

RAZUMEVANJE PROCESOV MED IZVAJANJEM PROJEKTOV PROGRAMSKEGA INŽENIRSTVA

M. Krašna, I. Rozman
Fakulteta za elektrotehniko, računalništvo in informatiko
Laboratorij za informacijske sisteme

Povzetek

Vodenje projektov programskega inženirstva se sooča s posebnim problemom projektnega vodenja. Pomanjkanje zgodovinskih podatkov še povečuje tveganje pri gradnji plana projekta. V članku je predstavljena trditev, da procesi v izvajanju projekta programskega inženirstva ne samo niso stabilni, temveč so hiperobčutljivi. Opisani teoretični model lahko pomaga projektnemu vodji pri predvidevanju medosebnih motenj, do katerih prihaja v projektih.

Abstract

Software engineering project management is a specific project management problem. Lack of historic data increases risk in building the project plan. The authors present the thesis that processes in software engineering projects are not only unstable, but hypersensitive. By help of the theoretical model, described in the article, a project manager might foresee interpersonal disturbances in projects.



Uvod

Računalniška znanost je najhitreje razvijajoča se znanstvena disciplina sploh. Nove tehnologije se porajajo vsakodnevno. Metodologije razvoja programske opreme se spreminjajo. Uporaba programskih orodij in generatorjev kode je široko razširjena med podjetji, ki se ukvarjajo z kakršnim koli razvojem programske opreme. Edino področje, ki sodeluje v razvoju programske opreme in se skoraj ni spremenilo, je projektno vodenje. Od leta 1984, ko je pričel veljati standard ANSI/IEEE Std. 730-1984 (Software Quality Assurance Plan) in je bil 1989 dopolnjen s kontrolo konfiguracije, trenin- gom in upravljanjem tveganja ter znova potrjen, ni bilo nobenega napredka v aktivnostih vodenja projektov programskega inženirstva.

Planiranje pomeni razgradnjo projekta na delovne aktivnosti, ki jih je potrebno izvesti za izpolnitev cilja ali ciljev projekta. S pogledom na prej opisani standard lahko skoraj z gotovostjo trdimo, da je ta del planiranja postal inženirska disciplina. Prav tako planiranje pomeni tudi ocenitev potrebnega napora za izvedbo delovnih aktivnosti. Za ocenitev delovnega napora pa še ne moremo trditi, da je postalo inženirska disciplina.

Po drugi strani pa lahko najdemo članke, ki opisujejo

“padec” briljantno planiranih in organiziranih projektov. Problem leži v razumevanju procesov ob izvajanju projektov. Mi trdimo, da so procesi, ki se odvijajo v izvajanju projekta, medsebojno povezani in so hiper občutljivi. S to hipotezo je možno razložiti, kako lahko projekti, ki so pazljivo planirani in tudi kontrolirani, postanejo neuspešni. Namen članka je odpreti nov pogled na razumevanje motenj med izvajanjem projektov programskega inženirstva.

Trenutna situacija

Obsežna je literatura, ki se ukvarja z različnimi področji projektnega vodenja. Naš namen ni opisovanje že znanih dejstev. Bolj pomembno je česa v literaturi ne najdemo, če pa že najdemo pa to ni dovolj jasno. Interesantno je vzporedno področje vodenja projektov programskega inženirstva, ki ima veliko korelacijo s produktivnostjo: to je motiviranje. Članek B.W. Boehma [Boehm_89] pravi, da je najpomembnejši objekt pri razvoju informacijskih sistemov človek-uslužbenec, ki ga ni možno obravnavati kot orodje. Prav tako lahko zasledimo, da je največji motivacijski faktor zadovoljitev

pričakovani osebja. Dopolnil je staro zlato pravilo, ki se sedaj glasi: "Delaj z drugimi tako, kot bi hotel, da bi oni delali s tabo, če bi bil ti na njihovem mestu". To pravilo pogojuje razumevanje potreb človeka in razdelitev opravil, ki morajo biti v okviru projekta izvedena na tak način, da zadovoljijo človekove potrebe. V članku ni navedeno, da bi implikacija veljala tudi v drugi smeri. Kar pomeni, da tudi če poskrbimo za dobro motivacijo to še ni potreben in zadosten pogoj, da bi projekt uspel.

Namen naše raziskave je najti faktorje, ki vplivajo na stabilnost projekta programskega inženirstva. Začnemo lahko pri metrikah produktivnosti in ugotovimo, da je Walston-Felixova študija produktivnosti [Walston_77] izpostavila 29 faktorjev, ki vplivajo na proces razvoja programske opreme. Zanimivo je, da niti eden izmed faktorjev ni povezan z motivacijo. Naslednja študija je Boehmov COCOMO [Boehm_81] model, ki je zmanjšal število faktorjev na 15 v štirih kategorijah. Pet izmed faktorjev je povezanih z atributi osebja, a le na izkušnje in zmogljivosti. Šele pri P.J. Shankovi raziskavi [Shank_93] najdemo med pomembnejšimi faktorji tudi motivacijo. Problem te študije pa je, da je bila izvedena v okviru ameriške organizacije za zračno obrambo, kjer je osebje pazljivo izbrano preden ga nastavi na delovno mesto. Ta raziskava je tako namenjena za veliko ožje področje in ni povsem veljavna za navadne projektne organizacije.

Druga plat medalje so publikacije, ki so jih izdali praktiki - vodje projektov. Ti so znali zelo dobro opozoriti na probleme, ki se porajajo ob izvajanju projektov. Njihove ugotovitve so, da je motivacija eden zelo pomembnih faktorjev, če ne že ključni faktor pri projektih programskega inženirstva.

Študije so bile izvedene tako, da so izbrali veliko število potencialnih faktorjev in s pomočjo statistične korelacije izločili bistvene - invariantne faktorje. Ti invariantni faktorji so pokrili večji del variance sistema. Delovali pa so v dopustnih mejah le za izbrane meritve. Nad drugim vzorcem so se slabo izkazale.

Statistika je močno orodje v pravih rokah. Če hočemo, da deluje, moramo imeti dovolj veliko skupino vzorcev. Vzorca morajo biti primerljivi (merjeni na enak način, pod enakimi pogoji), da lahko podajo zakonitosti. Problem meritve željenih vzorcev pa pri vodenju projektov programskega inženirstva še ni zadovoljivo rešen.

Še en problem, na katerega moramo opozoriti, so zgodovinski podatki. Ker se računalniška znanost hitro razvija, je velika količina zgodovinskih podatkov nepriemerna zaradi različnih pristopov pri razvoju programske opreme in spreminjajočih se programskih tehnik. Drug del problema pa je, če vzamemo primerljive podatke iz neke druge organizacije, ki se ukvarja z razvojem programske opreme. Kakšen je nivo tveganja ob upoštevanju teh podatkov?

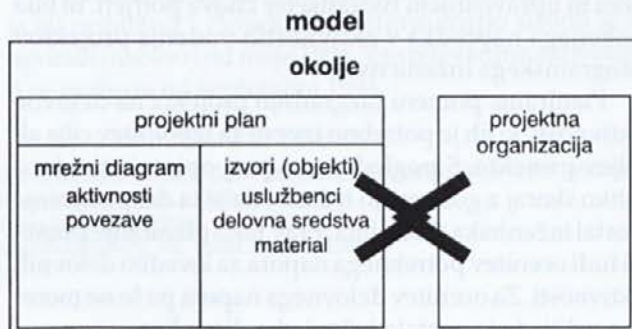
Predstavitev nove teorije: hiper - občutljivo vodenje projektov

V okviru vodenja projektov programskega inženirstva lahko najdemo dovolj natančne podatke, da izdelamo dober plan projekta. Raziskavo smo pričeli z vprašanjem ali je mogoče izdelati popoln plan. Poskusili smo zavreči to trditev. Ob predpostavki, da imamo popoln plan, smo poskusili najti vplivne faktorje, ki bi povzročili, da izvajanje projekta ne bi bilo uspešno. Da bi ugotovili, če projektna organizacija vpliva na stabilnost projekta, smo ta problem postavili na temelju znanih projektnih organizacij. Rezultati usklajenega razmišljanja so bili presežni. Največje tveganje pri projektu programskega inženirstva je bilo na področju motivacije in medčloveških odnosov. Sama razlaga je preprosta. Posameznik je kritični faktor pri razvoju programske opreme. Mentalnih aktivnosti, ki so potrebne za izvršitev plana projekta programskega inženirstva, v današnjih pogojih, še ni mogoče planirati. Ob tem nam je postalo jasno, da bo potrebno vključiti znanje psihologije in sociologije pri razvoju programske opreme.

Da bi lahko bolj sistematično raziskovali predstavljeni problem, smo morali izdelati procesni model projekta programskega inženirstva. V model smo morali vključiti vse objekte (projektne vire), ki sodelujejo pri razvoju programske opreme. Shema modela je predstavljena na sliki 1. Na sliki vidimo, da v okolju obstajata projektna organizacija in projektni plan, ki sta medsebojno odvisna. Projektni plan sestavlja mrežni diagram, ki vsebuje aktivnosti in povezave med njimi. Aktivnosti na najnižjem nivoju se lahko izvajajo le s pomočjo virov. Izvori se delijo v uslužbence, delovna sredstva in material. S shemo smo predstavili, kaj bo v našem modelu zajeto.

Projektne vire smo imenovali objekte, ker smo jim dodali attribute, in funkcije prehajanja stanj, ki vplivajo na spremembo stanja atributov, ki opisujejo stanje objektov.

Atributi imajo lahko različna stanja, ki so običajno med seboj primerljiva. Stanje objekta je predstavljeno z vektorjem stanj njegovih atributov. Vrednosti atributov niso konstantne v toku izvajanja projekta. To je bil



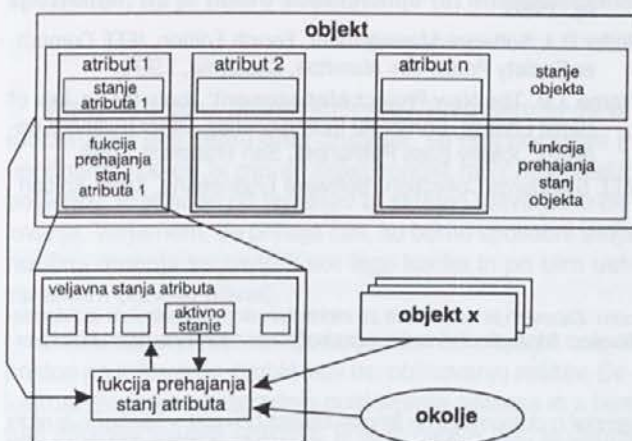
slika 1: Shema projektnega modela

razlog, da smo uvedli tudi funkcijo prehajanja stanj atributov. Funkcija prehajanja stanj objekta je sestavljena iz vektorja funkcij prehajanja stanj njegovih atributov. Funkcije prehajanja stanj objekta so odvisne od okolja in drugih objektov. Funkcije prehajanja stanj atributa so, v odvisnosti od objekta, lahko tudi v medsebojni odvisnosti. Hkrati pa so posledično odvisne tudi od drugih objektov in okolja, zaradi funkcije prehajanja stanj objekta, ki je na višjem nivoju in jih združuje.

Na sliki 2 so prikazani vplivi na funkcijo prehajanja stanj atributa in kje se le ta nahaja. V povečanem delu lahko vidimo, da na funkcijo prehajanja stanj vpliva:

- trenutno stanje atributa - aktivno stanje,
- objekt - s tem ponazorimo, da obstaja možnost medsebojnega vpliva atributov,
- drugi objekti in
- okolje.

Funkcija prehajanja stanj je utežnostna funkcija. S tem lahko reguliramo stopnjo posameznih vplivov. Lahko ima tudi verjetnostno komponento.

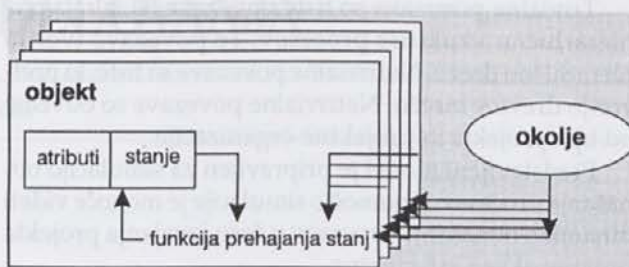


slika 2: Vplivi na funkcijo prehajanja stanj objekta

Če uporabimo model na sliki 1, od katerega vzamemo objekte, temu dodamo vplive okolja in medobjektnne povezave, dobimo model medobjektnih povezav, ki je prikazan na sliki 3.

Abstraktni - univerzalni objekt je objekt na najvišjem nivoju abstrakcije, ki pa ima strukturo podobno vsem ostalim objektom v našem modelu. Univerzalni objekt je osnovna komponenta našega modela. Model nadgrajujemo na naslednji način. Vzamemo projektni plan (predpostavimo, da je popoln), ki nam služi kot shema medobjektnih povezav. Projektni plan je razgrajen in na najnižjih nivojih najdemo opravila, katerim pripnemo objekte. V naš model vključimo tudi vpliv projektne organizacije, ki se kaže v dodatnih medobjektnih povezavah.

Model ima vse informacije o objektih, ki sodelujejo pri izvajanju projekta. V tej fazi moramo definirati še



slika 3: model medobjektnih povezav

processe med izvajanjem projekta programskega inženirstva.

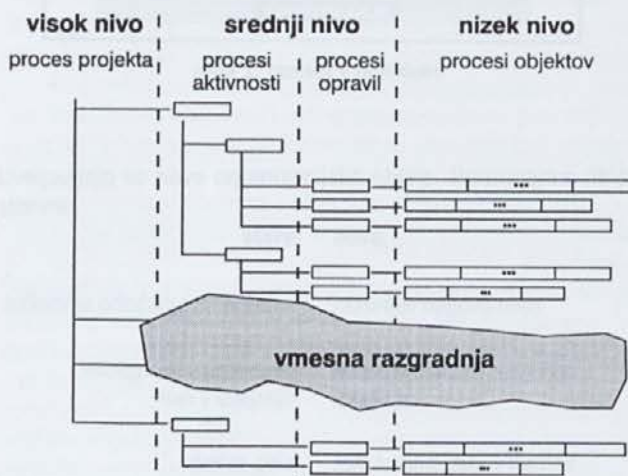
Procesi so večnivojski. Najvišji nivo vsebuje le en proces in ta predstavlja izvajanje projekta programskega inženirstva. Vsaka aktivnost ali opravilo, iz katerih je sestavljen projektni plan, je proces, ko pride v stanje izvajanja. Tak proces bomo imenovali proces projektne aktivnosti ali opravila. Razgradnja gre še dalje, da pridemo do vsakega objekta, ki sodeluje v procesu projektne opravila. Hierarhijo lahko predstavimo s pomočjo slike (slika 4).

Za naše pojmovanje je dekompozicija v tri nivoje povsem zadostna. Dekompozicija, ki nastopa pri strukturi delitve dela, je potrebna le za povečanje vidljivosti. Dejansko se pri izvajanju projekta izvajajo le opravila na najnižjem nivoju.

Tako nastali model sedaj vsebuje model projekta in procese, ki so pripojeni projektnemu modelu. Nejasne so le še medprocesne povezave.

Definirali bomo dva tipa medprocesnih povezav:

- trivialne povezave in
- netrivialne povezave.



slika 4: hierarhija procesov

Trivialne povezave so tiste povezave, ki izhajajo iz hierarhične strukture procesov. Te povezave tvorijo hierarhično drevo. Netrivialne povezave so tiste, ki pretvorijo drevo v mrežo. Netrivialne povezave so odvisne od tipa projekta in projektne organizacije.

Predstavljeni model je pripravljen za simulacijo obnašanja procesov. S pomočjo simulacije je mogoče videti dinamiko obnašanja procesov v času izvajanja projekta programskega inženirstva.

Zaključek

Znanstven način preučevanja projektov programskega inženirstva je brez simulacije nemogoč, ker ni mogoče zagotoviti enakih začetnih pogojev za večkratno ponovitev izvajanja projekta. Z modelom predstavljena teorija pomaga pri razumevanju dinamike procesov in njihovih medsebojnih vplivov. S simulacijo, kjer je mogoče spreminjati stanja objektov in funkcije prehajanja stanj, je mogoče preskusiti različne scenarije izvajanja projekta. Model je mogoče uporabiti za predikcijo pred pričetkom izvajanja projekta, kakor tudi za sprotno preverjanje stanja projekta, ki je že v izvajanju. Možna je identifikacija situacij, ki bi bile potencialno nevarne za izgube v projektu. Model je mogoče prilagoditi katerikoli organizaciji in projektnemu planu. S pomočjo simulacije je možno predvideti odziv sistema na kakršnokoli motnjo ali vpliv. Na tak način je mogoče videti odziv sistema na

korektivne akcije, ki so potrebne in jih je zelo težko dozirati. Prevelika korektura običajno vrže projekt iz stanja stabilnosti in se pretvori v svoje nasprotje. Model je prav tako sposoben pokazati v kateri situaciji se pojavi hiper - občutljivost procesov med izvajanjem projekta. S tem nas obvaruje, da bi prišli v situacijo, kjer bi projekt prenehal.

Literatura

- [Walston_77] Walston, C.E., and C.P. Felix. A Method of programming measurement and estimation. IBM System Journal 16, 1 (1977): 54-73
- [Boehm_81] Boehm, B.W. Software Engineering Economics. Englewood Cliffs, NJ: Prentice Hall, 1981
- [Boehm_89] Boehm, B.W. Theory-W Software Project Management: Principles and Examples, IEEE Transaction on Software Engineering, Vol. 15, No. 7, July 1989, pp. 902-916.
- [Shank_93] SHANK, P.J. Modeling as a Tool in Defense Aerospace Software Project Management, Claremont Graduate School, 1993
- Conte S.D., Dunsmore H.E., Shen V.Y. Software Engineering Metrics and Models, Benjamin/Cummings Publication Company, 1986
- Reifer D.J. Software Management, Fourth Edition, IEEE Computer Society Press, Los Alamitos, California, 1993.
- Frame J.D. The New Project Management: Tools for an Age of Rapid Change, Corporate Reengineering, Other Business Realities, Jossey-Bass Publishers, San Francisco 1994
- IEEE Standards Collection, Software Engineering, 1994 Edition

Mag. Marjan Krašna je diplomiral in magistriral na Tehniški fakulteti v Mariboru. Zaposlen je na Fakulteti za elektrotehniko, računalništvo in informatiko v Mariboru, v Laboratoriju za informacijske sisteme, kot mladi raziskovalec. Pripravlja disertacijo s področja upravljanja programskih projektov.

Dr. Ivan Rozman je diplomiral na Fakulteti za elektrotehniko v Ljubljani, magistriral in doktoriral pa na Tehniški fakulteti Univerze v Mariboru. Je redni profesor Univerze v Mariboru in ustanovitelj Laboratorija za informacijske sisteme, ki ga vodi še danes. Je avtor več kot 200 publikacij, vodil je številne znanstveno raziskovalne projekte.