti razvoj normi i njima srodnih aktivnosti u svijetu sa ciljem olakšavanja svjetske razmjene dobara i usluga, kao i razvoja suradnje u području intelektualnih, znanstvenih, tehnoloških i ekonomskih aktivnosti. Dok je cilj Bootstrap metode procijeniti softverski proces u SPU (software production unit; hr. softverska proizvodna jedinica (ona se procjenjuje kod organizacije koja se bavi proizvodnjom softvera)) koja se nalazi u europskom tržištu softvera. Opis ISO modela koji se bave kvalitetom softvera, osiguranjem njegove kvalitete i Bootstrap metode biti će obuhvaćen u sljedećim poglavljima.

### **4.1.ISO 9001**

Kada se spominje kvaliteta i osiguranje proizvoda i usluga ono što većini ljudi prvo padne na pamet su ISO norme. ISO 9001 je jedna od najraširenijih međunarodnih normi koja postavlja zahtjeve za uspostavu i održavanje sustava upravljanja kvalitetom. Razlog njezine raširenosti je taj da se ona može primjenjivati na organizacije svih vrsta npr. profitne, neprofitne, proizvodne, uslužne, male, srednje, velike. Trenutni ISO 9001 standard koji sadrži zahtjeve za sustav upravljanja kvalitetom je ISO 9001:2008 a u iščekivanju je ISO 9001:2015 norma. Ovaj se standard ne odnosi toliko na softverske proizvode nego za procese proizvodnje svih vrsta proizvoda i usluga. Dokumenti i aktivnosti koje organizacija provodi u skladu s normom zajednički se nazivaju sustav upravljanja kvalitetom. Sustav upravljanja kvalitetom obuhvaća razne aktivnosti unutar organizacije koje su: planiranje i održavanje samog sustava, upravljanje resursima (ljudskim i infrastrukturom), planiranje, ugovaranje, prodaja, projektiranje i razvoj, nabava, proizvodnja i pružanje usluga mjerenja te analiza i poboljšanje procesa i sustava. ISO 9001 sadrži i načela koja su:

1. **Usmjerenost na kupca.** Da bi se uspješno unaprijeđivala kvaliteta proizvoda potrebno je razumjeti i zadovoljiti potrebe kupaca te nastojati nadmašiti njihova očekivanja.
2. **Vodstvo mora biti u skladu sa svrhom postojanja organizacije.** To znači da se treba stvoriti okruženje u kojem organizacija može ostvariti svoje ciljeve.
3. **Uključivanje ljudi.** Ovo načelo govori o tome da svi ljudi u organizaciji moraju razumjeti što se treba raditi i zašto te najvažnije znati kako.
4. **Procesni pristup.** Razumjeti slijed radnji i potrebne resurse.
5. **Sustavni pristup.** Ovo načelo govori o tome da se razumijevanjem međuovisnosti procesa postižu uspješnost i učinkovitost organizacije.
6. **Stalno poboljšavanje.** Neprestano učenje ljudi unutar organizacije čini organizaciju iskusnijom i spremnom za nove izazove i zahtjeve kupaca.
7. **Činjenični pristup odlučivanju.** Kod ovog načela učinkovite odluke temelje se na analizi podataka i informacija.
8. **Partnerski odnos s dobavljačima.** Uspješno poslovanje s dobavljačima osigurava kvalitetu proizvoda i njegovih dijelova.

Sustav upravljanja kvalitetom prema zahtjevima norme ISO 9001 danas se koristi u cijelom svijetu. Dok je certifikacija tog sustava najbolji način dokazivanja sadašnjem i potencijalnom partneru da će proizvod ili usluga zadovoljiti njegove zahtjeve prema kvaliteti. Također dobro razvijen i održavan sustav upravljanja kvalitetom ima pozitivan doprinos na ostvarenje ciljeva poslovanja poduzeća poboljšavajući zadovoljstvo i povjerenje kupaca te bolju kvalitetu proizvoda u što se ubraja bolja radna učinkovitosti, manje grešaka i troškova itd.

#### *4.1.1. TickIT*

TickIT postoji od ranih 1990-tih godina kada je nastao kao projekt poticanja dobrog IT inženjeringa, auditiranja i certifikacijskih postupaka, dok je sada zastario i zamjenjen sa TickIT plus-om 30. Studenog 2014 godine. TickIT je nastao od strane British Computer Society (BSC), UKAS (United Kingdom Accreditation Service), SWEDAC (Swedish Board for Accreditation and Conformity Assessment), i još 12 organizacija koje su ovlaštene za njegovo izdavanje. TickIT je prvenstveno bio zamišljen za tržište Velike Britanije, ali tijekom godina proširio se na druge države. Od svog izlaska TickIT se oslanja na normu ISO 9001.

TickIT je primarno izašao da se bavi problemima unutar područja klasičnog razvoja softvera a jedna od važnih svrha TickIT-a je bila stimuliranje proizvođača softvera da razmišljaju što je zapravo kvaliteta te kako se ona može postići u razvoju softvera. Kada se na ISO 9001 normi nalazi TickITplus to prikazuje da je taj određeni sustav upravljanja kvalitetom usuglašen sa najboljim praktičnim saznanjima o sustavu upravljanja kvalitetom softvera. Važno je znati da se TickIT može koristiti samo u kombinaciji s ISO 9001 normom te da obuhvaća procjenu i certifikaciju upravljanja kvalitetom softvera u organizaciji prema ISO 9001. Od studenog 2014 godine TickITplus dodaje novu dimenziju postojećoj TickIT shemi kombinirajući najbolje postupke za osiguranja kvalitete softvera u računalnoj industriji s međunarodnim informatičkim standardima. Sa ISO 9001:2008 normom TickITplus pruža akreditirani certifikat sa mogućnosti procjenjivanja razina zrelosti svih tipova IT organizacija. TickITplus je podjelio te razine zrelosti na početnu razinu Nastanka, Brončanu razinu, Srebrnu razinu, Zlatnu razinu i Platinum razinu. TickITplus se povezuje sa ISO/IEC 15504 (Procjena informatičko tehnoloških procesa, eng. Information technology Process assessment) i ISO/IEC 12207 (Sistemska i softverska inženjering – životni ciklus softverskog procesa; eng. Systems and software engineering - Software life cycle processes). Također promovira trening unutar uspostavljenih kvalifikacijskih standarda.

Do listopada 2005. godine bilo je izdano 1093 TickIT certifikata raspodijeljenih u 44 države. Tako da je najviše 807 (73.8%) u Velikoj Britaniji, 131(12%) u ostatku Europe, 89 (8.1%) u sjevernoj Americi (SAD i Kanada) te 66 (6.1%) u ostatku svijeta. Zanimljivo je još napomenuti da je TickIT-om certificiran cijeli niz poznatih proizvođača softvera kao: Oracle, IBM, Hewlett - Packard, Eriksson, Siemens, ali Microsoft nije među njima.<sup>25</sup>

#### **4.2.ISO/IEC 90003**

Standard ISO/IEC 90003 službeno je objavljen 15. veljače 2004. godine kada je zamjenio stari ISO 90003:1997 standard. ISO 90003:1997 objašnjavao je kako se stari ISO 9001:1994 standard može primijeniti na računalni softver. Može se reći da povijest ISO 90003 standarda započinje 1991. godine. Tada je bilo malo standarda koji su se odnosili na

---

<sup>25</sup> SINKOVIĆ, G. (2012.) Upravljanje kvalitetom informacijskih sustava. TECHNE 13. [Online] (Kolovoz). str.15 Dostupno na - [http://www.politehnika-pula.hr/download/repository/Techne\\_br13\\_svibanj\\_2012%5B1%5D.pdf](http://www.politehnika-pula.hr/download/repository/Techne_br13_svibanj_2012%5B1%5D.pdf) [Pristupljeno: 06.07.2015.]

sam softver i njegovu kvalitetu. Nakon što je prva verzija standarda objavljena taj standard nije na dobar način definirao životni vijek softvera te je nakon tri godine došlo do izmjene. A kada je ISO 90003 prihvatio zahtjeve ISO 9001:1994 nastao je ISO 90003:1997. Kada je objavljen ISO 9001:2000 standard to se sve promijenilo jer stari ISO 90003 nije više mogao podržavati ISO 9001 te se pojavila potreba za unapređenjem starog standarda. ISO 90003 je ISO standard razvijen od strane ISO/IEC JTC1, SC7. JTC1 je tehnički odbor unutar ISO organizacije koji je zadužen za standarde svih vrsta informacijskih tehnologija, dok je njegov pododbor SC7 zadužen samo za standarde koji se tiču računalnog softvera i inženjerskih sustava. Prije nego što se krene u objašnjivanje ISO 90003 zanimljivo je istaknuti da je u razvoju standarda bilo uključeno 23 države, a ukupni troškovi iznosili su oko 2 milijuna američkih dolara.

Razlog zašto je ISO 90003 nastao i zašto postoji je povezan s pitanjem što je kvaliteta softvera. Kako je navedeno u prošlom tekstu ISO 90003 je standard koji je povezan s standardom ISO 9001 koji specificira zahtjeve za sustav vođenja kvalitete. Tako razlog zbog kojeg postoji ISO 90003 standard je taj da se ISO 9001 ne može direktno primijeniti na računalni softver. ISO 90003 je nastao kao vodič za primjenu ISO 9001:2000 standarda na računalni softver. Tako ISO 90003 opisuje primjenu zahtjeva standarda ISO 9001 u okviru računalnog softvera. Važno je istaknuti da se ISO 90003 sastoji od dviju klauzula koje su zahtjevi (eng. Requirements) iz standarda ISO 9001 i smjernice (eng. Guidelines). Zahtjevi su nešto što se mora napraviti dok su smjernice opisuju kako primijeniti zahtjeve. ISO 90003 standard zahtijeva poštivanje svih ISO 9001 zahtjeva a poštivanjem tih zahtjeva poduzeća dobivaju ISO certifikat. Kod ISO 90003 uz zahtjeve postoje i smjernice kod kojih postoji dva tipa smjernica koji su preporuke (eng Recommendations) i sugestije (eng. Suggestions). Preporuke opisuju što bi se trebalo napraviti kako bi se zadovoljili ISO 9001 zahtjevi a sugestije se odnose na ono što bi se moglo napraviti kako bi se uklopilo u okvire zahtjeva ISO 9001 standarda.

Zahtjevi i smjernice ISO 90003 se dijele u pet grupa koje su:

1. **Sistemske zahtjevi i smjernice** – oni od organizacije zahtijevaju utvrđivanje kvalitete sustava upravljanja softverskim uslugama te dokumentiranje sustava kvalitete. Kod utvrđivanja kvalitete sustava potrebno je utvrditi procese koji čine taj sustav kvalitete.

Tako se trebaju identificirati poslovni procesi, odrediti slijed i međudjelovanje tih procesa, osigurati kriterije nadzora procesa, osigurati resurse, nadzirati i analizirati procese, uspostaviti radnje potrebne za postizanje planiranih rezultata i stalno se poboljšavati. Dok dokumentiranje sustava kvalitete sadrži zahtjevanu dokumentaciju pod koju spada izjava o politici i ciljevima sustava, poslovnik sustava upravljanja, procedure sustava upravljanja, dokumenti nužni za osiguravanje uspješnog planiranja, funkcioniranja te kontrole procesa organizacije. Također sam opseg dokumentacije se razlikuje od organizacije do organizacije radi toga jer postoji mnoštvo organizacija različitih veličina, vrsta djelatnosti, sposobnosti radnog osoblja, kompleksnosti procesa itd. Dokumentacija može biti u bilo kojoj formi i u bilo kojem obliku.

2. **Upravljački zahtjevi i smjernice** – ova grupa zahtijeva da organizacija podupire kvalitetu kako bi dokazala predanost ISO 9001 standardu, da se usmjerava na kupce, da definira politike kvalitete, da vrši planiranje, da ocjenjuje sustav upravljanja te da uspostavlja kontrolu nad sustavom upravljanja.
3. **Zahtjevi i smjernice za osiguranje resursa** - zahtijevaju od organizacije kvalitetne resurse ( pod to spada utvrđivanje resursa potrebnih za potporu sustavu kvalitete koji su potrebni za upoznavanje sa zahtjevima kupaca i potrebni za upoznavanje sa zahtjevima za ispravnošću softverskog proizvoda), kvalitetno osoblje (osoblje mora biti kompetentno te ta se kompetentnost mora poticati), kvalitetnu infrastrukturu (ovdje je potrebno utvrditi, organizirati i održavati potrebnu infrastrukturu koja se u ovom slučaju odnosi na hardver i softver potreban za razvoj softvera i alate potrebne za izradu, prodaju, zaštitu i kontrolu softvera) te kvalitetnu okolinu (gdje je potrebno utvrditi, organizirati i održavati potrebnu radnu okolinu).
4. **Zahtjevi i smjernice za realizaciju proizvoda** – ova grupa zahtijeva od organizacije kontrolu nad realizacijom proizvoda (treba se koristiti model životnog vijeka proizvoda te prikladne metode razvoja softvera), procesima vezanim uz kupce (upoznavanje sa zahtjevima kupaca, osiguravanje softverskog proizvoda sa samim zahtjevima, procjena rizika zahtjeva na organizaciju i uspostavljanje komunikacije s kupcima), dizajniranjem i razvojem softvera (potrebno je utvrditi koje aktivnosti moraju biti ostvarene, koji su potrebni resursi, koja pravila i konvencije je potrebno poštivati prilikom dizajna i razvoja te je potrebno definirati alate i tehnike koji se koriste prilikom razvoja), postupkom nabave (potrebno je voditi računa o dijelovima softvera koje organizacija dobiva od dostavljača), proizvodnjom i održavanjem (tj.



kontrola nad softverom u procesima proizvodnje i puštanja u funkciju, umnožavanja, dostave, instalacije i održavanja softvera) te opremom zaduženom za mjerenje i nadgledanje (potrebno je odabrati, kalibrirati i zaštititi uređaje koji su prikladni i sposobni za nadgledanje i mjerenje).

5. **Zahtjevi i smjernice za poboljšanjima, mjerenjem i analizom** - zahtijevaju od organizacije da provodi mjerenja, analize i procese poboljšanja kako bi se pokazala usklađenost proizvoda sa zahtjevima i kako bi se kontinuirano poboljšavala kvaliteta sustava upravljanja. To se postiže nadgledanjem i mjerenjem (ovdje se od organizacije zahtjeva definiranje metoda za mjerenje i nadgledanje kupčevog zadovoljstva), upravljanjem neusklađenim softverskim proizvodima (gdje se vrši detekcija i kontrola neusklađenih softverskih zahtjeva), analiziranjem informacija o kvaliteti (gdje je potrebno definirati tip informacija koji je potreban za analizu kvalitete sustava upravljanja) te poduzimanjem akcija za poboljšanjem (gdje organizacija mora ispraviti neusklađenosti softvera sa zahtjevima i spriječiti potencijalnu pojavu neusklađenosti u budućnosti).<sup>26</sup>

Korist od primjene ISO 90003 je unapređenje cjelokupnog sustava upravljanja proizvodnje softverskog proizvoda. Primjenom ISO 90003 standarda organizacija podiže kvalitetu svojih proizvoda te unapređuje svoju komunikaciju s kupcima i samim time unapređuje samu organizaciju. Uz korist od primjene ISO 90003 standarda postoje i posljedice koje se javljaju kada se standard primjeni. Te posljedice su vezane za implementaciju procesa u procese koji su vezani za sam softver. Ti procesi su sustav kvalitete upravljanja, odgovornost vodstva poduzeća, upravljanje resursima, kontrola nad realizacijom proizvoda te mjerenje, analiza i poboljšanja.

### **4.3.Bootstrap metoda**

Bootstrap metoda je metoda koja analizira, redizajnira i unaprijeđuje poslovne procese razvoja softvera. Bootstrap metoda je bazirana prema japanskom stilu rada i managementa zvanom Kaizen koji se u zapadnim zemljama nazivao CIP (Continuous Improvement Programmes; hr. Kontinuirani programi unaprijeđenja). Bootstrap metoda je derivat projekta Esprit 5441 kojeg je vodila Europska unija između 1990. i 1993. godine. Bootstrap institut

---

<sup>26</sup> <http://fly.srk.fer.hr/~sevo/fer/ergonomija/zahtjevi.html>

osnovan 1994. godine je neprofitna organizacija koja vodi, promovira i poboljšava metodu koju zastupa. Ova metoda je specifična po tome što je elemente CMM metode proširila na karakteristike standarda ISO 90003. Ciljevi Bootstrap metode su da procjenjuje softverski proces SPU-a (informatički odjel u velikoj ili većoj kompaniji gdje se razvijaju i održavaju softverski proizvodi) u softverskom tržištu Europe i unaprijeđuje ga. Odnos organizacije (O), tehnologije (T) i metodologije (M) u ovom modelu je :

$$O \gg M \gg T$$

Što znači da nikakva tehnologija ne može riješiti niti unaprijediti metodologiju projektiranja, niti metodologija ne može riješiti problem organizacije. Glavne faze Bootstrap procesa procjene su:

1. **Faza pripreme** – koja se sastoji od edukacije organizacijskog osoblja da bi razvili odgovornost SPU-a. Osoblje tako uči osnove Bootstrap metode poput konceptata i sadržaja te se nakon toga definira tim za procjenjivanje. Kada se tim definira potpisuje se izjava o povjerljivosti koja navodi da su samo profesionalni i licencirani Bootstrap procjenitelji ovlaštene vršiti procjene.
2. **Faza izvršavanja** – kod koje se vrše intervjui organizacijskog osoblja sa ciljem evaluacije tj. procjene procesa proizvodnje. Također se ta projektna procjena vrši procjenom softverskih procesa koji su nastali u fazi pripreme.
3. **Faza planiranja poboljšanja** – kod koje je cilj izraditi završno izvješće. To se sastoji od korištenja rezultata procjene kako bi se definirao ciljani profil za unaprijeđenje i identifikaciju prioriteta. Također u ovoj se fazi vrši verifikacija rezultata procjene.

Bootstrap metoda koristi elemente drugih metoda i normi za osiguranje i unaprijeđenje kvalitete softvera poput CMM-a, ISO 9000 standarda, SPICE terminologije i drugih. Metoda se najviše koristi u Europi. U Bootstrap metodi razina zrelosti poduzeća određuje se putem upitnika koji skupljaju informacije o kriterijima analize postignutih rezultata. Također se u ovoj metodi nalazi i auto-analiza zvana Bootcheck kojom se dobiva uvid u snage i slabosti ispitanih projekata. Važno je napomenuti da se licence za Bootstrap metodu izdaju organizacijama koje su zainteresirane za davanje usluga procjene, organizacijama koje žele procijeniti svoje unutarnje procese, institutima koji se bave razvojem i žele se detaljnije baviti poboljšanjem kvalitete softvera.

## ZAKLJUČAK

Osiguranje kvalitete je važan proces koji omogućava razvoj proizvoda i usluga koje imaju značenje kod kupca. Sami pojam kvalitete varira tj. kupci ponekad imaju drugačiji pogled na kvalitetu nego proizvođači koji se moraju prilagođavati zahtjevima kupaca da bi opstali na tržištu. Tako sam softver kroz tijek svoje proizvodnje treba proći kroz razne izmjene i unaprijeđivanja prije nego što dođe na tržište gdje ga korisnici tj. kupci procjenjuju svojim korištenjem te mu time dodaju ili oduzimaju kvalitetu. Razvoj softvera je težak proces a još teže je održavati i osiguravati kvalitetu tog softvera kada dođe na tržište. U svijetu postoje razne norme, metode i standardi koji pokušavaju svaki na svoj način pomoći softverskim poduzećima da kroz poboljšanje i održavanje svojih organizacijskih procesa osiguraju kvalitetu svojih proizvoda i usluga.

CMM kao metoda za osiguranje softvera nudi svoje smjernice koje su potrebne da bi softver ostao kvalitetan. Kroz svojih pet razina zrelosti CMM kategorizira tipove organizacija koje se bave softverom te im pomaže da dostignu onu željenu razinu te da osiguravaju kvalitetu svojih softverskih proizvoda. Tako kroz procjenu organizacija i njihovih razina zrelosti CMM uspostavlja kriterije za opisivanje karakteristika zrelih softverskih organizacija. Pomoću tih kriterija organizacije mogu prepoznati na kojoj se razini nalaze te koristiti same kriterije kod unaprijeđivanja svojih procesa za razvoj i održavanje softvera. CMM je nastao prvotno kao metoda za rješavanje problema razvoja softvera u vojne svrhe te se može koristiti za procjenu rizika od strane vojske ili komercijalnih organizacija prije nego što budu krenuli raditi na softverskom projektu sa određenim poduzećima. CMM također uz fokus na osiguranje kvalitete softvera i procjenu zrelosti organizacije i njezinih procesa pruža razne preporuke i procedure za područja razvoja, nabave proizvoda te pružanje usluga.

IT poduzeća su danas pod velikim pritiskom od strane svojih kupaca te se poduzeća oslanjaju i na ostale modele i norme koje služe za osiguranje kvalitete softvera poput ISO standarda ili Bootstrap metode. Poduzeća koja prođu procjenu svoje organizacije u cilju nabave potvrde ili certifikata imaju bolje šanse za opstanak na sve većem konkurentnom tržištu softverskim proizvodima gdje se na kvalitetu softvera gleda kao nešto što u zadanom trenutku zadovoljava sve potrebe i zahtjeve korisnika softvera. Osiguranje kvalitete softvera se nalazi među važnim procesima unutar organizacije koja promatra tržište i želje korisnika njezinog softvera.

Korištenjem CMM-a i drugih metoda za osiguranje kvalitete softvera, korištenjem njihovih smjernica za neprestano unaprijeđenje svojih procesa razvoja, unaprijeđenje softvera i zadovoljavanje zahtjeva korisnika, IT poduzeća imaju mogućnost da uspješno posluju u dinamičnom tržištu softverskim proizvodima.

## LITERATURA

### Knjige:

1. KONDIĆ, Ž. (2002.) Kvaliteta i ISO 9000 – primjena -. Varaždin: TIVA
2. ŠIŠKO KULIŠ, M. I GRUBIŠIĆ, D. (2010.) Upravljanje kvalitetom. Split: Sveučilište u Splitu, Ekonomski fakultet
3. PAULK, M. CURTIS, B. CHRISIS, M.B. WEBER C. (1993.) Capability Maturity Model for Software, Version 1.1. Software Engineering Institute
4. SKOKO H. (2000.) Upravljanje kvalitetom, Sinergija, Zagreb

### Članci:

1. CINDRIĆ J. (2009.) ZRELOST ORGANIZACIJE; stručni članak . Str. 180-193. Dostupno na: <http://hrcak.srce.hr/file/78143> [Pristupljeno: 13. lipnja 2015.]
2. GALINAC T. (2008.) Analiza upravljanja kvalitetom u suvremenom evropskom razvoju programskog proizvoda. Str. 65-76. Dostupno na: <http://hrcak.srce.hr/file/48473> [Pristupljeno: 13. lipnja 2015.]
3. ČUBRANIĆ D. KALUŽA M. NOVAK J. (2013.) Standardne metode u funkciji razvoja softvera u Republici Hrvatskoj. Str. 239-256. Dostupno na: <http://hrcak.srce.hr/file/151922> [Pristupljeno: 13. lipnja 2015.]
4. BEVANDA V. SINKOVIĆ G. (2009.) Sustavi znanja u potpori upravljanju kvalitetom softverskog proizvoda. Str. 284-292. Dostupno na: <http://hrcak.srce.hr/file/66287> [Pristupljeno: 13. lipnja 2015.]
5. OREŠKI P. (1990) Metrika softvera. Str. 75-87 Dostupno na: <http://hrcak.srce.hr/file/118917> [Pristupljeno: 13. lipnja 2015.]

6. DAY B. LUTTEROTH C. (2011) Climbing the ladder: capability maturity model integration level 3. *Enterprise Information Systems*. 5 (2). Str. 125–144 Dostupno na: [http://www.researchgate.net/publication/220478940\\_Climbing\\_the\\_ladder\\_capability\\_maturity\\_model\\_integration\\_level\\_3](http://www.researchgate.net/publication/220478940_Climbing_the_ladder_capability_maturity_model_integration_level_3) [Pristupljeno: 13. lipnja 2015.]
7. NEWSHAW F. (2005.) Capability Maturity model integration: Technical writers needed. *Intercom*. 9 (11). Str. 13-15 Dostupno na: <http://connection.ebscohost.com/c/articles/18716353/capability-maturity-model-integration-technical-writers-needed> [Pristupljeno: 13. lipnja 2015.]
8. GRIGGS G. (2004.) Quality Management of the Software Industry. Dostupno na: <http://www.umsl.edu/~sauterv/analysis/CMMandISOPaper.pdf> [Pristupljeno: 13. lipnja 2015.]
9. KASSER J. HITCHINS D. FRANK M. ZHAO Y. (2012) A Framework for Benchmarking Competency Assessment Models. *Systems Engineering*. 16. Str. 29–44 Dostupno na: <http://onlinelibrary.wiley.com/doi/10.1002/sys.21217/abstract> [Pristupljeno: 13. lipnja 2015.]
10. SHARI S. SHANGA C. LINB S. (2009.) Understanding the effectiveness of Capability Maturity Model Integration by examining the knowledge management of software development processes. *Total Quality Management* 20 (5). str. 509-521 Dostupno na: [http://www.researchgate.net/publication/247523684\\_Understanding\\_the\\_effectiveness\\_of\\_Capability\\_Maturity\\_Model\\_Integration\\_by\\_examining\\_the\\_knowledge\\_management\\_of\\_software\\_development\\_processes](http://www.researchgate.net/publication/247523684_Understanding_the_effectiveness_of_Capability_Maturity_Model_Integration_by_examining_the_knowledge_management_of_software_development_processes) [Pristupljeno: 13. lipnja 2015.]
11. SINKOVIĆ G. TROŠT F. (2012) Upravljanje kvalitetom informacijskih sustava. *Techne* 13. (8). Str. 10-16 Dostupno na: [http://www.politehnika-pula.hr/download/repository/Techne\\_br13\\_svibanj\\_2012%5B1%5D.pdf](http://www.politehnika-pula.hr/download/repository/Techne_br13_svibanj_2012%5B1%5D.pdf) [Pristupljeno: 13. lipnja 2015.]
12. MATEJAŠ M. Metode ugradnje sustava za upravljanje poslovnim procesima. Dostupno na:

[https://www.fer.unizg.hr/download/repository/KDI\\_Mladen\\_Matejas.pdf](https://www.fer.unizg.hr/download/repository/KDI_Mladen_Matejas.pdf)

[Pristupljeno: 13. lipnja 2015.]

### Internet stranice:

1. <http://searchsoftwarequality.techtarget.com/definition/Capability-Maturity-Model>  
[Pristupljeno: 13. lipnja 2015.]
2. <http://www.svijet-kvalitete.com/index.php/upravljanje-kvalitetom/1918-cmmi-metodologija> [Pristupljeno: 13. lipnja 2015.]
3. <http://cmmiinstitute.com/#home> [Pristupljeno: 13. lipnja 2015.]
4. <http://mcqsets.com/s/fundamentals/computer-software-concept/> [Pristupljeno: 13. lipnja 2015.]
5. <http://fly.srk.fer.hr/~sevo/fer/ergonomija/> [Pristupljeno: 13. lipnja 2015.]
6. <http://www.praxiom.com/iso-90003.htm> [Pristupljeno: 13. lipnja 2015.]
7. <http://web.efzg.hr/dok//inf/pozgaj/pisani%20materijali/T12%20Procjena%20kvalitete.pdf> [Pristupljeno: 13. lipnja 2015.]
8. <http://iso25000.com/index.php/en/iso-25000-standards> [Pristupljeno: 13. lipnja 2015.]
9. <http://web.efzg.hr/dok//inf/pozgaj/sazeci/12%20Procjena%20kvalitete.pdf>  
[Pristupljeno: 13. lipnja 2015.]
10. <https://sas.cmmiinstitute.com/pars/pars.aspx> [Pristupljeno: 13. lipnja 2015.]
11. <http://www.tuv-nord.com/hr/certifikacija-sustava/iso-9001-454.htm> [Pristupljeno: 13. lipnja 2015.]

12. <http://www.qualitas.hr/poslovno-savjetovanje/iso-9001-sustavi-upravljanja-kvalitetom.html> [Pristupljeno: 13. lipnja 2015.]
13. <http://foswiki.cs.uu.nl/foswiki/MethodEngineering/TheBOOTSTRAPApproach20112012> [Pristupljeno: 13. lipnja 2015.]
14. <http://www.kagor.hr/hr/usluge/implementacija-iso-standarda/iso-9001/> [Pristupljeno: 13. lipnja 2015.]
15. <http://www.ticketplus.org/information.aspx> [Pristupljeno: 13. lipnja 2015.]
16. <http://www.sei.cmu.edu/reports/10tr033.pdf> [Pristupljeno: 13. lipnja 2015.]
17. <https://www.sei.cmu.edu/reports/07tn009.pdf> [Pristupljeno: 13. lipnja 2015.]
18. <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1622&context=sei> [Pristupljeno: 13. lipnja 2015.]
19. <https://www.sei.cmu.edu/reports/09tr003.pdf> [Pristupljeno: 13. lipnja 2015.]
20. <http://limun.hr/main.aspx?id=13738&Page=2> [Pristupljeno: 02. kolovoza 2015.]
21. <http://struna.ihjj.hr/naziv/planiranje-kvalitete/18270/> [Pristupljeno: 02. kolovoza 2015.]
22. <http://ibiblio.org/gferg/ldp/SCM-OpenSource/cmm.png> [Pristupljeno: 10. kolovoza 2015.]
23. [http://www.masfak.ni.ac.rs/milan.zdravkovic/files/articles/2005\\_CMM.pdf](http://www.masfak.ni.ac.rs/milan.zdravkovic/files/articles/2005_CMM.pdf) [Pristupljeno: 10. kolovoza 2015.]



## POPIS SLIKA

Slika 1. Koncept softvera.....	4
Slika 2. Vodopadni model životnog ciklusa softvera.....	7
Slika 3. Karakteristike norme ISO 25000.....	15
Slika 4. Razine zrelosti CMM-a.....	24
Slika 5. Certifikat pete razine zrelosti.....	37
Slika 6. CMM struktura.....	39
Slika 7. Poznata poduzeća certificirana prema CMM-u.....	52

## POPIS TABLICA

Tablica 1. Najpoznatije CASE tehnologije.....	10
Tablica 2. Obilježja i ključni procesi razina zrelosti.....	26
Tablica 3. CMM statistika za projekte od 200.000 LOC.....	33
Tablica 4. Razine zrelosti 4 i 5 u svijetu 2004 godine.....	38
Tablica 5. Broj certificiranih organizacija.....	50