

# ZNANJE IN MANAGEMENT ZNANJA V PODJETJU

20.7.2011

Management znanja v procesu razvoja  
programske opreme

Avtor: Aleš Zebec, managing partner/IT/KMO  
dipl. ekon. (UN), ing. rač. in inf., MCSD



BuyITC inovativne internet rešitve d.o.o.  
Prvomajska ulica 30, 2000 Maribor, Slovenija  
Telefon: (02) 461 0 461  
Telefax: (02) 4 624 625  
[www.buyitc.si](http://www.buyitc.si)  
info@buyitc.si

**Kazalo**

<b>1</b>	<b>UVOD</b> .....	<b>2</b>
<b>2</b>	<b>RAZVOJ PROGRAMSKE OPREME: TEŽAVE IN REŠITVE</b> .....	<b>3</b>
2.1	Težave .....	3
2.2	Rešitve .....	3
<b>3</b>	<b>MANAGEMENT ZNANJA</b> .....	<b>5</b>
3.1	Znanje .....	6
3.2	Paradigma izboljšanja kakovosti .....	8
3.3	Tovarna izkušenj .....	9
3.4	Implikacije tovarne izkušenj .....	11
3.5	Primeri tovarn izkušenj .....	11
3.6	Primeri paketov izkušenj .....	12
3.7	Model managementa z znanjem .....	13
<b>4</b>	<b>MANAGEMENT ZNANJA V RAZVOJU PROGRAMSKE OPREME</b> .....	<b>15</b>
4.1	Tehnologije managementa znanja in programskega inženiringa .....	15
4.2	Študije primerov managementa znanja v procesu razvoja programske opreme .....	15
4.2.1	NASA Software Engineering Laboratory (SEL) .....	15
4.2.2	Daimler .....	16
4.2.3	ICL High Performance Systems .....	16
4.2.4	Telenor Telecom Software .....	17
<b>5</b>	<b>MANAGEMENT ZNANJA V PODJETJU BUYITC</b> .....	<b>18</b>
<b>6</b>	<b>KLJUČNI DEJAVNIK USPEHA</b> .....	<b>20</b>
6.1	Strategija .....	20
6.2	Procesi .....	20
6.3	Programske rešitve .....	20
6.4	Rezultati .....	21
<b>7</b>	<b>SKLEP</b> .....	<b>22</b>
<b>8</b>	<b>LITERATURA IN VIRI</b> .....	<b>23</b>

**Kazalo slik**

Slika 1: SECI model .....	7
Slika 2: koraki paradigme izboljšanja kakovosti .....	9
Slika 3: model tovarne izkušenj .....	10
Slika 4: model komponent managementa znanja v organizaciji (sistemi in program) .....	13
Slika 5: primer vmesnika za vizualizacijo znanja in izkušenj sistema BuyITC KM .....	19
Slika 6: pregled aktivnosti organizacij in uporaba strategij .....	20
Slika 7: implikacije uvedbe managementa znanja .....	21

## 1 UVOD

Organizacije se večinoma zavedajo, da je glavna gonilna sila znanje zaposlenih oziroma intelektualni kapital. Glavna težava z intelektualnim kapitalom je v tem, da je vezan na zaposlene, ki prihajajo in odhajajo. Hkrati prihajajo in odhajajo tudi izkušnje. Organizacije so tako postavljene pred izziv vzdrževanja nivoja kompetenc, potrebnih za konkurenčno poslovanje.

Management znanja in izkušenj sta ključni sredstvi za sistematičen razvoj in proces izboljšav programske opreme. Na področju programskega inženiringa (angl. *Software Engineering – SE*) predstavlja kvaliteta še vedno eno pglavitnih skrbi. Rešitve kot so programski jeziki četrte generacije, strukturne tehnike in objektno orientirane tehnologije so bistveno pripomogle k reševanju, vendar kljub temu težava ostaja. Management znanja (angl. *Knowledge Management – KM*) daje organizacijam priložnost, da zajamejo celotno kompleksnost in širino, ki je povezana z razvojem programske opreme.

Namen naloge je pregled literature iz področja managementa znanja v procesu razvoja programske opreme. Cilj naloge je s pomočjo pregleda študij primerov ugotoviti različne pristope k managementu znanja in implikacije uporabe. Opredeliti napore podjetja BuyITC, v katerem delam, glede managementa znanja in izvesti komparativno analizo s podatki iz študij primerov. Skozi nalogo bom preveril ali drži trditev, da osredotočenje na uporabo in ponovno uporabo izkušenj vpliva na proces razvoja programske opreme iz poslovnega, organizacijskega in vidika zaposlenih. Konkretno bom poskusil odgovoriti na naslednja vprašanja:

- a.) Ali management znanja v procesu razvoja izboljša kvaliteto programske opreme?
- b.) Ali zniža stroške razvoja?
- c.) Ali izboljša položaj zaposlenega v organizaciji?

Prvo poglavje opisuje problematiko, ki narekuje uporabo managementa znanja v procesu razvoja programske opreme, težave v razvoju in možne rešitve.

V drugem poglavju naloge povzema osnovne definicije znanja, management znanja, teoretične modele in strategije. Opisuje eno od implementacij koncepta managementa znanja, tovarno izkušenj (angl. *The Experience Factory*) in model managementa znanja, ki se uporablja v organizacijah. Tretje poglavje predstavlja primere implementacij managementa znanja v procesu razvoja programske opreme in povzetke pregleda študij primerov. Opisuje tudi napore uvedbe managementa znanja v podjetju BuyITC. Četrto poglavje opredeljuje ključne dejavnike uspeha implementacije managementa znanja skozi komparativno analizo posameznih študij primerov in obravnavanega podjetja. Sklep povzame vsebino seminarske naloge in ugotovitve, ki zagotavljajo dolgoročen uspeh uvedbe managementa znanja v organizaciji.

## 2 RAZVOJ PROGRAMSKE OPREME: TEŽAVE IN REŠITVE

Programski inženiring (angl. *Software Engineering – SE*) obsega razvoj in vzdrževanje programske opreme. Definicija, kot jo navaja Sommerville, opredeljuje programski inženiring kot inženirsko disciplino katere fokus je stroškovno učinkovit razvoj visoko kvalitetnih programskih sistemov. Programska oprema je abstraktna in neoprijemljiva. Ni omejena z materiali, fizikalnimi zakoni ali procesom proizvodnje. Potencial programske opreme tako ni omejen. Vendar pomanjkanje naravnih omejitev pomeni, da programska oprema hitro postane kompleksna in tako težko razumljiva (Sommerville, 2006).

### 2.1 Težave

Razvoj programske opreme je velikokrat težaven. Poročilo, ki ga izdaja The Standish Group glede projektov razvoja programske opreme navaja, da je uspešnih samo 32% vseh projektov (izdelani v času, stroškovnih okvirjih in zahtevanimi funkcionalnostmi), 44% imajo težave (zamujejo, večji stroški in omejene funkcionalnosti) in 24% propadlih (ustavljeni pred končanjem ali nameščeni, vendar neuporabljeni). 52,7% projektov bo za 189% dražjih, kot definirani izhodiščni stroški ("CHAOS Summary 2009 Report," 2009). Stroški teh težav so le del celotnih stroškov, ki vključujejo tudi oportunitetne stroške.

Zakaj torej obstaja toliko težav pri razvoju programske opreme? Programska oprema je abstraktna, kar pomeni, da težko zagotovimo pregled nad celotnim sistemom, ki lahko vsebuje več 100.000 vrstic programske kode. Manjše napake v kodi lahko imajo velike posledice v celotnem sistemu.

Druga večja težava je komunikacija med končnim uporabnikom in razvijalci. Predvsem v delovnih okoljih, kjer odprava napak traja dalj časa in je težko predvideti roke končanja, s tem pa se onemogoča projektni management.

Kljub večjemu številu obstoječih težav pa obstajajo projekti, ki zagotavljajo uporabno in delujočo programsko opremo. Robert Glass nasprotuje slabim statistikam in trdi, da so neuspeli projekti prej izjema kot pravilo. Glavni razlog za trditev navaja, da se težave s programsko opremo uporabljajo kot motivacija za raziskave in analize, ki so pogosto objavljene in dajejo splošno sliko o neuspešnih projektih. Glede na njegove ocene se neuspehi gibljejo med 10 in 15% (Glass, 2005).

Glede na pregled literature ugotavljam, da se pojavlja veliko objav o neuspešnih projektih, bistveno več, kot o uspešnih. Vendar klub tem razlikam, poročilo CHAOS opredeljuje 68% projektov s težavami, kar nakazuje, da je na tem področju veliko možnosti izboljšav.

### 2.2 Rešitve

Na temo predlaganih rešitev je bilo veliko razprav, vendar končna rešitev ne obstaja. Obstaja več rešitev za izboljšanje načina razvoja, samega procesa razvoja, novi programski jeziki in ostala razvojna orodja. Cilj teh izboljšav je po navadi dvig produktivnosti in kvalitete razvite programske opreme. Večino teh rešitev je povzel Robert Glass in jih razvrstil glede na vpliv (Glass, 1999):

- a.) Strukturne tehnike: s strukturno analizo, načrtovanjem in programiranjem.

- b.) Četrta generacija programskih jezikov (angl. *Fourth generation programming languages - 4GL*).
- c.) Orodja CASE (angl. *Computer Aided Software Engineering – CASE*): orodja za podporo razvoju programske opreme, predvsem pri analizi in načrtovanju.
- d.) Formalne metode: formalne specifikacije in preverjanje programske opreme.
- e.) Cleanroom metodologije: metode za odpravo napak programske opreme.
- f.) Procesni modeli: opisi ustreznih procesov za razvoj programske opreme.
- g.) Objektno usmerjene tehnologije: opredeliti problem s pomočjo objektov in z njimi sestaviti programsko rešitev.

Večina tehnologij kaže obetavne rezultate, vendar obstaja relativno malo znanstvenih člankov za dejansko oceno vpliva različnih metod. Velikokrat literatura navaja izboljšave, ki pa ne izvirajo iz uporabe novih tehnologij, temveč iz niza ostalih sprememb v organizaciji, uporabe programskih jezikov itd.

### 3 MANAGEMENT ZNANJA

Pri preučevanju managementa znanja gre za interdisciplinaren pristop, ki zajema področja informacijskih znanosti, psihologije, sociologije itd. Obstaja več avtorjev in različnih definicij ravnanja z znanjem, ki se ločijo predvsem glede na vidike in pristope: celovit, ravnanja človeških virov, informacijskih sistemov ali strategije. Glede na dejstvo, da je management znanja multidisciplinarne narave, je najbolj primerna definicija, ki jo ponujata avtorja Davenport in Prusak (Davenport & Prusak, 1998), ki pravita, da management znanja črpa iz obstoječih virov organizacije, kot je informacijski sistem, sistem za management sprememb in praks pri ravnanju s človeškimi viri. Torej gre za planiranje, organiziranje, motiviranje in kontroliranje ljudi, procesov in sistemov organizacije za povečanje neotipljivega kapitala v obliki znanja in učinkovite uporabe.

Neotipljiv kapital v obliki znanja predstavljajo dokumenti v papirni ali elektronski obliki, kot so navodila, dokumentacija, znanje v obliki najboljših praks, ki opredeljujejo znanje zaposlenih, kako izvajati opravila, znanje v lasti skupin, ki iščejo rešitve specifičnih problemov ter znanje vgrajeno v izdelke, organizacije, procese in odnose. Povezan izraz je »organizacijski spomin«, ki sicer nima natančne definicije, je pa opredeljen kot intuitivnost organizacije, da vsebuje sled lastnih preteklih aktivnosti. Spomin opisuje kot enovito, monolitno skladišče celotne organizacije (Ackerman & Halverson, 2000). Razumevanje tega izraza velikokrat zajema proces zbiranja, uporabe in skladiščenja informacij.

Management znanja vključuje procese pridobivanja, ustvarjanje, izpopolnitve, hrambe, prenosa, distribucije in uporabe. Nadzira te procese, razvija podporne metodologije, sisteme in motivira ljudi k sodelovanju. Cilji je izboljšanje neotipljivega kapitala, kar omogoča boljše poslovne in prakse znanja, izboljšano organizacijsko vedenje, boljše odločitve in izboljšano organizacijsko uspešnost. Čeprav lahko določene procese izvajajo posamezniki je management znanja aktivnost celotne organizacije, ki nalaga managerjem odgovornost, da poskrbijo za pogoje, ki omogočajo doseganje ciljev, kot so: motivacija zaposlenih k sodelovanju pri doseganju ciljev, kako ustvariti socialne procese, ki bodo zagotovili uspeh.

Socialni procesi vključujejo skupnosti praks - samoorganizacijo skupin ljudi, ki imajo skupni interes, ustvariti strokovno mrežo, preko katere tisti z manj znanja pridejo v stik s tistimi z več strokovnega znanja in izkušnjami. Takšni socialni procesi so potrebni, ker znanje prvotno leži pri posamezniku, za uspešno managiranje znanja pa se mora prenašati preko socialnih skupin in mrež. Posledično ugotavljamo, da je management znanja tesno povezan z ljudmi in manj s tehnologijo, čeprav informacijske in komunikacijske tehnologije v učeči se organizaciji zagotavljajo ključno podporo pri razvoju znanja v organizaciji.

Cilji managementa znanja je podpreti inteligentno delovanje organizacije, ki zagotavlja preživetje in uspeh. Thomas Devenport obravnava management znanja kot metodo za poenostavitev procesa deljenja, distribucije, nastajanja, zajemanja in razumevanja znanja (Davenport, De Long, & Beers, 1998).

V osnovi lahko ločujemo dve osnovni strategiji managementa znanja (Hansen, 1999):

- Kodificiranje: sistematizacija in hramba informacij, ki predstavljajo znanje organizacije, ter zagotoviti enostaven dostop uporabnikom

- Personalizacija: za podporo pretoka informacij v organizaciji, ki zagotavlja shranjevanje informacij o virih znanja

Poudariti je potrebno, da strategija kodifikacije ne ustreza vsem vrstam znanja. Kadar je znanje odvisno od konteksta, ta pa je težko prenosljiv, je ponovna uporaba znanja nevarna brez predhodne kritične analize (Hansen, 1999).

V nadaljnji obravnavi managementa znanja v procesu razvoja programske opreme se bom osredotočil predvsem na dve omenjeni osnovni strategiji kodificiranja in personalizacije.

Naslednje poglavje obravnava znanje, ki je osnovni gradnik opisanih konceptov. Kot opisano v prejšnjem poglavju je bistvena težava procesa razvoja programske opreme zagotavljanje kakovosti. V nadaljevanju opisujem koncepte izboljšanja kakovosti, ki izvirajo iz strategije razvijanja kakovosti, bolj natančno doseganja celovite kakovosti (angl. *Total Quality Management – TQM*). Na podlagi celovite kakovosti je v programskem inženiringu razvita paradigma izboljšanja kakovosti, na kateri bazira tovarna izkušenj. Tovarna izkušenj je ogrodje za zagotavljanje ponovne uporaba izkušenj, procesov in produktov v razvoju programske opreme. Ogrodje je namenjeno zbiranju, pakiranju in hrambi izkušenj, ki nastajajo v okviru projektov razvoja programske opreme. Pakiranje vključuje posploševanje, prilagajanje in formaliziranje izkušenj za kasnejšo uporabo.

Opisane koncepte združuje celovit model managementa znanja opisan v nadaljevanju.

### 3.1 Znanje

Po navadi opredelimo znanje kot zaporedje, ki se prične s podatki, informacijami in s postavitvijo teh v kontekst dobimo znanje. Znanje je pogosto opredeljeno tudi kot osebno prepričanje.

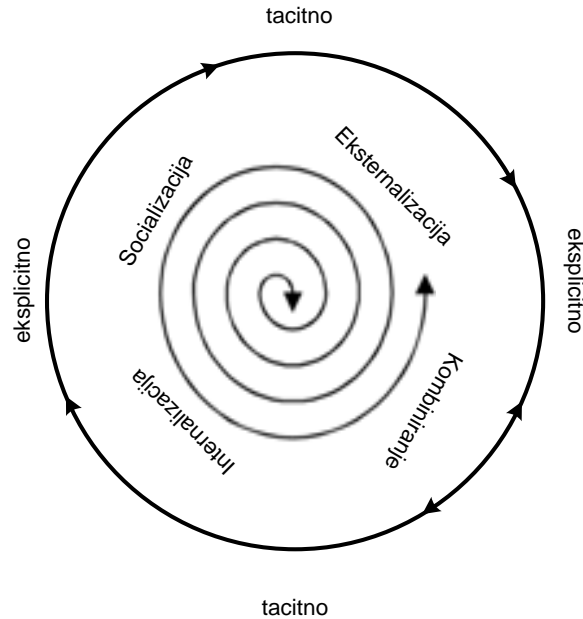
Obstajajo različne klasifikacije znanja, najbolj temeljni sta tiho ali prikrito in eksplicitno znanje.

**Prikrito** ali tacitno znanje se nahaja v glavah ljudi in ga je nemogoča ali težko oblikovati. Večina znanja je prvotno v tej obliki. Oblikuje se skozi čas na podlagi poizkušanja in ni učinkovito izrabljeno. Organizacija tako v bistvu ne ve kakšno znanje ima. Nekatera znanja so vgrajena v poslovnih procesih, dejavnostih in odnosih, ki so bili ustvarjeni v času izvajanja izboljšav.

**Eksplicitno** znanje obstaja v obliki besed, stavkov, dokumentov, urejenih podatkov, računalniških programov in drugih eksplicitnih oblik. Tako lahko opredelimo temeljni problem ravnanja z znanjem, ki je poiskati in oblikovati prikrito znanje v obliko, da ga bodo lahko uporabljali drugi.

Nonaka in Takeuchi navajata, da je tacitno znanje preko eksternalizacije možno konvertirati v eksplicitno in obratno skozi internalizacijo. Skozi socializacijo nastaja in se prenaša tacitno znanje, eksplicitno pa nastaja s kombinacijo (Nonaka & Toyama, 2003).

SLIKA 1: SECI MODEL



Vir: Nonaka, I., & Toyama, R. (2003). *The knowledge-creating theory revisited: knowledge creation as a synthesizing process. Knowledge Management Research & Practice, str. 2*

Znanje lahko delimo glede na ravni “vedeti nekaj”, “vedeti kako” in “vedeti zakaj”.

“**Vedeti nekaj**” znanje določa, kako posameznik ukrepa, če se sooča z določeno množico dražljajev. Na primer: prodajalec, ki je usposobljen za svetovanje kateri izdelek je najbolj primeren za različno uporabo ima “vedeti nekaj” raven znanja.

Naslednjo višja raven znanja je “**vedeti kako**” se odločiti v primeru določenega odziva ali dražljajev. Takšno znanje je potrebno, ko v primeru preprostih programiranih odnosov med dražljaji in odzivi, znanje “vedeti nekaj” ni zadovoljivo. “Vedeti kako” znanje omogoča posamezniku, da ugotovi, kateri način in pristop je najboljši, tudi v primeru motenj.

Najvišja raven znanja je “**vedeti zakaj**”. Na tej ravni, posameznik poglobljeno razume vzročne odnose, interaktivne učinke in stopnje negotovosti, povezane z opazovanimi dražljaji ali simptomi. To običajno vključuje razumevanje osnovnih teorij in vrsto izkušenj iz določenega področja.

Z znanjem se velikokrat povezujejo tudi izkušnje in informacije. Večina dojema izkušnje kot nekaj kar smo pridobili skozi prakso. Informacije pa kot nekaj, kar nam je bilo povedano.

Informacije v informatiki opredeljujemo kot podatke, ki imajo pomen. Znanje pa kot informacije, ki jih uporabljamo v računalniškem sistemu. Področje umetne inteligence (angl. *Artificial Intelligence*) opredeljuje razmerje med podatki, informacijami in znanjem skozi reševanje problemov, kjer lahko gledamo na posamezno rešitev kot podatek, informacijo ali znanje, v odvisnosti od vloge, ki jo ima rešitev pri sprejemanju odločitev in učenju iz izkušenj (Aamodt & Nygård, 1995). Kritiki definicije navajajo, da znanje zahteva tudi procese, ki niso del računalniških sistemov in so del okolja v katerem znanje nastaja. Bolj pragmatično definicijo podaja Taylor, ki informacije deli na instrumentalne in informacije dejstev, kjer so instrumentalne tiste, ki



posamezniku povedo kako nekaj narediti, informacije dejstev, pa zagotavljajo dejstva (Taylor, 1991).

### 3.2 Paradigma izboljšanja kakovosti

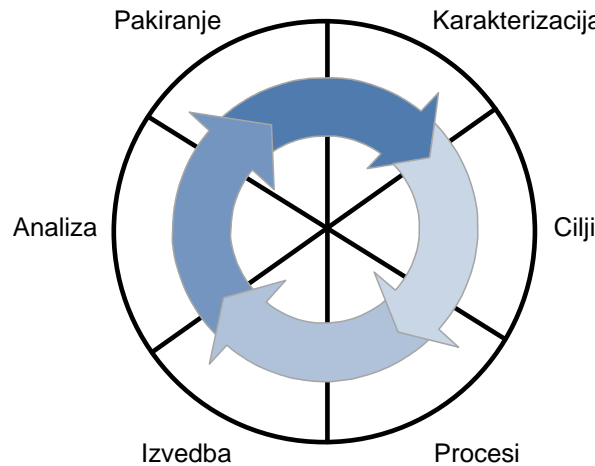
Glede na poslovne potrebe razvoja programske opreme, programski inženiring zagotavlja okvir, ki temelji na evlucijski paradigmi zagotavljanja kakovosti, prirejeni potrebam področja razvoja programske opreme z imenom paradigma izboljšanja kakovosti. Paradigmo podpira orodje za vzpostavitev projekta, poslovnih ciljev, mehanizmov merjenja, doseganja ciljev in organizacijski pristop za zagotavljanje kompetenc potrebnih za razvoj programske opreme, imenovan organizacija tovarne izkušenj (angl. *Experience Factory Organization*).

Paradigmo izboljšanja kakovosti je razvil Basili, et al. in je rezultat uporabe znanstvenih metod pri reševanju problema zagotavljanja kvalitete v procesu razvoja programske opreme. Posledično je paradigma povezana s Shewart-Deming ciklom načrtuj/izvedi/preveri/spremeni, ki je široko uporabljen za implementacijo zagotavljanja kakovosti (Deming, 2000).

Paradigmo izboljšanja kakovosti opišemo z naslednjimi šestimi koraki:

1. Karakterizacija: razumevanje okolja na podlagi modelov, podatkov intuicije, itd. Vzpostavitev izhodišč glede na obstoječe poslovne procese v organizaciji in opredelitev njihove samokritičnosti glede izvajanja.
2. Cilji: na podlagi osnovne karakterizacije in glede na zmožnosti, ki so strateško pomembne za organizacijo, določimo merljive cilje za uspešne projekte, učinkovitost in izboljšave organizacije. Razumna pričakovanja se oblikujejo glede na izhodišča opredeljena v predhodni fazi.
3. Procesi: na podlagi karakterizacije in ciljev izvedemo primerne procese izboljšav, podporne metode in orodja, ki morajo biti skladna z zastavljenimi cilji.
4. Izvedba: izvedemo procese produkcije in spremljamo ter zbiramo podatke, glede na merila zastavljena s cilji.
5. Analiza: na koncu specifičnega projekta se izvede analiza zbranih podatkov in informacij za oceno trenutnih praks, identifikacije težav, beleženja dognanj in pripravo predlogov za izboljšave v prihodnosti.
6. Paket: združiti izkušnje, pridobljene v obliki novih ali posodobljenih modelov ali oblik strukturiranega znanja, ki so bila pridobljena iz preteklih projektov in hramba v bazi izkušenj, za potrebe projektov v prihodnosti.

SLIKA 2: KORAKI PARADIGME IZBOLJŠANJA KAKOVOSTI



Vir: Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). *The Experience Factory: Encyclopedia of Software Engineering*, str. 5

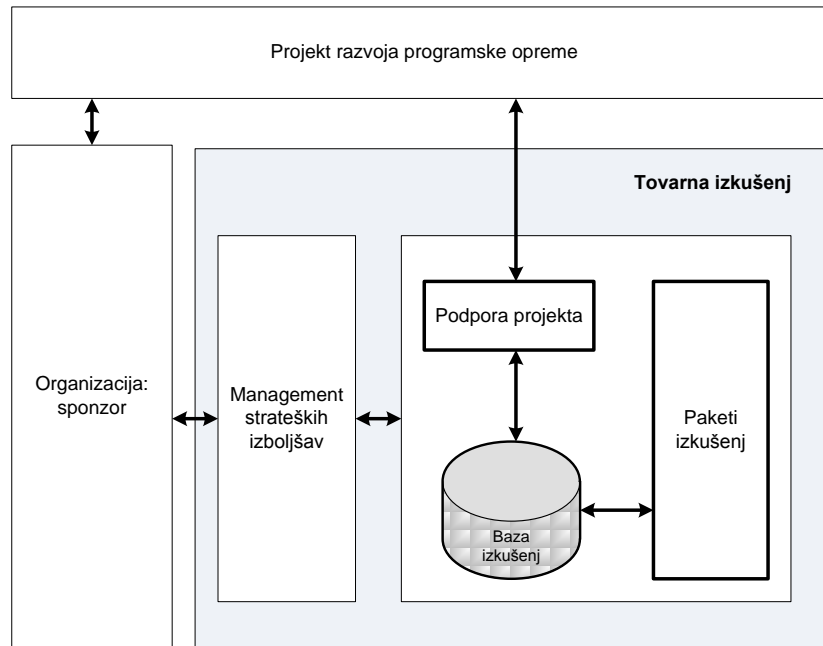
Paradigma izboljšanja kakovosti vsebuje dva cikla povratnih informacij:

- Cikel povratnih informacij projekta (kontrolni cikel) nudi povratne informacije glede projekta v fazi izvedbe: ne glede na cilje organizacije, mora pilotni projekt uporabljati vire glede na potrebe; zato so pomembni kvantitativni kazalci na nivoju projekta in naloge.
- Cikel povratnih informacij organizacije (kapitalizacijski cikel) vključuje povratne informacije glede delovanja celotne organizacije in ima dva namena:
  - zagotavljanje analitičnih informacij glede uspešnosti projektov za čas dokončanja, kjer primerja podatke projekta z nominalnim obsegom, ki ga je določila organizacija, analizira skladnost in odstopanja
  - akumulacija izkušenj za ponovno uporabo v obliki programske opreme ali artefaktov, ki jih lahko uporabimo za druge projekte in so izboljšane na podlagi opravljene analize

### 3.3 Tovarna izkušenj

Eden on načinov management znanja je, da predamo odgovornost za zbiranje in ponovno rabo izkušenj ločenemu delu organizacije. To je osnovna ideja tovarne izkušenj (angl. *Experience Factory*), ki predstavlja socio-tehnološko infrastrukturo za management znanja, za ponovno uporabo izkušenj pridobljenih v življenjskem ciklu, procesih in produktih (Basili, Caldiera, & Rombach, 1994). Izkušnje se zbirajo na projektih razvoja programske opreme, se pakirajo in hranijo v bazi izkušenj.

SLIKA 3: MODEL TOVARNE IZKUŠENJ



Vir: Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). *The Experience Factory: Encyclopedia of Software Engineering*, str. 9

Tovarna izkušenj je del paradigme izboljšanja kakovosti. Vključuje povratno zanko za pobude izboljšav, ki je definirana s šestimi koraki, definiranimi v poglavju paradigme izboljšanja kakovosti. Zadnji korak predstavlja različne oblike paketov izkušenj:

- Produktni paket: vsebuje informacije o življenjskem ciklu produkta, kako te informacije uporabiti in kaj se lahko iz njih naučimo.
- Procesni paket: vsebuje informacije o izvedbi procesa življenjskega cikla in ponovni uporabi.
- Paket razmerij: uporablja se za analize in napovedi. Vključuje stroškovne modele in modele napak in virov.
- Paket orodij: navodila za uporabo orodij in izkušnje glede uporabe.
- Management paket: referenčne informacije za projektne managerje.
- Podatkovni paketi: podatki pomembni za projekt razvoja programske opreme ali povezanih aktivnosti. Vključuje projektne podatkovne baze ali zapise glede kvalitete.

Organizacija tovarne izkušenj tako pripomore k razvoju nove programske opreme z zbranimi izkušnjami in predlaga izboljšave glede procesov, ki delujejo na osnovi zbranih izkušenj. Ta koncept imenujemo management strateških izboljšav, kot je razvidno na sliki 3. Slika prikazuje tudi interakcijo med tovarno izkušenj, sponzorjem (organizacijo) in projektom razvoja programske opreme.

### 3.4 Implikacije tovarne izkušenj

Tovarna izkušenj ponuja organizacijsko strukturo, ki ločuje fokus razvoja produktov in učenja. Podpira učenje in ponovno rabo znanja z generiranjem oprijemljivih sredstev v obliki paketov izkušenj. Povezuje raziskovanje z razvojem.

Tovarna izkušenj se uporablja za utrjevanje in integracijo aktivnosti, kot je pakiranje izkušenj, svetovanje, zagotavljanje kvalitete, izobraževanje, usposabljanje, procese, podporna orodja, ocenjevanje in meritve.

Tovarna izkušenj omogoča večjo izrabo in učinkovitost obstoječih tehnologij. Raziskave usmerja k poslovnim potrebam in prenosu tehnologij. Povezana področja raziskav so definicija in prilagoditve modelov, integracij tehnologij, širša uporaba tehnik in metod, gradnja in uporaba baze izkušenj ter avtomatizacija. Omogoča večjo učinkovitost uporabe formalne izobrazbe, modelov in notacij. Zahteva izobraževanje v tehnologijah preverjanja, formalnih zahtevah, notacijah za specifikacije, formalnih modelih za meritve in management ter ocenjevanju tehnologij.

Financiranje tovarne izkušenj je odvisno od organizacijske strukture. Vsekakor pa je smiselno ločiti tovarno izkušenj v lastno stroškovno mesto, ki je v začetku financirano s presežki poslovanja, kasneje pa jo financirajo projekti, ki jih podpira. Na podlagi obstoječih primerov uporabe tovarne izkušenj stroški zbiranja podatkov znašajo na projekt 5% (Basili, et al., 1994), kar pa se pokrije z izboljšavami na naslednjem projektu. Stroški samega delovanja tovarne izkušenj vključujejo minimalno dva zaposlena in so v veliki meri odvisni od števila podprtih projektov, stopnjo napora in nabora aktivnosti, kot je zagotavljanje kvalitete, izdelava orodij, izobraževanje in usposabljanje.

### 3.5 Primeri tovarn izkušenj

Organizacija, ki je že pred časom prepoznala vrednost akumulacije in ponovne uporabe izkušenj je NASA Goddard Space Flight Center, ki je leta 1977 ustanovil Software Engineering Laboratory. Software Engineering Laboratory (SEL) je danes napreden primer koncepta tovarne izkušenj. Paketi izkušenj, ki jih razvijajo v SEL so večinoma usmerjeni v projektni management ter nadzor, nabavo in prilaganje novih tehnologij za razvoj in vzdrževanje programske opreme. SEL je razvil več paketov izkušenj. Zanimivejši so:

- Software Management Environment (SME) - nabor podatkov, orodij in analitičnih tehnik namenjenih projektnemu management za nadzor izvajanja projekta in možnost primerjave s podobnimi projekti, identifikacija in analiza težav ter iskanje rešitev. Bistveni element paketa je spletna podatkovna baza za hrambo in dostop do podatkov programskega inženiringa.
- Drug zanimiv paket izkušenj je Cleanroom procesni model. V okviru tega paketa analitik pripravi zahteve, razvijalci pripravijo in preverijo zasnovo, kodirajo, izdelek pa je nato objavljen za test. Izvede se statistični test in po zadovoljivih rezultatih se rešitev objavi s certifikatom zanesljivosti.

Akumulacijo in uporabo izkušenj kot strateško usmeritev je sprejelo tudi več japonskih (Toshiba, Hitachi, NEC, Fujitsu) in ameriških podjetij. Japonska podjetja so prevzela strategijo, ki temelji na

manjših izboljšavah, ki so predstavljajo omejeno oblike tovarne izkušenj. Paketi izkušenj, ki jih uporabljajo so večinoma usmerjeni v: ponovno uporabo programske kode, modeliranje zanesljivosti, izboljšanje produktivnosti in izboljšanje kvalitete storitev.

### 3.6 Primeri paketov izkušenj

Tovarna izkušenj lahko pakira več oblik izkušenj. Lahko ustvarja izhodišča in modele virov, napak, sprememb, produktov, definicije procesov, modele, metode, tehnike, tehnike ocenjevanja, produkte in modele produktov, knjižnice znanja in različne modele zagotavljanja kvalitete.

Obstajajo različne oblike paketov izkušenj:

- Enačbe, ki definirajo relacije med spremenljivkami
- Histogrami ali strukturni krogi analiziranih podatkov
- Grafi, ki definirajo obsege norm
- Specifično znanje, ki je povezano z različnimi tipi projektov, faz projektov ali aktivnosti in v obliki analize tveganj in predlogo
- Modeli ali algoritmi, ki specificirajo procese, metode ali tehnike

Primeri modelov virov in izhodišč obsegajo stroškovne modele, modele alokacije virov, človeških virov, časovnice, računalniško podporo. Razlikujejo med viri in različnimi faktorji, ki vplivajo na vire. Pri pripravi modelov virov je tovarna izkušenj usmerjena v zajem podatkov povezanih z različnimi faktorji, ki so bili del preteklih projektov. Te relacije lahko uporabimo za enačbe, ki definirajo empirična razmerja med faktorji. Paketi razmerij povzemajo izkušnje podjetja glede na različne vire in opredelijo lastnosti faktorjev, ki vplivajo na vire.

Primeri modelov napak in sprememb vključujejo: spremembe izhodišč glede na klasifikacijo, modeli za napoved napak, modele zanesljivosti (Weiss & Basili, 1985).

Primerna analiza lahko zajame večje število napak, težav povezanih z različnimi fazami razvoja. Histogrami in Paretovi grafi so vgrajeni z analizo različnih oblik in skupin napak za lažje identifikacijo vzorcev ali razlik glede na različna okolja. Ti paketi zajemajo izkušnje glede napak skozi lastnosti različnih projektov. Uporabljajo se za napovedi, spremljanje, ocenjevanje in iskanje možnosti izboljšav.

Modeli lastnosti projektov in izhodišč, ki jih izdelajo tovarne izkušenj obsegajo: zgodovino rasti in sprememb za velikost, človeške vire, računalniško podporo, različne testne primere, pokrivanje s testiranjem itd. To zgodovino lahko primerjamo z normami, ki jih določa okolje ali podobni projekti. Na podlagi tega se razvijajo napovedi, ocene za aktualni projekt in smernice za nadaljnji razvoj. Tovarna izkušenj lahko za večino podatkov pripravi grafične predstavitve. Akumulacija podatkov skozi različne projekte omogoča izdelavo izhodišč za načrtovanje in spremljanje projektov.

Za izboljšave se mora organizacija osredotočiti na nove tehnologije. Izvajati mora eksperimente in beležiti ugotovitve v obliki izkušenj, ki se nato uporabijo za prilagajanje trenutnih procesov.

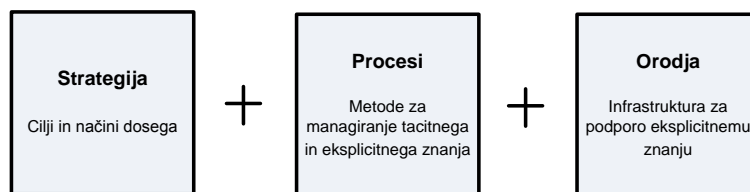
Glavni produkt tovarne izkušenj je paket izkušenj. Vsebina in struktura paketa se razlikuje glede na izkušnje, ki jih vključuje. Vsebujejo za zagotovo osrednji element preko katerega definiramo kaj paket predstavlja: produkt življenjskega cikla razvoja programske opreme ali procesa, matematično relacijo, empirični ali teoretični model, podatkovno bazo itd. Primeri paketov izkušenj:

- Produktni paketi: njihov osrednji element je produkt življenjskega cikla, združen z informacijami potrebnimi za ponovno uporabo. Primeri: programi, arhitekture, zasnove.
- Procesni paketi: osrednji element je proces življenjskega cikla, združen z informacijami potrebnimi za ponovno uporabo. Primer: procesni modeli, metode.
- Paketi razmerij: osrednji element je razmerje ali sistem razmerij med opazovanimi lastnosti in projektom razvoja programske opreme. Razmerja in relacije so lahko časovno pogojena. Uporabljajo se za analize in napovedi relevantnih situacij. Primeri: stroškovni modeli, modeli virov, modeli napak.
- Paketi orodij: osrednji element je specifično orodje. So konstruktiva npr. generator kode, orodje za management konfiguracij, ali analitična npr. regresijski tester, statistični analizator.
- Management paketi: osrednji element je referenčna informacija povezana z managementom projekta. Primeri: priročniki za vodenje projektov, modeli za podporo odločanju.
- Podatkovni paketi: osrednji element je zbirka definiranih in preverjenih podatkov, relevantnih za projekt razvoja programske opreme ali aktivnosti znotraj projekta. Primeri: projektne podatkovne baze, zapisi kvalitete.

### 3.7 Model managementa z znanjem

Analizo študij primerov, ki so opisana v nadaljevanju, bom izvedel na podlagi modela managementa znanja (sistemi ali programi), ki se uporablja v organizacijah. Model je predstavljen na naslednji sliki.

SLIKA 4: MODEL KOMPONENT MANAGEMENTA ZNANJA V ORGANIZACIJI (SISTEMI IN PROGRAM)



*Vir: Dingsøyr, T., & Conradi, R. (2002). A Survey of Case Studies of the Use of Knowledge Management in Software Engineering. International Journal of Software Engineering and Knowledge Engineering, str. 14*

Zapišemo lahko, da je model managementa znanja v organizacijah sestavljen iz treh delov:

- a.) Strategije managementa znanja, ki opredeljuje cilje in sredstva, načine za doseg ciljev. Ko govorimo o procesu razvoja programske opreme po navadi opisujemo cilje kot so znižanje stroškov razvoja in izboljšanje kakovosti. Lahko pa cilji vsebujejo tudi načine kako olajšati in poenostaviti delo inženirjev.

- b.) Procesov, ki vsebujejo aktivnosti omogočanja managementa znanja. Po navadi so to metode zbiranja in distribucije znanja, ali pa so aktivnosti ločenega dela organizacije, projektnih managerjev ali razvijalcev.
- c.) Orodja nastopajo kot programski sistem za podporo managementu znanja, kjer se znanje nahaja. Tega uporabljajo različne uporabniške skupine (razvijalci, projektni managerji, itd.). Znanje je predstavljeno kot podatkovna baza, spletne strani ali datoteke. Dodaten način predstavitve znanja v sistemu na način, da omogoča hitro iskanje relevantnih informacij, je sklepanje na osnovi primerov (angl. *Case-based reasoning - CBR*), pri katerem gre za reševanje aktualnih problemov na podlagi rešitev podobnih problemov v preteklosti. Znanje je dinamično, ki se s časom spreminja, tako morajo orodja z management znanja omogočati revizijo, brisanje in vpisa novega znanja v sistem.

## 4 MANAGEMENT ZNANJA V RAZVOJU PROGRAMSKE OPREME

Management znanja lahko glede na literaturo razdelimo v dve skupini: tehnične rešitve managementa znanja in vpliv managementa znanja na organizacijo. V nadaljevanju sta opisani obe skupini.

### 4.1 Tehnologije managementa znanja in programskega inženiringa

Za podporo managementu znanja in razvoja programske opreme obstaja veliko orodij. Obstajajo orodja kot je Experience Management System (Seaman, Mendonca, Basili, & Kim, 1999) ali sklepanje na osnovi primerov (CBR) za zadržanje in uporabo izkušenj. CBR je primeren tudi za ponovno uporabo izkušenj v programskem inženiringu. Obstajajo primeri implementacije idej tovarne izkušenj v obliki CBR sistema za izboljšavo sistemov razvoja izobraževalnih in drugih programskih rešitev. Podoben sistem uporablja tudi Hewlett Packard.

Eno od implementacij predstavlja tudi CODE4 (Conceptually Oriented Description Environment), ki je splošno namenski sistem za management znanja, namenjen analizi, razhroščevanju in uporabi znanja določene domene ali področja (Skuce & Lethbridge).

### 4.2 Študije primerov managementa znanja v procesu razvoja programske opreme

Dejanskih primerov uporabe management znanja je v literature malo. V nadaljevanju navajam 4 študije primerov uporabe managementa znanja v organizaciji. Sledi pregled uporabljenih konceptov posamezne študije in kakšne organizacijske aktivnosti so te organizacije izvajale.

#### 4.2.1 NASA Software Engineering Laboratory (SEL)

Prvo implementacijo tovarne izkušenj je izvedla NASA Software Engineering Laboratory (Seaman, et al., 1999). Izkušnje se zajemajo v obliki podatkov o stroških, procesih, informacije o metodologijah uporabljenih za vodenje projektov, orodjih in uporabljenih tehnologijah. Zajema tudi podatke o produktih, kot so informacije o spremembah, napakah, rezultate statističnih analiz objavljene kode, ki so bili nato uporabljeni za izdelavo modelov napovedi in izboljšanje procesa razvoja programske opreme.

Kot rezultat teh aktivnosti so zabeležili dramatičen padec napak (75% od let 1987-91, na 37% od 1991-95). Stroški izdelave programske opreme so padli za 55% od 1987-91 in 42% od 1991-95. Ponovna raba se je izboljšala za 300% od 1987-91 in 8% od 1991-95. Povečala se je tudi funkcionalnost programske opreme za 5 krat od 1976-92. Novejši podatki niso na voljo, ker so v NASI zaradi gospodarske krize zmanjšali sredstva in s tem se je zmanjša tudi obseg delovanja in razpoložljivih statističnih podatkov.

Organizacija izdeluje programsko opremo izključno za NASO, zato je primerjava z ostalimi organizacijami, ki delujejo na trgu težavna. Omenjena študija zajema podatke 15 let delovanja, v celoti pa deluje 25 let.



### 4.2.2 Daimler

Daimler (prej Daimler Chrysler) je implementiral tri tovarne izkušenj v različnih okoljih v obdobju dveh let v sodelovanju z nemško univerzo v Ulmu (Landes, Schneider, & Houdek, 1999). Okolja v katerih so izvedli implementacije:

1. Oddelek odgovoren za razvoj programske opreme za letalsko področje.
2. Oddelek, ki razvija manjše vgradne sisteme za avtomobile, kjer je poudarek na prenosljivosti programske opreme preko različnih mikrokrmilnikov
3. Administrativni oddelek programske opreme, ki upravlja notranje procese, kot je npr. prodaja avtomobilov. Oddelek deluje samo na zahtevo, dejanski razvoj programske opreme izvajajo najeti izvajalci.

Študija podaja informacije v obliki poročila naučenih lekcij (angl. *lessons learned*), ki opisuje ugotovitve opisanih treh okolij:

- Obstaja veliko virov izkušenj, meritve so le en od teh virov.
- Opisujejo težave glede oblike v kateri so želeli uporabniki prejeti izkušnje.
- Obdelava kvalitativnih podatkov je predstavljala ozko grlo.
- Gradnja modelov napovedi na podlagi kvantitativnih podatkov je bila težavna, kadar so manjkale kontekstne informacije.

Opisana je tudi razprava glede prednosti in težav uvedbe tovarne izkušenj na način od zgoraj-navzdol (angl. *top-down*) in spodaj-navzgor (angl. *bottom-up*).

### 4.2.3 ICL High Performance Systems

ICL High Performance Systems iz Velike Britanije je razvil ogrodje za izboljšanje inženirskega procesa (angl. *Engineering Process Improvement Framework*), ki vključuje repozitorij za deljenje znanja (Chatters, 1999). Inženirska baza znanja vsebuje informacije, ki so razdeljene v tri kategorije:

- a.) projekti in procesi: opisi procesov,
- b.) tematski poučni materiali za uvedbo novih konceptov in
- c.) splošno ozadje in dodatne informacije.

Glavni cilj uvedbe tega programa izboljšav je bil boljše napovedovanje in ocenjevanje stroškov in rokov izvedbe sistemov in programskih rešitev.

Avtorji trdijo, da so člani projekta zaznali, da je ogrodje povzročilo prenos novega načina dela, vendar so te trditve potrjene samo skozi pripovedi članov. Glavna prednost je bilo zmanjšanje tveganja za doseg zastavljenih časovnih rokov, stroškov ter zagotovitev zahtevane kvalitete. Poročajo o naučenih lekcijah glede uvedbe, predvsem glede potrebne podpore managementa in vključenosti razvijalcev že ob zasnovi ogrodja. Poročilo študije je v obliki kvalitativnega opisa naučenih lekcij.

#### 4.2.4 Telenor Telecom Software

V okviru naporov za ponovno uporabo izkušenj glede procesa razvoja programske opreme, se je podjetje Telenor Telecom Software, ki ima več kot 400 zaposlenih razvijalec na 5 geografsko ločenih lokacijah, odločilo za izboljšavo ocene potrebnih sredstev za razvoj programske opreme, kot tudi management tveganj (Jørgensen, Conradi, & Sjøberg, 1998). Za doseg zahtev so vzpostavili:

- proces ponovne uporabe izkušenj, z novimi vlogami in nalogami,
- orodje za podatkovno bazo izkušenj, dosegljivo preko intraneta organizacije in
- zagotovili vire za ponovno uporabo izkušenj in administracijo baze izkušenj.

Baza izkušenj je na voljo kot ekspertni sistem, ki uporabniku zastavlja vprašanja glede narave novega projekta, nato pa pripravi model predlogov in ocen, glede na podatke iz preteklih projektov. Dodatno zajame in poda mnenja strokovnjakov znotraj podjetja glede podanega modela ocen. Baza je povezana z modulom za management tveganj, ki doda faktorje tveganj in jih pripravi na podlagi intervjujev izkušenj projektnih managerjev. Ta modul sestavlja nabor procesov najboljših praks (angl. *best practise*), orodja za identifikacijo, oceno in hrambo faktorjev tveganj. Na voljo pa je tudi orodje za vizualizacijo tveganj s časovno komponento.

Dodatno so uvedli vlogo izkušenega administratorja podatkovne baze, odgovorno za tehnične in uredniške vsebine, kot dodatne vloge analitikov procesov, odgovornih za analizo informacij pridobljenih iz obstoječih procesov kot je proces ocene projekta, procesa projektnega managementa in procesa testiranja programske opreme.

Avtorji priznavajo, da je bila študija izdelana prehitro po uvedbi, zato relevantni izsledki niso na voljo. Kljub temu pa zaznavajo nekatere izboljšave:

- Natančnost ocen se je izboljšala, posledično so se modeli ocen pričeli več uporabljati.
- V projektih se je več pozornosti pričelo namenjati izkušnjam glede tveganj.
- Organizacija je sprejela potrebo po zbiranju in deljenju izkušenj.

Poročilo študije je v obliki kvalitativnega opisa naučenih lekcij.

## 5 MANAGEMENT ZNANJA V PODJETJU BUYITC

BuyITC je slovensko storitveno podjetje s področja informacijske tehnologije, ustanovljeno leta 2000. V desetih letih poslovanja dosega podjetje kontinuirano rast in razvoj. Podjetje ima 21 zaposlenih, od katerih je 16 razvijalcev programske opreme.

Leta 2008 je podjetje sprejelo strateški načrt reorganizacije podjetja k procesni usmeritvi. Postavljeni so bili cilji glede spremenjenega načina dela. Uvedla so se orodja za vodenje delovnega časa in delovnih procesov.

Z uvedbo orodja za vodenje delovnih procesov (angl. *Workflow Management*) v letu 2010, se je pokazala potreba po ponovni uporabi izkušenj in znanja. Zaradi velikosti podjetja je neto fluktuacija zaposlenih velika, 20%. Nov zaposleni razvijalec potrebuje 6 do 12 mesecev za uvajanje. Količina pomoči pri izvajanju nalog in dela na projektih je v prvem letu velika in obremenjuje ostale člane projekta. V primeru servisnih storitev je takšnih situacij veliko, predvsem za starejše projekte oziroma programsko opremo.

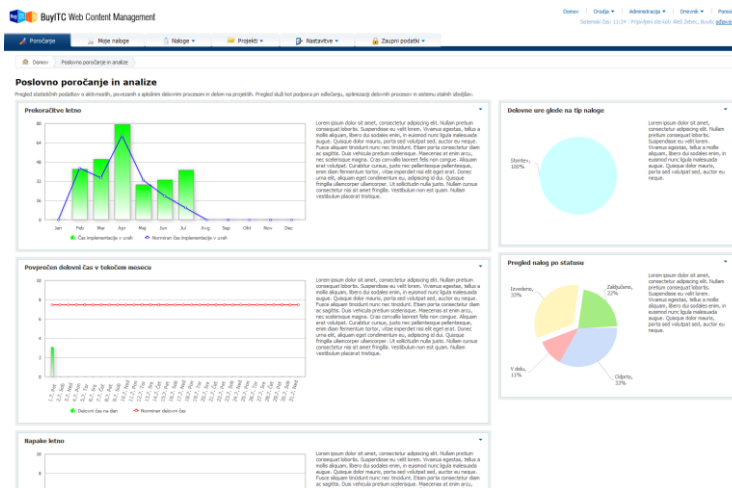
V podjetju je bilo ugotovljeno, da je veliko izkušenj in znanja shranjeno v nestrukturirani obliki v različnih dokumentih, e-pošti, vzorcih kode, shemah, modelih itd. Management je na podlagi dejstev in ugotovitev sprejel uvedbo sistema za podporo managementu znanja. Sestavljena je bila delovna skupina treh zaposlenih, katerih zadolžitve so povezane s kodificiranjem izkušenj in znanja. Skupino je sestavljala član managementa, vodja razvijalcev in vodja analitikov.

Na podlagi analize in načrtovanja je podjetje konec leta 2010 pričelo z razvojem lastne programske rešitve za podporo managementu znanja (BuyITC KM). Sistem je predstavljal informacijski portal, ki je del intraneta podjetja. Informacijski portal omogoča uporabnikom vnos in iskanje dobrih praks, navodil uporabe programskih orodij in sistemov, koščkov programske kode, ki rešuje določen problem, procesne, stroškovne in napovedne modele ter imenik znanja zaposlenih.

Zahteva managementa je bila, da je proces kodificiranja čim bolj avtomatiziran, da bodo stroški zajema nizki. V okviru sistema je tako podjetje razvilo več orodij za kodificiranje znanja in izkušenj, ki so se integrirale v delovno okolje, primeri:

- vtičniki MS Office orodij,
- obdelave za zajem podatkov o projektih iz sistema za upravljanje delovnih procesov,
- vmesniki za vnos in opis koščkov programske kode,
- sistem za pripravo navodil v strukturirani obliki,
- napredni vmesniki za objavo dobrih praks,
- obdelave za prenos in vizualizacijo podatkov poslovnega poročanja.

SLIKA 5: PRIMER VMESNIKA ZA VIZUALIZACIJO ZNANJA IN IZKUŠENJ SISTEMA BUYITC KM



Vir: lasten

Velik poudarek je bil tudi na uporabi strukturiranega in formaliziranega znanja ter izkušenj pri delu. Izdelan je bil napreden sistem, ki na podlagi iskalnih nizov uparja strukturirano znanje z delovnimi naloga zaposlenih.

Vsak zaposleni uporablja pri delu sistem za vodenje delovnih procesov. Sistem procese razgradi na naloge in jih dodeljuje posameznim izvajalcem, ki za rešitev naloge izvedejo potrebna opravila. Sistem za management znanja je integriran na nivoju prejete naloge. Ko izvajalec nalogo odpre sistem samodejno poišče relevantno vsebino za določeno nalogo iz baze znanja ter jo predstavi uporabniku. Uporabniku tako ni potrebno posebej iskati vsebine, znanja, izkušenj preko portala.

Glede na podatke sistema za upravljanje delovnih procesov je v prvih treh mesecih bilo porabljenega za iste naloge 5% manj časa kot šest mesec pred uvedbo. Glede na kvalitativno analizo (intervjuji zaposlenih) je razlog manjša potreba pomoči pri izvajanju nalog in hitrejši dostop do potrebnih informacij. Povečala se je tudi kvaliteta programske opreme, kjer se je število napak v istem časovnem obdobju zmanjšalo za 3%. Sistem se je začel uporabljati meseca marca 2011. Podjetje je na podlagi začetnih podatkov pozitivno ocenilo napore v smeri management znanja in odobrilo dodatne projekte razširitve sistema.

## 6 KLJUČNI DEJAVNIK USPEHA

V nadaljevanju opisujem cilje, ki so si jih zastavile organizacije v obravnavanih študijah glede managementa znanja, torej strategije. Sledi opis procesov in orodij za doseg zastavljenih ciljev.

### 6.1 Strategija

Organizacije so želele izboljšati situacijo za razvijalce, vendar niso imele jasno zastavljenih ciljev, glede kvalitete in stroškov razvoja. Med te uvrščamo Daimler in ICL, čeprav je ICL želel v osnovi izboljšati napovedovanje stroškov in rokov izvedbe. V NASI in Telenorju so opredelili nižanje stroškov in izboljšanje kvalitete kot primarna cilja strategije management znanja. Glede na strategijo management znanja lahko delimo organizacije v skupino, ki je izbrala strategijo personalizacije in skupini, ki je izbrala strategijo kodifikacije. Vse organizacije so imele strategijo kodifikacije, strategijo personalizacije pa nista imela ICL in BuyITC.

SLIKA 6: PREGLED AKTIVNOSTI ORGANIZACIJ IN UPORABA STRATEGIJ

Organizacija	Aktivnosti	Pristop k managementu znanja				
		Personalizacija	Kodifikacija?	Kvantitativno?	Kvalitativna?	Reorganizacija?
NASA SEL	Vzpostavitev ločene organizacije za zbiranje in distribucijo izkušenj.	Da	Da	Da		Da
Daimler	Vzpostavitev treh tovarn izkušenj v treh različnih oddelkih.	Da	Da	Da	Da	Da
Telenor	Izdelava ekspertnega sistema, zasnovanega na lastnih empiričnih podatkih za oceno naporov in upravljanje tveganj, skupaj z novimi vlogami.	Da	Da	Da	Da	Da
ICL	Uvedel intranet sistem, ki vključuje podatkovno bazo inženirskega znanja.		Da		Da	Da
BuyITC	Uvedel informacijski portal, ki vključuje bazo dobrih praks, opise in modele procesov, napredni iskalnik, ki samodejno povezuje znanje z nalogami in aktivnostmi zaposlenih.		Da		Da	Da

*Vir: lasten*

### 6.2 Procesi

Ko pogledamo procese, ki so prisotni v študijah, vidimo, da organizacije povdarjajo potrebo aktivnega sodelovanja razvijalcev pri zbiranju in distribuciji znanja. Vse organizacije so v okvirju uvedbe management znanja izvedle reorganizacijo, kjer se je ločen del organizacije ukvarjal z znanjem. Oblika znanja, ki se je zbirala preko teh procesov je bolj kvantitativna (npr. meritve velikosti kode), kot kvalitativna (opisi izkušenj).

### 6.3 Programske rešitve

Organizacije uporabljajo naslednja orodja: intranet sisteme za izmenjavo znanja, Telenor vključuje opise tehničnih procesov, ICL pa vključuje sezname znanja, kompetenc in spretnosti zaposlenih, skupaj s seznamami stran, poslovnih partnerjev in projektov. Telenor in BuyITC sta podjetji, ki sta razvili ekspertni sistem, Telenor za izdelavo modelov napovedi in ocen, ki je dostopen preko intranet organizacije, BuyITC pa za samodejno povezovanje znanja in izkušenj z nalogami zaposlenih.

## 6.4 Rezultati

V nadaljevanju so opisani rezultati uvajanja managementa znanja, kot povzeto iz študij. Tabela opisuje implikacije, prednosti in koristi.

SLIKA 7: IMPLIKACIJE UVEDBE MANAGEMENTA ZNANJA

Organizacija	Implikacije	Prednosti in koristi		
		Zadovoljstvo razvijalcev?	Nižji stroški?	Izboljšana kvaliteta?
NASA SEL	Zmanjšanje števila napak, nižji stroški, izboljšane funkcionalnosti.		Da	Da
Daimler	Študija ne vsebuje podatkov implikacij.			
Telenor	Zaznana večja natančnost glede ocen in osredotočenosti na upravljanje tveganj.		Da	
ICL	Zaznava novega načina dela.	Da		
BuyITC	Zaznava bolj učinkovitega načina dela in večji prenos znanja med zaposlenimi in projekti.	Da	Da	Da

*Vir: lasten*

Odgovor na vprašanje naloge, ali uvedba managementa znanja izboljša kvaliteto programske opreme, najdemo pritrdilen odgovor študiji NASA in BuyITC.

Ali uvedba managementa znanja zniža stroške razvoja? Študije NASA, Telenor in BuyITC navajajo, da so stroški nižji po uvedbi. Kvantitativno dokazano samo v študiji NASA in BuyITC, kjer so se stroški znižali s hitrejšo izvedbo projektov. Telenor zaznava večjo natančnost ocenjevanja stroškov in dela, kar ugodno vpliva na stroške. Tri organizacije od petih zaznavata nižje stroške po uvedbi managementa znanja.

Zadnje vprašanje, ali uvedba managementa znanja izboljša položaj zaposlenega v organizaciji, je pritrdilno v študiji ICL in BuyITC, kjer navajajo, da so zaposleni skozi uvedbo managementa znanja zaznali nov način dela in so manj obremenjeni z iskanjem rešitev za probleme.

Glavna težava uporabljenih študij so večinoma kvalitativne oblike podatkov, predvsem poročila v obliki naučenih lekcij. Kvantitativni podatki so na voljo samo pri študiji NASA, katere podatki niso aktualni (reorganizacija zaradi gospodarske krize) (Basili, 2002). Dodatno so v študijah management znanja in implikacije ocenjevali isti ljudje, ki so ga uvajali. Omejeno je tudi število študij primerov, kar onemogoča izpeljavo splošnih ugotovitev.

## 7 SKLEP

Namen naloge je bil pregled literature iz področja managementa znanja v procesu razvoja programske opreme. Cilj naloge pa s pomočjo pregleda študij primerov ugotoviti različne pristope k managementu znanja in implikacije uporabe. Opredeliti napore podjetja BuyITC, v katerem delam, glede managementa znanja in izvesti komparativno analizo s podatki iz študij primerov.

Prvo poglavje opisuje razloge za uporabo managementa znanja ter predlaga rešitve. Sledi poglavje, ki opisuje koncepte znanja, management znanja, modele, strategije, koncept tovarne izkušenj in model managementa znanja. Tretje poglavje je pregled študij primerov uvedbe managementa znanja v organizacije, ki razvijajo programsko opremo. V poglavju je posebej izpostavljena uvedba managementa znanja v lastnem podjetju. Četrto poglavje opredeljuje ključne dejavnike uspeha implementacije managementa znanja v proces razvoja programske opreme skozi komparativno analizo posameznih študij primerov in obravnavanega podjetja.

Analiziral sem štiri študije primerov in izvedel pregled področja v lastnem podjetju. Ugotovitve so naslednje: vsa podjetja so izvedla reorganizacijo in vzpostavila ločene oddelke za management znanja. Vse organizacije hranijo izkušnje (strategija kodifikacije), tri od petih pa omogočajo pretok znanja v organizacijo (strategija personalizacije). Večina prenaša znanje v kvalitativni obliki. Koristi in prednosti uvedbe managementa znanja je težko opredeliti, saj jasne podatke glede izboljšane kvalitete programske opreme navajata samo dve študiji. Isti študiji navajata znižane stroške. Ostale študije so v kvalitativni opisni obliki glede naučenih lekcij ob uvajanju in ne vključujejo kvantitativnih informacij. Na vprašanje glede izboljšane položaja zaposlenih ob uvedbi tovarne izkušenj, študije nakažejo koristi v prihranjenem času, poenostavljenih delovnih opravilih in hitrejšega vključevanja zaposlenih v projektne skupine.

Organizacije so večinoma izbrale kombinacijo strategije kodificiranja in personalizacije, povezave med vrsto managementa znanja, rezultati in koristmi tako ni mogoče opredeliti.

Po pregledu literature in študij primerov uvajanja managementa znanja v programskem inženiringu lahko sklepamo, da obstaja veliko iniciativ za razvoj podpornih tehnologij, vendar je na voljo malo empiričnih analiz, ki opisujejo kako deljenje in izmenjava izkušenj dejansko deluje. Metode ocenjevanja so večinoma kvantitativne, kar onemogoča neposredne primerjave rezultatov.

Znanje danes razumemo kot strateško najpomembnejši vir konkurenčnosti. Management znanja sicer ni nov koncept, vendar je postal aktualen, ko so managerji pričeli razmišljati o sistemskem pristopu ustvarjanja, pridobivanja, organiziranja, prenašanja in uporabe znanja.

## 8 LITERATURA IN VIRI

1. Jashapara, A. (2004). Knowledge Management, An Integrated Approach. *Pearson Education Limited*.
2. Morten T Hansen, Nitin Nohria, & Thomas Tierney. (1999, March). What's your strategy for managing knowledge? *Harvard Business Review*, 77(2), 106-116. (Document ID: 39332672).
3. Cindy Hubert, & Darcy Lemons. (2008, June). *APQC's Levels of Knowledge Management Maturity*. APQC. Najdeno 23. junija 2011 na spletnem naslovu <http://www.apqc.org/knowledge-base/documents/apqcs-levels-knowledge-management-maturity>
4. Sveiby, K. (1998). Measuring Intangibles and Intellectual Capital - An Emerging First Standard. Najdeno 23. junija 2011 na spletnem naslovu <http://www.sveiby.com/articles/emergingstandard.html>
5. Sveiby, K. (2001). Methods for Measuring Intangible Assets. Najdeno 23. junija 2011 na spletnem naslovu <http://www.sveiby.com/articles/IntangibleMethods.htm>
6. Schneider, K. (2009). Experience and Knowledge Management in Software Engineering *Springer; 1 edition*
7. Aamodt, A., & Nygård, M. (1995). Different roles and mutual dependencies of data, information, and knowledge - an AI perspective on their integration. *Published in Data and Knowledge Engineering*, 16, 191-222.
8. Ackerman, M. S., & Halverson, C. A. (2000). Reexamining organizational memory. *Association for Computing Machinery. Communications of the ACM*, 43(1), 58-58-64.
9. Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). *The Experience Factory: Encyclopedia of Software Engineering*: John Wiley.
10. CHAOS Summary 2009 Report. (2009). Massachusetts.
11. Chatters, B. (1999). *Implementing an experience factory: maintenance and evolution of the software and systems development process* Paper presented at the (ICSM '99) Proceedings. IEEE International Conference on Software Maintenance. <http://ieeexplore.ieee.org.nukweb.nuk.uni-lj.si/stamp/stamp.jsp?tp=&arnumber=753514>
12. Davenport, T. H., De Long, D. W., & Beers, M. C. (1998). Successful Knowledge Management Projects. *MIT Sloan Management Review*, 39(2), 43-43-57.
13. Davenport, T. H., & Prusak, L. (1998). *Working Knowledge: How Organizations Manage What They Know*. Cambridge, MA: Harvard Business School Press.
14. Deming, W. E. (2000). *Out of the crisis*: MIT Press.



15. Glass, R. L. (1999). The realities of software technology payoffs. *Association for Computing Machinery. Communications of the ACM*, 42(2), 74-79.
16. Glass, R. L. (2005). IT Failure Rates--70% or 10-15%? *Software, IEEE*, 22(3), 112-111. doi: 10.1109/MS.2005.66
17. Hansen, M. T. (1999). The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Administrative Science Quarterly*, 44(1), 82-82-111.
18. Jørgensen, M., Conradi, R., & Sjøberg, D. (1998). *Reuse of software development experience at Telenor Telecom Software*. Paper presented at the Telenor Telecom Software, "Proceedings of the European Software Process Improvement Conference (EuroSPI'98).
19. Landes, D., Schneider, K., & Houdek, F. (1999 ). Organizational Learning and Experience Documentation in Industrial Software Projects. *International Journal of Human Computer Studies*.
20. Nonaka, I., & Toyama, R. (2003). The knowledge-creating theory revisited: knowledge creation as a synthesizing process. *Knowledge Management Research & Practice*, 1(1), 2-2.
21. Seaman, C., Mendonca, M., Basili, V., & Kim, Y. M. (1999). *An Experience Management System for a*
22. *Software Consulting Organisation*. Paper presented at the Proceedings. of the Twenty-fourth annual NASA Software Engineering Workshop.
23. Skuce, D., & Lethbridge, T. CODE4: Conceptually Oriented Description Environment, from <http://www.site.uottawa.ca/kaml/CODE4.html>
24. Sommerville, I. (2006). *Software Engineering: (Update) (8th Edition)*: Addison Wesley.
25. Taylor, R. S. (1991). Information Use Environments. *Proc. of Progress in Communication Science*, 217-254.
26. Weiss, D. M., & Basili, V. R. (1985). Evaluating Software Development by Analysis of Changes: Some Data from the Software Engineering Laboratory. *Software Engineering, IEEE Transactions SE-11* (2), 157.
27. Basili, V.R.; McGarry, F.E.; Pajerski, R.; Zelkowitz, M.V.; , "Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory," *Software Engineering*, 2002. ICSE 2002. *Proceedings of the 24rd International Conference on* , vol., no., pp.69-79, 25-25 Maj 2002 URL: <http://ieeexplore.ieee.org.nukweb.nuk.uni-lj.si/stamp/stamp.jsp?tp=&arnumber=1007957&isnumber=21739>