

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Igor Rožanc

Uvod v kakovost programske opreme

**Študijsko gradivo za interno uporabo pri
predmetu Testiranje in kakovost (TiK)**

Ljubljana, 2017/18

Vsebina:

- Kakovost v Tehnologiji programske opreme (Software Engineering)
- Kakovost programske opreme
- Zagotavljanje kakovosti programske opreme (QA)
- MODELI in STANDARDI KAKOVOSTI
 - Zmožnostno zrelostni model zagotavljanja kakovosti programske opreme (CMM)
 - Skupinski proces razvoja programske opreme (TSP)
 - Osebni proces razvoja programske opreme (PSP)



Tehnologija programske opreme je ...

1

- vzpostavitev in uporaba zdravih inženirskih principov s ciljem ekonomičnega razvoja zanesljive programske opreme, ki učinkovito deluje v realnem okolju (Nauer, 1969)
- sistematičen pristop pri razvoju, uporabi, vzdrževanju in "upokojitvi" programske opreme (IEEE, 1983)
- tehnološka in menedžerska panoga, ki se ukvarja z sistematičnim razvojem in vzdrževanjem programskih produktov, ki so razviti pravočasno in v okviru predvidenih stroškov (Fairley, 1984)
- opisuje sistematične, metodološke, merljive in (predvsem) v praksi učinkovite pristope pri razvoju, uporabi in vzdrževanju programske opreme. Obsega znanja, orodja in metode za zajem in upravljanje zahtev, učinkovito načrtovanje, izdelavo programskih izdelkov, oblikovanje uporabniških vmesnikov, izvedbo testiranja in vzdrževanje programskih izdelkov. (Wikipedia, 2018)

Kakovost v Tehnologiji programske opreme

2

TPO je širok pojem, ki pokriva številna področja:

- 1.) Specifikacija programske opreme (kaj)
- 2.) Načrtovanje programske opreme (kako)
- 3.) Programerske tehnike in orodja (na kakšen način)
- 4.) **Validacija programske opreme**
- 5.) Vodenje projektov razvoja programske opreme

Vsako področje tako ali drugače zajema kakovost programske opreme, **pri nekaterih je zagotavljanje kakovosti glavno vodilo**

1.) Specifikacija programske opreme (kaj)

- a) definicija zahtev (Software Requirements Definition)
- b) modeliranje sistema (System Modelling)
- c) specifikacija zahtev (Requirements Specification)
- d) strukturirane metode (neformalne, npr. SSADM, Information Engineering,...)
- e) formalne metode (stroga matematična podlaga, npr. VDM, Z sheme)
- f) prototipiranje (Software Prototyping)

2.) Načrtovanje programske opreme (kako)

- a) funkcijsko usmerjeno načrtovanje
- b) objektno usmerjeno načrtovanje
- c) načrtovanje sistemov v realnem času
- d) načrtovanje uporabniških vmesnikov

3.) Programerske tehnike in orodja (na kakšen način)

- a) zanesljivost programske opreme
- b) ponovna uporaba programske opreme
- c) CASE
- d) okolja za razvoj programske opreme (SDE)

4.) Validacija programske opreme

- a) verifikacija in validacija
- b) zanesljivost programske opreme
- c) varnost
- d) testiranje (ugotavljanje in odpravljanje napak)
- e) orodja za testiranje
- f) statično preverjanje kode

5.) Vodenje projektov razvoja PO

- a) izdelava planov in tehnike mrežnega planiranja
- b) ocenjevanje stroškov razvoja
- c) vzdrževanje programske opreme
- d) upravljanje s konfiguracijo (Configuration Management)
- e) izdelava dokumentacije
- f) **zagotavljanje kakovosti (modeli in standardi)**

TPO pomeni uporabo računalniških znanj in računalnikov za reševanje realnih problemov

- ločimo razumevanje problema od izvedbe rešitve
- osnovni princip: analiza - sinteza
- reševanje problemov uporabljamo metode, orodja, postopke in paradigme
 - **metoda** (tehnika): priprava določene jedi (sestavine, čas, ne pribor)
 - **orodje**: vilice ali stepalnik ("boljša", "lažja" izvedba)
 - **postopek**: recept (metode in orodja za določen rezultat)
 - **paradigma**: kuharski stil

Sistematičen princip reševanja problemov zagotavlja izvajanje kakovostnega razvoja programske opreme

Razmerje Znanost : Tehnologija (Science : Engineering)

- podatki, dejstva, pravila, teorije -> znanje na določenem področju : uporaba tega znanja za reševanje primernih problemov
- primer 1: kemik raziskuje kemične snovi, sestavo, povezave, pojasnjuje naravo kemijskih reakcij, lastnosti snovi : inženir kemije uporablja ta znanja pri proizvodnji zdravil ali za ugotavljanje čistosti vode
- primer 2: računalništvo proučuje arhitekturo računalnikov, programske jezike, algoritme, izračunljivost, teorijo informacij, ... : TPO uporablja ta znanja za razvoj konkretne programske opreme (informacijski sistem, spletna aplikacija, programska podpora za iskanje cilja podmornice)
- cilj je ustrezno predstaviti in pojasniti koncepte na določenem področju: **cilj je kakovostna rešitev** (pravilna, učinkovita, zanesljiva, uporabna, robustna, vzdržljiva, prilagodljiva, ...)

Zakaj je kakovost pomembna?

8

- določa stopnjo odličnosti rešitve
- uporabniki jo zahtevajo
- vsi jo pričakujejo
- vsi menijo, da jo razumejo (čeprav različno)
- pomeni, da smo dosegli cilj (zadovoljili uporabnika)

Z vidika podjetja:

- nujna za preživetje
- potrebna za uspeh, ugled, prepoznavnost, povečanje dobička

- Odgovornost : izvajalci / vsi
- Vidnost napak : skrivanje / osveščanje
- Učinek: sramota / korektnost
- Dokumentiranje: čimmanj / izdatno
- Stroški kakovosti: večji / manjši
- Pogled na kakovost: izvajalec / stranka
- Izvedba: nadzor / ustrezen postopek

- PO je glavna (najdražja) komponenta računalniškega sistema
- 80% stroškov za stvari, ki tako ali drugače vsebujejo PO
- trend naraščanja uporabe PO
- naraščajo tudi pričakovanja glede kakovosti
- napake v PO so izredno drage (denar, čas)
- slabe izkušnje ...

VENDAR:

- vedno več in bolj zapletena PO
- nove ali specifične zahteve za kakovost PO (spletna varnost)
- vedno krajši časi za razvoj PO

Slabe izkušnje ...

11

- Hopperjev hrošč na relejskem stroju
- Davčna uprava ZDA (1980 - 1996) : zaradi neprimernega planiranja cca. trikratni stroški in neuporaben rezultat
- PO za ameriško strateško obrambo (1980, 2000): nuklearni napad 1980 - ni mogoče preveriti /zagotoviti delovanja brez napak
- Protiraketni sistem v ZDA: preverjanje? 10 000 000 vrstic kode, Space Shuttle v celoti zahteva 3 000 000 vrstic 2005, 100 000 leta 1985
- Terac 25 (1993) : napaka na obsevalniku povzročila smrt več ljudi
- Pentium Bug (1994): napačna inicializacija 5 od 1066 polj
- Ariane 5 (1996) – nepokrita izjema pri računanju z realnimi števili – eksplozija po 35 sekundah, 370 mio EUR škode
- Sea Harrier in radarska pištola, Škotska (1996)
- Mars Polar Lander (1999) – izguba vozila zaradi neusklajenih merskih enot
- „Severovzhodna zatemnitev“ (2003) – preobremenitev, napaka pri sporočanju napake, odpoved 256 elektrarn, 50 mio prizadetih, 6 mlrd USD škode
- Nova Ljubljanska Banka – sistem Sigma (2003)

Kaj je kakovost programske opreme?

12

- Stopnja ustreznosti zahtevam oz stopnja ustreznosti pričakovanjem in potrebam uporabnika (za sistem, komponento ali proces) (IEEE)
- Doseganje visoke stopnje zadovoljstva, prenosljivosti, vzdrževalnosti, robustnosti, primernosti za uporabo (Barry Boehm)
- Ustreznost z uporabnikovimi zahtevami (Phil Crosby)
- Usmerjenost k odličnosti glede zanesljivosti in funkcionalnosti z izboljševanjem procesa in statističnim merjenjem (Edward Deming)
- Doseganje visoke stopnje zrelosti - primernosti za uporabo, skladnosti zahtev, zanesljivosti, vzdrževalnosti (Watts Humphrey)
- Doseči uporabnikove zahteve pravočasno in brez prekoračitve porabe virov (James Martin)
- Doseči visoko stopnjo uporabnikovega zadovoljstva, brez napak, z nizko zapletenostjo izdelkov (Tom McCabe)
- ...

a) Osnove kakovosti programske opreme

- Kultura in etika TPO
- Vrednost in strošek kakovosti
- **Modeli kakovosti (nabor značilnosti)**
- Izboljševanje kakovosti (procesa in izdelka)

b) Proces upravljanja kakovosti programske opreme

- **Zagotavljanje kakovosti PO (planiranje not./zun.lastnosti, množica ukrepov)**
- Verifikacija in validacija
- Ponovni pregledi in ocenjevanje

c) Praktični pogled

- Zahteve za kakovost aplikacije
- Značilnosti napak
- Tehnike za upravljanje kakovosti programske opreme
- Merjenje kakovosti programske opreme

Kakovost procesa : kakovost izdelka

- Predpostavka: proces neporedno vpliva na izdelek
- Kakovost procesa: standardi, nadzor, poročanje = **zagotavljanje kakovosti** (Software Quality Assurance - SQA)
- Kakovost izdelka: preverjanje ustreznosti izdelka zahtevam = **nadzor kakovosti** (Quality Control)

Proces zagotavljanja kakovosti programske opreme

- Planiranje notranjih in zunanjih značilnosti kakovosti
- Vsebuje sistematično planiranje, izvajanje in nadzor množice ukrepov v procesu razvoja, ki (s primerno stopnjo zaupanja) prepričajo o tehnični ustreznosti, upravljanju in stroških izdelka
- Obsega določanje procesa, izbiro orodij, preverjanja kakovosti, usposabljanje...

1. Zavedati se, kaj počneš

- Kaj natanko izdelujemo?
- Kako to počnemo?
- Kako izdelek deluje?
- Zahteva: planiranje, spremljanje, poročanje

2. Zavedati se, kaj bi moral početi

- Imeti jasne zahteve
- Definirati specifikacije
- Določite sprejemne teste

3. Vedeti, kako določiti (prepoznati) razliko

- Definirati ustrezne metode: formalne metode, teste, preglede, metrike

Uporaba modelov/standardov zagotavlja kakovostne izdelke

Ključno vodilo: dorečen (discipliniran) proces razvoja zagotavlja kakovostno izvedbo vsega razvoja PO, to pa kakovostne izdelke

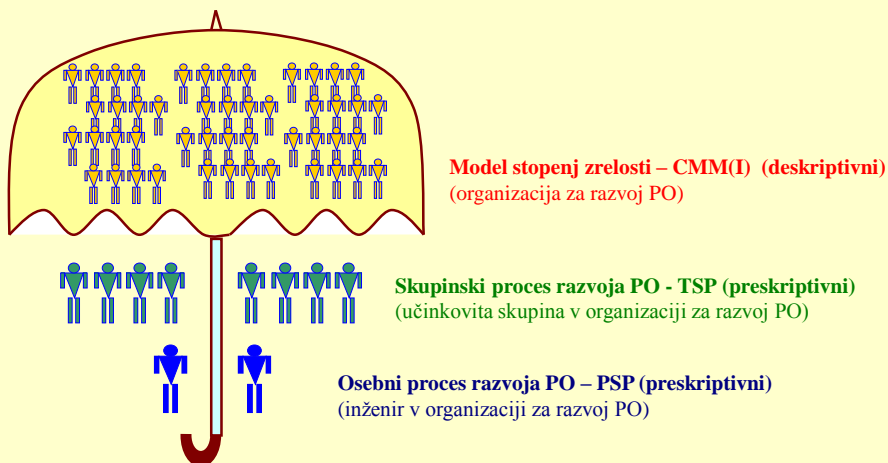
- +: skupne dobre prakse so se prelile v splošno sprejete standarde/modele
- +: obstaja veliko preverjeno uspešnih modelov za razvoj PO
- +: so zelo uporabljani, ključni dejavnik uspeha
- -: tipično namenjeni za velike organizacije
- -: velika poraba resursov, zahtevna uporaba
- -: problem prilagoditve za majhne organizacije, skupine in posameznike

Primeri: družina CMM modelov, ISO standardi, Six Sigma



Družina CMM modelov

17



© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Model stopenj zrelosti – CMMI (Capability Maturity Model Integration)

18

Watts Humphrey, Software Engineering Institute pri CMU

Model za ovrednotenje kakovosti procesa razvoja PO

Sprva model za ocenjevanje razvijalcev PO za vladne organizacije

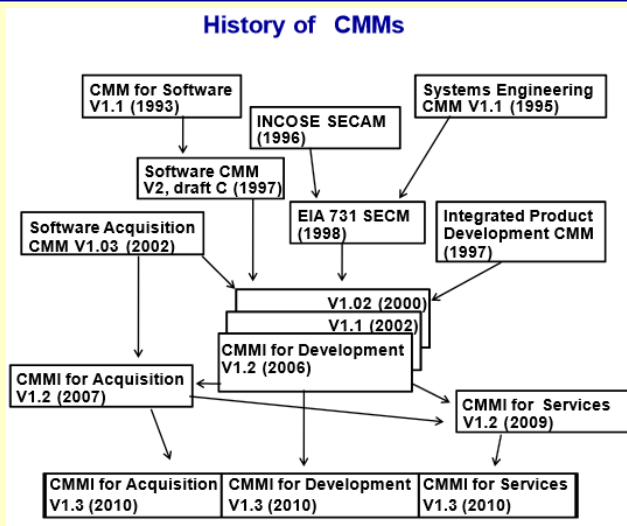
Različice:

- osnutek (1988), CMM v1.0 (1991), CMM v1.1 (1993),
- CMMI (po 1998): SE-CMM za sis.inženiring, SA-CMM za nakup PO...
- CMMI verzija 1.1 (avgust 2002) - ločen del CMMI-SW
- CMMI verzija 1.2 (avgust 2006)
- danes tri ločeni deli CMMI verzije 1.3 (december 2010):
 - CMMI-DEV – za razvoj izdelkov in storitev
 - CMMI-SVC - za vzpostavitev, upravljanje in izvajanje storitev
 - CMMI-ACQ – za nabavo izdelkov in storitev

Od marca 2016 je CMMI Institute pri SEI del ISACA (Information Systems Audit and Control Association)

© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



CMMI določa kakovosten, dobro definiran in nadziran proces razvoja PO v podjetju.

Namen:

- integrira tradicionalno ločene funkcije organizacije,
- določa cilje in prioritete za izboljševanje procesa organizacije,
- usmerja proces izboljšave kakovosti v podjetju,
- omogoča primerjavo trenutnega procesa s procesi primerljivih podjetij.

De facto industrijski standard



Ideja CMM

21

Total Quality Management (TQM): graditi na kakovosti - za krajši razvojni cikel, večjo produktivnost, večje zadovoljstvo in poslovni uspeh strank

Hierarhičen model:

- več (5) zrelostnih nivojev (ang. maturity level)
- več (ključnih) področij (ang. (key) process area ali (K)PA),
- več ciljev (ang. goal),
- več tipičnih praks (ang. key practices)
- veliko podpraks (ang. subpractices) in primerov (ang. examples).

Elementi dejansko določajo strukturo organizacije, vloge, aktivnosti, dokumente...

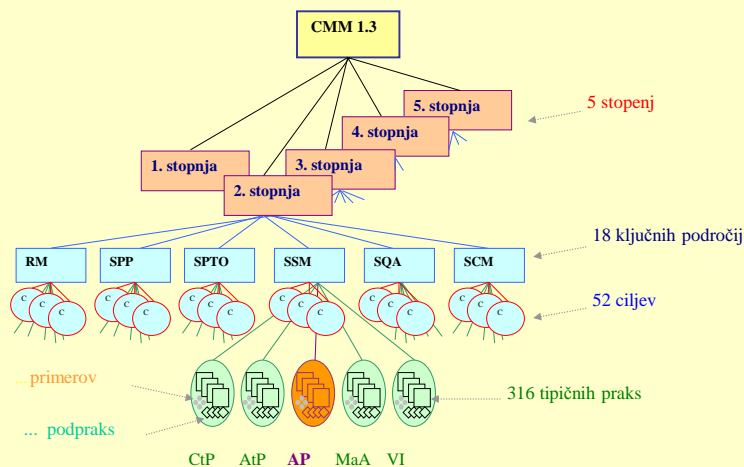
CMM dejansko definira 5 zrelostnih nivojev organizacije =>

ta na primeren način dosega vse cilje, ki jih določajo ključna področja tega (in vseh nižjih) nivojev.



Struktura CMM 1.3

22





Način predstavitve CMM

23

Predstavljen na dva enakovredna načina:

a) **NIVOJSKO** (staged):

- več nivojev zrelosti procesa organizacije,
- na vsakem nivoju določena ključna področja, cilji, tipične prakse
- **Namen: doseganje zrelostnega nivoja**

b) **ZAPOREDNO** (continuous):

- posamezna ključna področja, cilji, tipične prakse
- kako dobro so pokriti cilji ključnega področja
- **Namen: doseganje zrelosti ključnega področja**



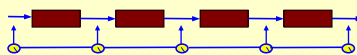
Učinek CMM nivojev

24

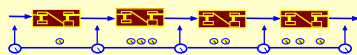
1. Kaotičen



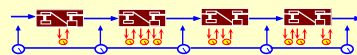
2. Ponovljiv



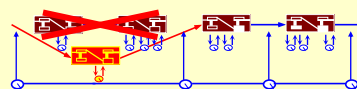
3. Definiran



4. Upravljan



5. Optimizirajoč





CMM 1.3 podrobno

25

Stopnja CMM	Ključna področja		
5. stopnja Optimizirajoč proces	Preprečevanje napak	Defect Prevention	DP
	Uvajanje tehnoloških sprememb	Technology Change Management	TCM
	Uvajanje sprememb v procesu	Process Change Management	PCM
4. stopnja Upravljan proces	Kvantitativno vodenje procesov	Quantitative Process Management	QPM
	Upravljanje za kakovost PO	Software Quality Management	SQM
3. stopnja Definiran proces	Osredotočenje na proces organizacije	Organization Process Focus	OPF
	Definicija procesa razvoja PO	Organization Process Definition	OPD
	Program izobraževanja	Training Program	TP
	Integrirano vodenje projektov PO	Integrated Software Management	IPM
	Tehnologija izdelkov PO	Software Product Engineering	SPE
	Koordinacija različnih skupin	Intergroup Coordination	IC
	Pregledi med kolegi	Peer Reviews	PR
2. stopnja Ponovljiv proces	Upravljanje z zahtevami	Requirements Management	RM
	Planiranje projektov	Software Project Planning	SPP
	Zasledovanje in nadzor nad projekti	Software Project Tracking & Oversight	SPTO
	Urejanje razmerij s podizvajatelji	Software Subcontract Management	SSM
	Zagotavljanje kakovosti PO	Software Quality Assurance	SQA
	Upravljanje s konfiguracijo	Software Configuration Management	SCM
1. stopnja Kaotičen proces			

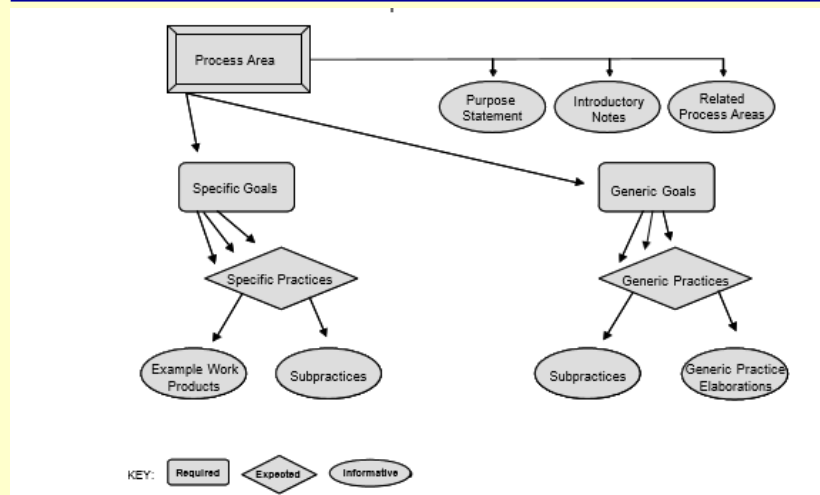
© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Opis CMMI

26



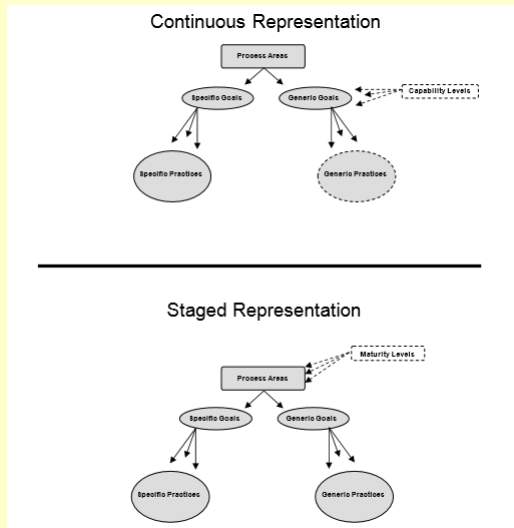
© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Dva načina predstavitve CMMI

27



© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Dva načina predstavitve CMMI

28

Level	Continuous Representation Capability Levels	Staged Representation Maturity Levels
Level 0	Incomplete	
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4		Quantitatively Managed
Level 5		Optimizing

© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



CMMI podrobno

29

Name	Abbr.	ML	CL1	CL2	CL3
Configuration Management	CM	2	Target Profile 2		
Measurement and Analysis	MA	2			
Project Monitoring and Control	PMC	2			
Project Planning	PP	2			
Process and Product Quality Assurance	PPQA	2			
Requirements Management	REQM	2			
Supplier Agreement Management	SAM	2			
Decision Analysis and Resolution	DAR	3	Target Profile 3		
Integrated Project Management	IPM	3			
Organizational Process Definition	OPD	3			
Organizational Process Focus	OPF	3			
Organizational Training	OT	3			
Product Integration	PI	3			
Requirements Development	RD	3			
Risk Management	RSKM	3			
Technical Solution	TS	3			
Validation	VAL	3			
Verification	VER	3			
Organizational Process Performance	OPP	4	Target Profile 4		
Quantitative Project Management	QPM	4			
Causal Analysis and Resolution	CAR	5	Target Profile 5		
Organizational Performance Management	OPM	5			

© Igor Rožanc

Univerza v Ljubljani, fakulteta za računalništvo in informatiko



Kategorije procesnih področij CMM in CMMI

30

Tri (štiri) kategorije (ključnih) procesnih področij (KPA):

- Upravljanje projekta (Project Management)
- Upravljanje procesa (Proces Management)
- Tehnologija (Engineering)
- Podpora (Support)

© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Kategorije procesnih področij CMM in CMMI

31

Capability Maturity Model Integration (CMMI) Core Process Areas			
Abbreviation	Name	Area	Maturity Level
CAR	Causal Analysis and Resolution	Support	5
CM	Configuration Management	Support	2
DAR	Decision Analysis and Resolution	Support	3
IPM	Integrated Project Management	Project Management	3
MA	Measurement and Analysis	Support	2
OPD	Organizational Process Definition	Process Management	3
OPF	Organizational Process Focus	Process Management	3
OPM	Organizational Performance Management	Process Management	5
OPP	Organizational Process Performance	Process Management	4
OT	Organizational Training	Process Management	3
PMC	Project Monitoring and Control	Project Management	2
PP	Project Planning	Project Management	2
PPQA	Process and Product Quality Assurance	Support	2
QPM	Quantitative Project Management	Project Management	4
REQM	Requirements Management	Project Management	2
RSKM	Risk Management	Project Management	3
SAM	Supplier Agreement Management	Support	2



CMMI konkretno

32

GENERIC GOALS			
GG-1	Achieve Specific Goals		
The specific goals of the process area are supported by the process by transforming identifiable input work products into identifiable output work products.			
GP 1.1	Perform the specific practices of the process area to develop work products and provide services to achieve the specific goals of the process area.		
GG-2	Institutionalize a Managed Process		
The process is institutionalized as a managed process.			
GP 2.1	Establish and maintain an organizational policy for planning and performing the process.		
GP 2.2	Establish and maintain the plan for performing the process.		
GP 2.3	Provide adequate resources for performing the process, developing the work products, and providing the services of the process.		
GP 2.4	Assign responsibility and authority for performing the process, developing the work products, and providing the services of the process.		
GP 2.5	Train the people performing or supporting the process as needed.		
GP 2.6	Place selected work products of the process under appropriate levels of control.		
GP 2.7	Identify and involve the relevant stakeholders of the process as planned.		
GP 2.8	Monitor and control the process against the plan for performing the process and take appropriate corrective action.		
GP 2.9	Objectively evaluate adherence of the process and selected work products against the process description, standards, and procedures, and address noncompliance.		
GP 2.10	Review the activities, status, and results of the process with higher level management and resolve issues.		
GG-3	Institutionalize a Defined Process		
The process is institutionalized as a defined process.			
GP 3.1	Establish and maintain the description of a defined process.		
GP 3.2	Collect process related experiences derived from planning and performing the process to support the future use and improvement of the organization's processes and process assets.		



CMMI konkretno

33

CONFIGURATION MANAGEMENT - CM		SUPPORT	(ML2)
The purpose of Configuration Management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.			
SG 1	Baselines of identified work products are established.		
SP 1.1	Identify configuration items, components, and related work products to be placed under configuration management.		
SP 1.2	Establish and maintain a configuration management and change management system for controlling work products.		
SP 1.3	Create or release baselines for internal use and for delivery to the customer.		
SG 2	Changes to the work products under configuration management are tracked and controlled.		
SP 2.1	Track change requests for configuration items.		
SP 2.2	Control changes to configuration items.		
SG 3	Integrity of baselines is established and maintained.		
SP 3.1	Establish and maintain records describing configuration items.		
SP 3.2	Perform configuration audits to maintain the integrity of configuration baselines.		



Uporaba CMM in CMMI

34

- Objektivna ocena procesa v podjetju
 - za informiranje potencialnih strank
 - na zahtevo naročnika
- Izboljšanje kakovosti v podjetju
 - kje smo – primerjava z najboljšimi praksami CMMI
 - pot rasti – ključne možnosti napredka



Ocenjevanje

35

Appraisal, not certification

Appraisal Requirements for CMMI – ARC zahteve

Standard CMMI Appraisal Method for Process Improvement – SCAMPI

SCAMPI Appraisal results: nivo 1 – 2 (5 mesecev), nivo 1 – 3 (21 mesecev)

Izvedba ocenjevanja

- A – najbolj formalen, zrelost nivoja
- B – formalen, področje, lahko objava ali ne (SCAMPI)
- C – neformalen SCAMPI C



Težave CMM in CMMI

36

- Ustreznost modela za konkretne organizacije
- Stroški (uvajanje, usposabljanje, certificiranje, sprememba organizacije)
- Zahtevnost – primeren za velike organizacije:
 - število področij, ki jih pokriva
 - organizacija: število vlog in skupin
 - dokumentacija
- Velik začeten statičen vložek za dolgoročen uspeh in konkurenčnost
- Nasprotje CMM – agilnost
- Ni edini način za doseganje kvalitetnega poslovanja podjetja (ZDA : EU standardi)



Skupinski proces razvoja PO – TSP (Team Software Process)

37

Skupinski proces razvoja PO:

- nadgradnja osebnega procesa razvoja, ki določa discipliniran proces za učinkovito inženirsko delo skupine razvijalcev PO
- implementira CMM principe na ravni razvojnih skupin

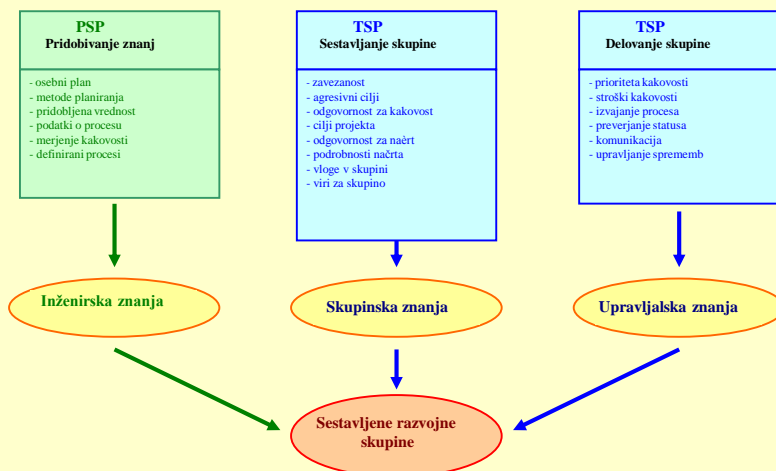
SEI, Watts Humphrey

- začetek razvoja: leto 1996, več verzij
- definira pristope in metode za skupine z manj kot 20 člani:
 - optimalno 5 do 12 članov,
 - razvoj različice mTSP za skupine do 150 razvijalcev
- danes: usposabljanje trenerjev za projektne ekipe
- aktualno: uporaba TSP za majhna podjetja



Skupinski proces razvoja PO – TSP (Team Software Process)

38





Delovanje TSP

39

Pridobivanje znanj

- Članov ekipe (PSP), trenerja, vodstva

Sestavljanje ekipe

- Vodi usposobljeni trener
- Določanje ciljev (ekipa in vodstvo)
- Določanje vlog
- Prepoznavanje tveganj
- Ocena napora
- Razporeditev zadolžitev
- Načrt izvedbe

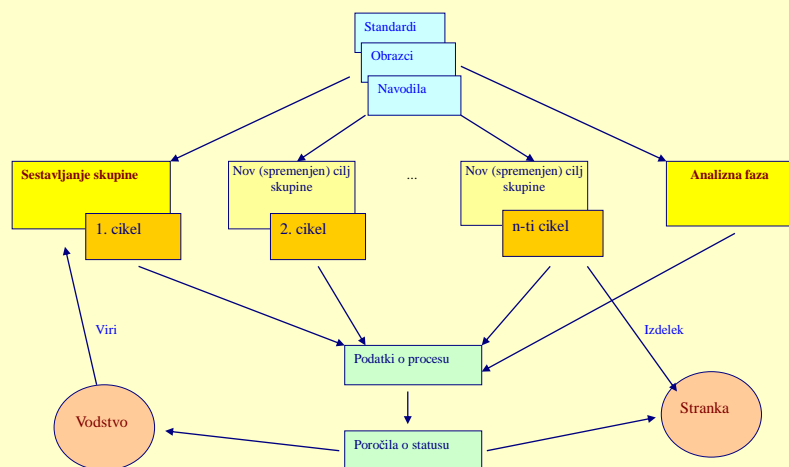
Delovanje ekipe

- Izvajanje razvoja
- Spremljanje planiranega/dejanskega napora, poteka, napak
- Tedenska srečanja
- Redno poročanje
- Usklajevanje planov



Shema delovanja TSP

40





Dokumenti TSP

41

Navodilo (ang. script):

- prikaz vseh korakov, ki predstavljajo proces ali del procesa
- koraki so podani kot kratek opis aktivnosti (v enem stavku)
- več zaporednih korakov služi določenemu namenu
- nekateri deli so lahko bolj natančno definirani v posebnem navodilu
- navodilo ima definirane tudi vhodne in izhodne kriterije.

Obrazec (ang. form):

- vnaprej določene predloge za zbiranje in zapisovanje rezultatov v natanko določeni obliki
- natanko določajo število in vrsto podatkov, nam služijo kot opomnik, kaj je treba opraviti.



Dokumenti TSP

42

Zapisek (ang. log)

- podobni formularjem s to razliko, da jih uporabimo samo po potrebi
- manj podatkov
- tipično služijo za zapis izmerjenih rezultatov

Standard (ang. standard)

- opis, kako pravilno izvajati določene aktivnosti (recimo štetje vrstic kode pri merjenju velikosti).
- jasno opredeljeni elementi: opis, tipične vrednosti, primeri, izjeme ipd.



Skupinski proces razvoja PO – TSP

43

Razmerje med CMM in TSP

- ustrezno obravnava projektni del razvoja
- CMM ne more nadomestiti v celoti – manjkajo organizacijske aktivnosti
- primeren za začetek uvajanja CMM v organizaciji
- formalna alternativna pot za CMMI izboljšanje procesa (2010)
- dosledna uporaba TSP-ja dosega zelo dobre rezultate

Usposabljanje za TSP

- TSP trenerji (coach): skrb za zagotavljanje učinkovitosti delovanja ekipe
- Uspešen program usposabljanj za trenerje



Osebni proces razvoja PO – PSP (ang. Personal Software Process)

44

Osebni proces razvoja (PSP) definira nadziran proces razvoja PO za inženirje v organizaciji za razvoj PO.

Konkretnije: množica metod, predlog in navodil o tem, kako načrtovati, meriti in voditi potek dela inženirja, ki dela v razvoju PO.

Implementira CMM principe na ravni osebnega dela inženirjev:

- podmožico primernih metod in praks so prilagodili ter sestavili v model
- programer postopoma razvija svoj proces:
 - začne z osnovnimi postopki
 - te postopoma nadgrajuje z uvedbo merjenja, principov za zagotavljanje kakovosti dela in aktivnosti za obvladovanje večje količine dela
 - slednjič doseže optimalen proces, ki mu omogoča popolno obvladovanje svojega dela

Povezava z Six Sigma pristopom in agilnimi metodami



PSP principi

45

Splošni cilj: pravočasen in poceni razvoj PO brez napak.

PSP principi:

- Vsak inženir je različen; da bi bil pri svojem delu kar najbolj učinkovit, mora skrbno načrtovati svoje delo na osnovi preteklih izkušenj.
- Izboljšanje učinkovitosti dela zahteva discipliniran proces razvoja PO z merjenjem izdelkov (velikosti, napak) in vloženega napora.
- Inženirji lahko izdelajo kakovostne izdelke samo tako, da se čutijo osebno zavezane in odgovorne za najprimernejši način dela, ki do tega vodi.
- Zgodnje odpravljanje napak v procesu je cenejše od kasnejšega; še ceneje je preprečiti nastanek napak.
- Najboljši način dela je tisti, ki najhitreje in najceneje vodi do rezultata.



Sedem verzij PSP

46

Osebni procesi od najbolj enostavnega (PSP0) do najbolj popolnega (PSP3)

Vsaka verzija obsega več področij

Vsaka višja verzija vsebuje tudi vsa področja nižje verzije

Inženir napreduje postopoma:

- na začetku izbere proces PSP0
- ko ga obvlada, bo njegov cilj PSP0.1,
- ko obvlada tega napreduje proti PSP1 ...

Po analogiji CMM lahko verzije PSP razvrstimo v 4 nivoje zrelosti:

PSP0: Osnovni osebni proces razvoja PO

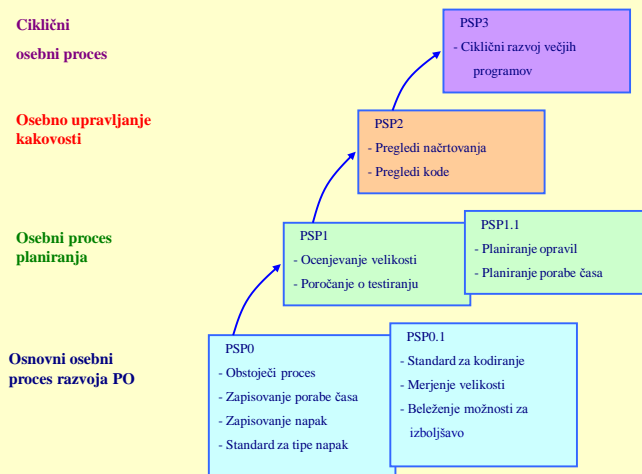
PSP1: Osebni proces planiranja

PSP2: Osebno upravljanje kakovosti

PSP3: Ciklični osebni proces

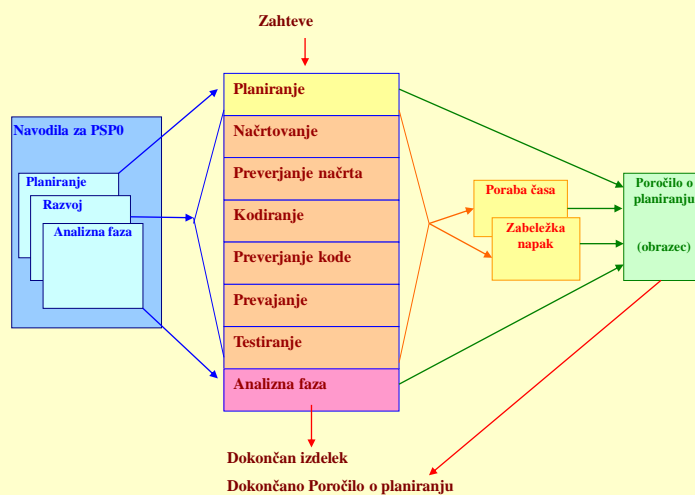
Nivoji zrelosti PSP

47



Struktura PSP2

48





Dokumenti in razmerje PSP – TSP - CMM

49

Različne verzije PSP-ja je definirana z različnim številom dokumentov:

- PSP0: 11 dokumentov (navodil, obrazcev, zapiskov in standardov)
- PSP1: 19 dokumentov
- PSP2: 25 dokumentov
- PSP3: 39 dokumentov.

PSP lahko z vidika posameznika nadomesti CMM:

- pokrita so vsa ključna področja nivojev 4 in 5 (kakovost!),
- manjkajo posamezna področja iz 2. in 3. nivoja (projekti, organizacija)

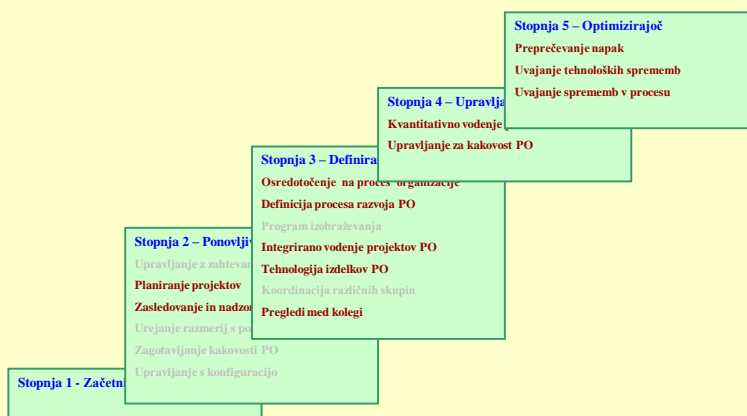
PSP - TSP:

- PSP je običajno uporabljen znotraj TSP-ja
- lahko namenjen projektom, pri katerih sodeluje en sam razvijalec
- če je razvijalcev več, potrebujemo vsaj TSP



PSP pokritost za ključna področja CMM

50





Razmerje PSP – agilnost

51

PSP ni omejen na določeno vrsto aktivnosti, metodo dela ali uporabo tehnologije

Z agilnimi principi dobro sovпада:

- PSP preskriptivni, agilnost deskriptivna
- Omogoča učinkovito doseganje ciljev ekipe:
 - Definiranje ciljev in standardov
 - Ocenjevanje in spremljanje opravljenega dela
 - Določanje realističnih planov, ki se jih je mogoče držati
 - Planiranje izboljšav
 - Delitev odgovornosti razvijalcem
 - Kreativno okolje

Ključna razlika: dokumentiranje



Izvajanje PSP pri TiK

52

Za 1. domačo nalogo:

- Privzamemo uporabo PSP1
- Sproti merimo: porabo časa, velikost izdelkov in število napak
- Podatke beležimo v ustrezne dokumente in povzamemo v dokumentu Projektni plan

Za 2. domačo nalogo:

- Privzamemo PSP2
- Uporabimo projektni plan za ocenjevanje
- Sproti merimo: porabo časa, velikost izdelkov in število napak
- Podatke zopet beležimo v ustrezne dokumente in povzamemo v dokumentu Projektni plan

V nadaljevanju izgled ključnih dokumentov

Vse podrobnosti bodo predstavljene na vajah!



PSP: Beležka porabe časa

53

Andrej Novak

Datum začetka: 05.03.2018

Namen: Študij predmeta RPS

Zap.št.zapisa: 1. teden

Datum	Začetek	Konec	Prekinitev	Čas	Aktivnost	Opis	Z	E
05.03.	10:15	12:00		105	Lab.vaje	Oracle potral 2: Uvod	x	2
	19:10	20:02		52	Branje	Fenton: SW Metrics (1.pog)	x	22
	20:02	21:30	14	74	Program	Prijava v testni portal, uporaba		
07.03.	8:17	8:33		16	Branje	Ponovitev snovi pret.tedna (zvezek)		
	11:15	12:00		45	Av.vaje	Avditorne vaje – uvod v portal	x	1
	19:31	20:42	7	64	Branje	Ponovitev snovi pret.tedna (zvezek)	x	13
08.03.	7:30	10:00	15	135	Predavanje	Merjenje v TPO	x	3
	18:36	20:52	12+30+7	87	Branje	Fenton: SW Metrics (2.pog)	x	23
10.03.	10:10	11:53	15*	88	Internet	SEI: CMM	x	2



PSP: Tedenska poraba časa po kategorijah

54

Ime: Andrej Novak

Datum začetka: 05.03.2018

Namen: Študij predmeta RPS

Zap.št.tedna: 1. teden

Dan\Kat	Pred.	A.vaje	L.vaje	Branje	Račun	Program	Internet	SKUPAJ
PON,5.3.			105	52		74		231
TOR,6.3.								
SRE,7.3.		45		80				125
ČET,8.3.	135			87				222
PET,9.3.								
SOB,10.3.							88	88
NED,11.3.								
SKUPAJ	135	45	105	219		74	88	666



PSP: Seštevek časov tekočega tedna

55

Ime: Andrej Novak
Namen: Študij predmeta RPS

Datum začetka: 05.03.2018
Zap.št.tedna: 1. teden

SEŠTEVEK ČASOV TEKOČEGA TEDNA

Kat	Pred.	A.vaje	L.vaje	Branje	Račun	Program	Internet	SKUPAJ
SKUPAJ	135	45	105	219		74	88	666
POVPR.	135	45	105	219		74	88	666
MAX.	135	45	105	219		74	88	666
MIN.	135	45	105	219		74	88	666

Skupno število tednov: 1



PSP: Tedenska poraba časa po kategorijah

56

Ime: Andrej Novak
Namen: Študij predmeta RPS

Datum začetka: 12.03.2018
Zap.št.tedna: 2. teden

Dan\Kat	Pred.	A.vaje	L.vaje	Branje	Račun	Program	Internet	SKUPAJ
PON,12.3.			105	11				116
TOR,13.3.							74	74
SRE,14.3.		45		40				85
ČET,15.3.	135							135
PET,16.3.								
SOB,17.3.					42			42
NED,18.3.								
SKUPAJ	135	45	105	51	42		74	452



PSP: Seštevek časov prejšnjega tedna

57

Ime: Andrej Novak
Namen: Študij predmeta RPS

Datum začetka: 12.03.2018
Zap.št.tedna: 2. teden

SEŠTEVEK ČASOV PREJŠNJEGA TEDNA

Kat	Pred.	A.vaje	L.vaje	Branje	Račun	Program	Internet	SKUPAJ
SKUPAJ	135	45	105	219		74	88	666
POVPR.	135	45	105	219		74	88	666
MAX.	135	45	105	219		74	88	666
MIN.	135	45	105	219		74	88	666

Skupno število tednov: 1



PSP: Seštevek časov tekočega tedna

58

Ime: Andrej Novak
Namen: Študij predmeta RPS

Datum začetka: 12.03.2018
Zap.št. tedna: 2. teden

SEŠTEVEK ČASOV TEKOČEGA TEDNA

Kat	Pred.	A.vaje	L.vaje	Branje	Račun	Program	Internet	SKUPAJ
SKUPAJ	270	90	210	270	42	74	162	1118
POVPR.	135	45	105	135	21	37	81	559
MAX.	135	45	105	219	42	74	88	666
MIN.	135	45	105	51			74	452

Skupno število tednov: 2



PSP: Beležka opravil

59

Ime: Andrej Novak

Datum začetka: 05.03.2018

# Opr	Datum	Vrsta	Ocena		Izmerjeno			Doslej				
			Čas	Enot	Čas	Enot	Raz	Čas	Enot	Raz	Max	Min
1.	5.3.	branje	44	22	52	22	2.36	52	22	2.36	2.36	2.36
Opis: Fenton: SW Metrics (1.pog) (minut na stran)												
2.	5.3.	prog.	100	100								
Opis: Prijava v testni portal, uporaba (minut na ekvivalent vrstice kode)												
3.	7.3.	branje	31	13	80	13	6.15	132	35	3.77	6.15	2.36
Opis: Ponovitev snovi pret.tedna (zvezek)												
4.	7.3.	branje	87	23	97	23	4.22	229	58	3.95	6.15	2.36
Opis: Fenton: SW Metrics (2.pog)												



PSP: Beležka napak

60

Ime: Andrej Novak

Datum: 05.03.2018

Št.prog: 9

Datum	Št.nap.	Kategorija	Faza vnosa	Faza odprave	Čas odprave	Št.pop.nap.
5.3.	1	20	kodiranje	prevajanje	2	
Opis: manjkajoče ; # 21						
7.3.	2	40	načrtovanje	prevajanje	8	
Opis: napačen tip (int namesto float) – spremenljivka a # 5						
7.3.	3	20	kodiranje	prevajanje	1	
Opis: napačno ime spremenljivke (velike – male črke) – spremenljivka stLeta # 9						
10.3.	4					
Opis:						



PSP: Kategorije napak

61

Kategorija	Naziv	Opis
10	Dokumentacija	komentarji, sporočila, opis delov programa
20	Sintaksa	tipkarske napake, ločila, format izpisa podatkov
30	Izgradnja	obvladovanje zahtev, moduli, knjižnice, verzije
40	Prirejanje	deklaracije, podvojena imena, veljavnost pravil, omejitve
50	Vmesnik	klici podprogramov, V/I, format podatkov
60	Preverjanje	obvladovanje napak, sporočila, pomankljivo preverjanje
70	Podatki	struktura, vsebina
80	Funkcije	logika, kazalci, zanke, rekurzija, izračuni, fun. napake
90	Sistem	konfiguracija, časovna razporeditev, spomin
100	Okolje	načrt, prevajanje, testiranje, podpora sistemu

© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



PSP: Celotni projektni plan

62

PROJEKTI PLAN (verzija 7) – 1/2

Izvajalec: **ANDREJ NOVAK**

Program: **PROGRAM 9**

Vodja: **ROŽANC**

Datum: **5. 3. 2018**

Zap.št: **29**

Pr.jezik: **JAVA**

SKUPAJ:	Plan:	Dej.vrednost:	Skupaj doslej:	
Min/LOC:	6.64	5.65	6.16	
LOC/Uro:	9.04	10.63	9.74	
Št.napak/KLOC:	56.91	38.46	47.92	
Delež odprav. napak:	25.00*	66.67*	39.13*	
Razmerje I/O napak:	0.185	0.406	0.275	
VELIKOST (LOC):	Plan:	Dej. vrednost:	Skupaj doslej:	
Skupaj (nove+spr.):	205	234	480	
Največja velikost:	294			
Najmanjša velikost:	149			
ČAS RAZVOJA (min):	Plan:	Dej. vrednost:	Skupaj doslej:	Doslej %:
Planiranje	97	90	206	6.97
Načrtovanje	112	120	254	8.60
Kodiranje	828	810	1804	61.05
Pregled kode	37	65	110	3.72
Prevajanje	57	50	118	3.99
Testiranje	143	110	282	9.54
Analiza	87	76	181	6.13
Skupaj:	1361	1321	2955	100.00
Največji čas:	1952			
Najmanjši čas:	989			



PSP: Celoten projektni plan

63

PROJEKTNI PLAN (verzija 7) – 2/2

Izvajalec: **ANDREJ NOVAK**

Datum: **5.3. 2018**

Program: **PROGRAM 9**

Zap.št: **29**

Vodja: **ROŽANC**

Pr.jezik: **JAVA**

VNEŠENE NAPAKE:	Plan:	Dej. vrednost:	Skupaj doslej:	Doslej %:	Doslej Nap./Uro:
Planiranje				0.00	
Načrtovanje	3	4	7	30.43	1.65*
Kodiranje	9	5	16	69.57	0.53*
Pregled kode				0.00	
Prevajanje				0.00	
Testiranje				0.00	
Analiza				0.00	
Skupaj:	12	9	23	100.00	
ODPRAVLJENE NAPAKE:	Plan:	Dej. vrednost:	Skupaj doslej:	Doslej %:	Doslej Nap./Uro:
Planiranje				0.00	
Načrtovanje				0.00	
Kodiranje				0.00	
Pregled kode	3(2)	6	9	39.13	4.91*
Prevajanje	7	2	10	43.48	5.08*
Testiranje	2(3)	1	4	17.39	0.85*
Analiza				0.00	
Skupaj:	12	9	23	100.00	

© Igor Rožanc

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko



Literatura

64

1. Watts S. Humphrey: A Discipline for Software Process, Carnegie Mellon University, Addison-Wesley, 1995.
2. Watts S. Humphrey: Introduction to the Personal Software Process, Carnegie Mellon University, Addison-Wesley, 1997.
3. Domača stran CMMI: www.cmmiinstitute.com
4. Domača stran TSP: www.sei.cmu.edu/tsp
5. Domača stran PSP: www.sei.cmu.edu/psp

© Igor Rožanc

informatiko

Univerza v Ljubljani, Fakulteta za računalništvo in