

TABLE OF CONTENTS

CHAPTER TITLE PAGE	ALATI ZA UPRAVLJANJE SOFTVERSKOM KONFIGURACIJOM - ATACHMENT_1/23	UVOD
		3/23 RAZVOJ SCM-a KROZ ISTORIJU
		4/23 OSNOVNE FUNKCIJE SCM ALATA
		5/23 VERSION MANAGEMENT
		6/23 VARIANTS, WORKSPACE MANAGEMENT
		7/23 CONFIGURATION SELECTION
	8/23 BUILD, RELEASE AND DISTRIBUTED MANAGEMENT	9/23 CONCURRENT DEVELOPMENT
		10/23 CHANGE MANAGEMENT PROCESS
		11/23 TRACEABILITY
		12/23 PRIMER CR-A
		13/23 INTEGRATION WITH OTHER TOOLS
		14/23 COMERCIAL SCM TOOLS, ALLFUSION HARVEST CHANGE MANGER
	15/23 RATIONAL CLEARCASE	16/23 RATIONAL
	CLEARCASE, CM SYNERGY	17/23 CM SYNERGY
		18/23 MERANT PVCS
		19/23 MICROSOFT VISUAL SOURCE SAFE, LIST OF SCM
	COMMERCIAL TOOLS	20/23 LIST OF SCM COMMERCIAL TOOLS
		21/23
	FREEWARE SCM TOOLS, RCS TOOLS	22/23 CVS TOOLS, LIST OF FREE SCM
	TOOLS	23/23
		2/23

alati za upravljanje softverskom konfiguracijom

Boris Tikvenjac ITA/08

UVOD

U današnjem vremenu potreba za kompleksnim sistemima enormno raste. Takvi sistemi, sastavljeni od hardvera, softvera i prateće dokumentacije, pomeraju granice zahteva u integraciji sistema. Slozeni sistem se često razvija u nekoliko grupa i svaki njegov deo u razvoju zahteva kontrolu upravljanja PDM-a (Project Data Management) u slučaju hardverskih komponenti, ili SCM-a (Software Configuration Management) u slučaju softverskih komponenti. Često se dešava da integracija pojedinih delova sistema, koji sačinjavaju gotov proizvod, zahteva integraciju informacije pod kontrolom PDM i SCM-a. To nije nimalo lako za izvođenje, kada su u pitanju informacije različitog formata, bez direktne podrške PDM i SCM alata. PDM (Project Data Management) je disciplina koja se odnosi na kontrolu proizvoda, podataka i procesa koji se koriste tokom celokupnog životnog ciklusa proizvoda. Kompanije u razvoju proizvodnje hardverskih proizvoda tradicionalno koriste PDM. SCM (Software Configuration Management) je disciplina koja obuhvata kontrolu evolucije softverskog proizvoda. Mnoge firme imaju ozbiljne probleme u integraciji informacije posebno kada njima upravljaju različiti alati iz različitih domena. Zašto je to tako? Postoje mnogi razlozi, tehnički i ne tehnički. Tehnički gledano, teško je na efikasan način vršiti prenos i skladištenje podataka između različitih sistema. Interoperativnost PDM i SCM-a mogu biti ključan faktor u efikasnom upravljanju ukupnih informacija o proizvodu.

3/23

alati za upravljanje softverskom konfiguracijom

Boris Tikvenjac ITA/08

SCM (Software Configuration Management)

SCM je disciplina softverskog inženjerstva koja služi za kontrolu i upravljanje projektima i sinhronizaciju u radu različitih programera (developer) koji su učestvovali u projektu. SCM je projektovan da definiše metode i procese, pripremu plana i korišćenjem alata umnogome pomaže programerima ili projekt menadžerima u svakodnevnom radu u razvoju projekta. SCM je usmerena na podršku postepenog razvoja softvera. Iako se SCM koristi u svim fazama razvoja softvera, većina SCM aktivnosti je skoncentrisana na ranu fazu razvoja, kada je programski kod zapravo izrađen. Razvoj softverskog proizvoda je podeljen u tri dela: • Softver dizajner i programer (developer) kreiraju dokumentaciju i izrađuju izvorni kod • Konstuktovanje softvera po sastavljanjem izvornog koda i mašinskog programa • Proizvodnja i distribucija softvera (npr. CD) Specifičnost softvera uključuje jednostavnost modifikacije, visoke troškove razvoja, male ili nikakve troškove proizvodnje, nedostatak konkretne arhitekture u svojoj konačnoj formi i visok stepen složenosti tokom faze razvoja. Istorijat razvoja SCM-a počinje sa razvojem softvera. 60-tih godina XX veka softver je sačinjavao monolitni program implementiran u jedan izvorni modul i nije bilo potrebe za SCM-om. Softversko inženjerstvo je bilo fokusirano na proizvodnju efikasnih algoritama koji zauzimaju malo memorije. U 70-tim a posebno u 80-tim godinama softver postaje složeniji. Programi su izrađivani u dve faze, kompilacija (izvođenje binarnih modula iz izvornog koda) i povezivanje (kombinovanjem binarnih modula u programu tj. binarna biblioteka). Prva generacija SCM alata se pojavila u tom periodu: • Source code control system (SCCS) koji obuhvata osnovne discipline SCM-a: CM (kontrolu upravljanja), razvijanje upravljanja i upravljanje verzijom. Kasnije je na poboljšanim verzijama SCCS-a on zamenjen sa RCS (Revision control system). U tom trenutku SCCS je bila usmerena na „glomazno programiranje“ (promena, obnova i sastav sistema). Zanimljivo je napomenuti da su ovi alati dominirali na SCM-u tržištu dugi niz godina i ti principi su i dalje uvedeni i koriste se u većini modernih alata. Danas se RCS nalazi u različitim oblicima (npr. imake, gnumake i u različitim projektnim datotekama) i on se koristi kao osnova za mnoge druge, napredne SCM alate. Tokom 90-tih

godina fokus SCM-a je premešten na „mногоstruko programiranje“ sa naglaskom na timski rad (podrška procesa, istovremeni razvoj i simultano inženjerstvo u celini). Promena upravljanja, radnog prostora upravljanja i podrške procesa postaje novi model za SCM alate. Složenost SCM alata se povećava kako softver postaje složeniji. U tom periodu pojavljuje se nekoliko novih, složenih, skupih i naprednih SCM alata koje nije bilo lako uvesti u razvojni proces. Iz tog razloga, mnoge kompanije su tada razvijale sopstvene sisteme ili su koristeći jednostavne SCM alate izvršavale druge SCM procedure ručno. Upotreba SCM-a se značajno povećala tokom ovog perioda iz nekoliko razloga. Sa složenijim razvojem softvera pojavljuje se i potreba za alatima sposobnim za upravljanje ove složenosti. Istovremeno razvoj softvera postaje važan u poslovnim delatnostima i u mnogim novim softverskim kompanijama na tržištu. Konačno, razvoj softvera je fokusiran na razvoj procesa pod uticajem Capability Maturity Model-a (CMM) iz Software Engineering Instituta (SEI) koji je naglasio važnost SCM-a kao ključni proces u Software Engineering-u.

**----- OSTATAK TEKSTA NIJE PRIKAZAN. CEO RAD MOŽETE
PREUZETI NA SAJTU. -----**

www.maturskiradovi.net

MOŽETE NAS KONTAKTIRATI NA E-MAIL: maturskiradovi.net@gmail.com