

■ Izgradnja splošne metode za razvoj spletnih programskih rešitev

Sebastian Lahajnar
PRIS Inženiring, Ljubljana
sebastian.lahajnar@pris-inz.si

Andrej Kovačič
Ekonomski fakulteta, Univerza v Ljubljani
andrej.kovacic@ef.uni-lj.si

Povzetek

Razvoj spletnih programskih rešitev je kompleksen proces, ki vključuje številne aktivnosti, tehnike, orodja in izdelke kot rezultat sodelovanja skupin in posameznikov v različnih vlogah. V članku je opisana izgradnja splošne metode za načrtovanje in izvedbo spletnih programskih rešitev kot sestav neodvisnih komponent metode, z vključenimi ključnimi aktivnostmi, izdelki in tehnikami, ki se danes uporabljajo za modeliranje spletnih programskih rešitev, podatkovnih baz in poslovnih procesov. Predlagana metoda podaja dobre temelje za razvoj posameznim organizacijam in projektom prilagojenih metod za gradnjo spletnih programskih rešitev glede na specifične karakteristike projektov in okolje delovanja.

Ključne besede: spletne programske rešitve, spletni inženiring, inženiring metod

Abstract

Building a Common Method for Web Applications Development

Web application development is a complex process, which incorporates many activities, techniques, tools and artifacts as a result of collaboration of various groups and individuals, playing different roles. The article describes building of a common method for web application design and development as a composition of many independent method components, which incorporates key activities, artifacts and techniques, used nowadays for web application, database and business process modeling. The proposed method offers good foundation for developing organization and project specific situational web application development methods with regard to particular project characteristics and the surrounding environment.

Key words: web applications, web engineering, method engineering

Uvod

Spletno programsko rešitev opredelimo kot spletni sistem, ki vsebuje bolj ali manj kompleksno poslovno logiko z zmožnostjo spreminjanja stanja sistema v odvisnosti od vnosov uporabnikov [1]. Razvoj spletnih programskih rešitev postaja vse pomembnejši segment razvoja programske opreme, saj se danes kar 40 odstotkov razvojnih projektov s področja informacijske tehnologije ukvarja z elektronskim poslovanjem in spletnimi sistemi [2]. Če so bile v preteklosti spletne programske rešitve večinoma zgolj nadgradnja obstoječih informacijskih sistemov organizacij, danes vse več projektov s tega področja zadeva izgradnjo celovitih, funkcionalno samostojnih spletnih informacijskih sistemov za potrebe elektronskega poslovanja. Spletne programske rešitve združujejo funkcionalnosti hipermedijskih in klasičnih programskih rešitev, pri čemer prve zahtevajo sposobnost organiziranja obsežnih strukturiranih ali delno strukturiranih vsebin in definiranje različnih navigacijskih poti med njimi, druge pa opredeljuje funkcionalnih zahtev in različnih razvojnih vidikov.

Načrtovanje spletnih programskih rešitev je večdisciplinsko opravilo, ki vključuje analizo organizacije in poslovnih procesov, ki jih podpira spletna programska rešitev, strukturiranje različnih vrst podatkov kot temeljnih virov za podajanje vsebine, vizualno oblikovanje, upravljanje s podporno tehnologijo za izvajanje komponent spletnih programskih rešitev na strežniku in odjemalcu ter dostavo vsebine uporabnikom [3]. Razvoj spletnih programskih rešitev je zahteven projekt, ki zahteva integracijo vrste aktivnosti, izdelkov, tehnik in drugih gradnikov razvojnega procesa ter koordinacijo večjega števila ljudi z različnimi vlogami, znanji in pogledi na obravnavano problemsko področje.

Analiza obstoječih pristopov

Primerjava obstoječih razvojnih pristopov z vidika njihove primernosti za gradnjo spletnih programskih rešitev je vključevala analizo objektov razvojnih pro-

cesov, agilnih metodologij in specializiranih metod za načrtovanje spletnih programskih rešitev. Za analizo prvih dveh vrst pristopov je bil uporabljen nabor sedmih kriterijev oziroma zahtev, ki jih morajo izpolnjevati metode [4]: sposobnost upravljanja kratkih razvojnih ciklov, upravljanje spreminjajočih se zahtev, možnost sočasnega razvoja več različic izdelka, podpora postopkom za zagotavljanje ponovne uporabljivosti, vključenost tehnik za modeliranje, sposobnost prilagajanja različnim okoliščinam projekta in podpora celotnemu življenjskemu ciklu programske rešitve. Večina navedenih zahtev se v prvi vrsti nanaša na procesni vidik analiziranih pristopov, sam metodološki vidik, ki obravnava priporočene tehnike za izvajanje posameznih opravil, pa je vsebovan v zahtevi po vključenosti tehnik za modeliranje.

Analiza objektivnih razvojnih procesov razkriva številne pomanjkljivosti, ki zmanjšujejo primernost njihove uporabe za razvoj spletnih programskih rešitev. V prvi vrsti gre za probleme povezane z upravljanjem kratkih razvojnih ciklov in spreminjajočih se zahtev, saj večina objektivnih procesov poleg same programske rešitve narekuje izdelavo številnih vmesnih izdelkov (modelov, dokumentov itd.), kar posledično podaljšuje čas izvajanja projekta in čas izvedbe prilagoditev glede na nove oziroma spremenjene zahteve. Druga pomanjkljivost je odsotnost ustreznih tehnik za modeliranje specifičnih vidikov spletnih programskih rešitev kot tudi posebnosti drugih področij informacijske tehnologije (npr. podatkovnih baz, poslovnih procesov itd.), s katerimi je tesno povezan razvoj spletnih programskih rešitev. Odsotnost tehnik za modeliranje je tudi ena glavnih pomanjkljivosti agilnih pristopov, prav tako pa agilne metodologije, razen redkih izjem (npr. družina metodologij Crystal), ne vsebujejo mehanizmov za prilagajanje različnim tipom projektov. Sicer velja poudariti, da so agilne metodologije v svojih temeljih lahke (vključujejo zgolj nujne aktivnosti in izdelke), tako da s tega vidika večje prilagoditve, predvsem za majhne in srednje velike projekte, med katere spada tudi večina projektov izgradnje spletnih programskih rešitev, niti niso potrebne.

Za razliko od objektivnih razvojnih procesov in agilnih metodologij se specializirane metode za načrtovanje spletnih programskih rešitev posvečajo predvsem opredelitvi modelov in diagramskih tehnik za opis spletnih programskih rešitev na različnih ravneh (vsebine, hiperteksta in predstavitev), medtem ko je sam proces razvoja v drugem planu. Zapostavljenost

procesnega dela ima za posledico, da specializirane metode same po sebi ne posvečajo dovolj pozornosti zahtevam, kot so prilagajanje okoliščinam projekta, ponovna uporabljivost in sočasen razvoj več različic, medtem ko sta upravljanje kratkih razvojnih ciklov in spreminjajočih se zahtev po navadi omenjena v okviru priporočil in dobrih praks, ki jim velja slediti na tovrstnih projektih. Analiza specializiranih metod za razvoj spletnih programskih rešitev je bila tako omejena na obravnavo metodološkega vidika, torej tehnik in postopkov za opis različnih možnih vidikov sistema v izgradnji. Kot podlaga za oceno so bili izbrani naslednji vidiki [4]: analiza organizacije (zmožnost opisa temeljnih konceptov organizacije ter opisa podrobnosti poslovnih procesov), specifikacija zahtev (opredelitev funkcionalnih in nefunkcionalnih zahtev, klasifikacija in določitev prioritete), načrtovanje arhitekture (opredelitev ključnih gradnikov spletne programske rešitve na logični in fizični ravni), načrtovanje funkcionalnosti (podrobnejši opis funkcionalnih zahtev), načrtovanje podatkov (konceptualno načrtovanje podatkov in načrtovanje strukturiranih, delno strukturiranih ter večdimenzionalnih podatkov), načrtovanje navigacije (opredelitev različnih možnih poti navigacije po spletni programski rešitvi in mehanizmov dostopa do podatkov), načrtovanje predstavitve (specifikacija konceptualnega uporabniškega vmesnika in oblikovnih predlog) in načrtovanje prilagajanja (specifikacija statičnega in dinamičnega prilagajanja programske rešitve spreminjajočemu se okolju in zahtevam uporabnikov).

Primerjalna analiza specializiranih metod za načrtovanje spletnih programskih rešitev je pokazala, da v splošnem metode ustrezno pokrivajo specifikacijo zahtev, konceptualno načrtovanje podatkov, načrtovanje navigacije in načrtovanje predstavitve. Več pomanjkljivosti je zaslediti pri načrtovanju prilagajanja, saj metode vključujejo predvsem mehanizme za izvedbo statičnih prilagoditev, medtem ko je redkeje podprto dinamično prilagajanje. Velike razlike med metodami se kažejo pri načrtovanju funkcionalnosti, ki je ponekod odlično podprto z aktivnostmi, modeli in diagramskimi tehnikami, drugje pa ostaja v senci načrtovanja navigacije in predstavitve. Tudi pri modeliranju organizacije se metode med seboj razhajajo, pri čemer pa metode v prvi vrsti zagotavljajo mehanizme za opis poslovnih procesov, medtem ko je zapostavljeno modeliranje organizacije kot celote. Analiza je nadalje razkrila, da se večina metod

ukvarja zgolj s funkcionalnim vidikom programske rešitve, le redke podajajo koncepte za modeliranje arhitekturnega pogleda. Razen redkih izjem so zaposlavljeni tudi tehnike za opis strukturiranih in delno strukturiranih podatkov, medtem ko je modeliranje večdimenzionalnih podatkov povsem spregledano.

Teoretična podlaga metode

Primerjalna analiza obstoječih objektnih procesov, agilnih metodologij in specializiranih metod za načrtovanje spletnih programskih rešitev je pokazala, da nobeden od opisanih pristopov ne izpolnjuje v celoti postavljenih zahtev za gradnjo spletnih programskih rešitev. Rešitev je v opredelitvi lastnega metodološkega pristopa za gradnjo spletnih programskih rešitev z vključitvijo obstoječih, večinoma objektnih tehnik za modeliranje, ob upoštevanju dobrih praks, spoznanj in načel danes najširše uporabljenih pristopov za gradnjo informacijskih sistemov. Predlagani pristop poskuša združevati najboljše elemente (aktivnosti, izdelke, tehnike itd.) uveljavljenih pristopov v enovito metodo za razvoj spletnih programskih rešitev, ki v kar največji meri pokriva različne možne vidike izgradnje tovrstnih sistemov.

Teoretična podlaga predlagane metode za načrtovanje spletnih programskih rešitev je osnovana na konceptih discipline inženiringa metod [5], ki se ukvarja z načrtovanjem, izgradnjo in prilagajanjem metod, tehnik in orodij za razvoj programskih rešitev, oziroma njegove veje situacijskega inženiringa metod, katerega glavni cilj je prilagajanje obstoječih metod posameznim projektom. Temelj za izgradnjo novih metod so standardizirani sestavi, imenovani fragmenti metode, definirani kot skladni in dobro opredeljeni deli metod razvoja informacijskih sistemov [5]. Za potrebe razvoja lastnega ogrodja za situacijski inženiring metod kot podlage za izgradnjo predlagane metode za načrtovanje spletnih programskih rešitev je bil uporabljen koncept komponente metode, definiran kot samostojni del metode, ki opredeljuje postopek za preslikavo enega ali več izdelkov v neki končni izdelek ob hkratni utemeljitvi smisla tovrstne preslikave [6]. Komponenta metode (nadgradnja koncepta fragmenta metode) zagotavlja samostojnost in neodvisnost posameznega fragmenta metode, konsistentnost in koherentnost njenih elementov, racionalnost same komponente metode kot tudi njenih sestavnih delov, povezljivost z drugimi komponentami prek vmesnikov in ne nazadnje, uporabnost v raz-

ličnih situacijskih metodah ter izvedljivost s strani podpornih programskih orodij. Komponenta metode je opisana z uporabo dveh konceptualnih pogledov: notranjega in zunanega. Notranji model podaja notranjo strukturo komponente metode, ki temelji na naboru elementov (akcije, koncepti, notacije itd.) metode s pripadajočimi cilji, zunanji pogled pa podaja sliko, kako posamezne komponente metode sodelujejo pri doseganju zastavljenih ciljev. Komponente metode so v metodi med seboj povezane z vmesniki, ki vključujejo splošne cilje posamezne metode ter njene vhodne (pogoje) in izhodne izdelke (rezultate).

Osnovni koncept komponente metode je bil nadgrajen v več smereh [4]: z vključitvijo dodatnih elementov komponente metode v okviru notranjega pogleda, z razširitvijo zunanega pogleda prek vzpostavitve razmerij med komponentami metode in procesnim delom metode (fazami, iteracijami in mejniki) in z opredelitvijo mehanizmov za prilagoditev komponent metode značilnostim posameznih organizacij in razvojnih projektov. Nadgrajeni koncept komponente metode služi kot podlaga za izgradnjo ogrodja za situacijski inženiring metod za načrtovanje spletnih programskih rešitev, ki sestoji iz procesov: upravljanje repozitorija, izgradnje baze komponent metode, izgradnje temeljne metode organizacije in prilagajanje temeljne metode specifičnim okoliščinam posameznega projekta. Repozitorij ogrodja vključuje štiri med seboj povezane baze, katerih vsebina opredeljuje različne situacijske metode kot skupek prilagojenih komponent metode glede na lastnosti posamezne razvojne situacije oziroma projekta: bazo karakteristik, komponent metode, prilagoditev in bazo metod (temeljnih in prilagojenih). Konceptualna podlaga repozitorija ogrodja temelji na definiranih metamodelih notranjega pogleda, zunanega pogleda, vmesnika in konfiguracije komponente metode.

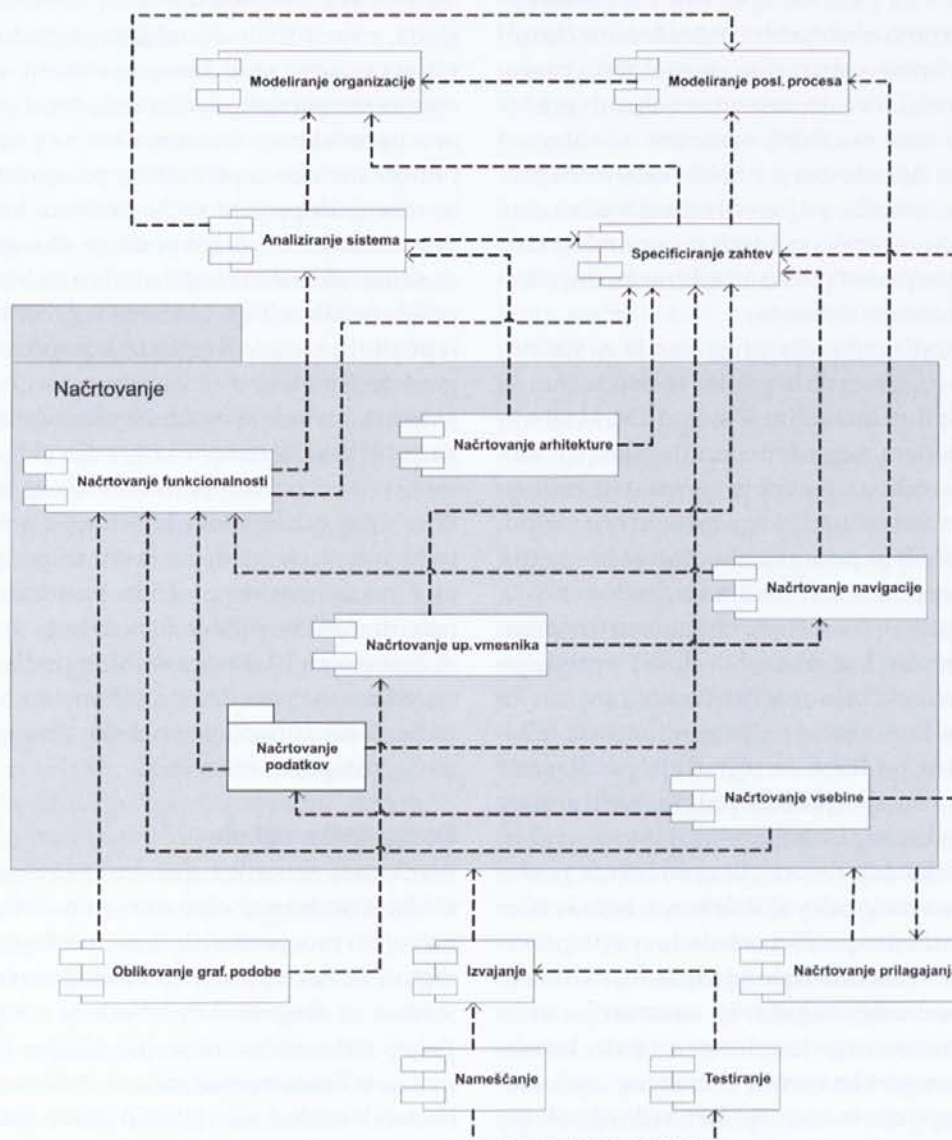
Komponente metode

Predlagani nabor komponent metode (slika 1) poskuša identificirati najboljše elemente obstoječih objektnih razvojnih procesov, agilnih metodologij, specializiranih metod za načrtovanje spletnih programskih rešitev, jezikov in diagramskih tehnik, ki v kar največji meri pokrivajo različne razvojne vidike. Definiranje posamezne komponente metode zahteva rekonstrukcijo izvornih metod ali njihovih delov skladno s postavljenimi pravili in metamodelom ogrodja za inženiring metod ter opredelitev vmesnikov. Predlagani pristop

za načrtovanje spletnih programskih rešitev je rezultat sinteze definiranih komponent metode z vzpostavitvijo povezav prek vmesnikov v enovito metodo. Izgradnja baz komponent metode, karakteristik in prilagoditev je neprestan proces, saj se začetni nabor podatkov sproti dopolnjuje v postopkih izgradnje novih metod ali prilagoditve obstoječih. V nadaljevanju je predstavljen povzetek osemnajstih v predlagano metodo vključenih komponent metode.

Glavni cilj komponente modeliranje organizacije je celovita analiza organizacije, ki predstavlja temelj za razumevanje njenega statičnega in dinamičnega vidika. Model organizacije skupaj z opredeljeno vizijo, po-

slovnimi pravili in slovarjem poslovnih izrazov zagotavlja enotno razumevanje različnih vidikov organizacije, njenih aktualnih problemov, kot tudi možnih izboljšav vsem vpletenim stranem (poslovodstvu, uporabnikom, razvijalcem). Komponenta metode modeliranje poslovnega procesa se ukvarja z modeliranjem in prenovo posameznega poslovnega procesa, identificiranega v modelu organizacije. Model poslovnega procesa omogoča boljše razumevanje delovanja procesa, opredelitev ključnih problemov in identifikacijo možnih izboljšav, obenem pa je tudi podlaga za analizo možnosti avtomatizacije, če proces še ni ustrezno računalniško podprt. Glavna vhoda v komponento



Slika 1: Odvisnosti med komponentami metode

sta model organizacije in vzpostavljena poslovna pravila. Za modeliranje organizacije in poslovnih procesov je na voljo več profilov jezika UML [7][8], predvsem na ravni posameznih poslovnih procesov pa lahko uporabimo tudi uveljavljene diagramске tehnike, kot so diagram EPC (Event Process Chain), Petrijeve mreže in notacija BPMN (Business Process Modeling Notation).

Komponenta metode specificiranje zahtev ima za cilj opredelitev in vzdrževanje funkcionalnih in nefunkcionalnih zahtev programske rešitve v izgradnji. Specificiranje zahtev omogoča identifikacijo ključnih potreb in pričakovani uporabnikov, zagotavlja ustrezno razumevanje obravnavane problematike in omogoča opredelitev mej sistema. V primeru, da je bilo predhodno izvedeno modeliranje organizacije ali modeliranje posameznih poslovnih procesov, sta vhoda v komponento metode model organizacije in nabor procesnih modelov. Specificiranje zahtev podpirajo različne tehnike od splošno uveljavljenih, kot so npr. študij obstoječe dokumentacije in obstoječih programskih rešitev, izvajanje intervjujev s ključnimi uporabniki, anketiranje itd., do novejših pristopov, kot sta specifikacija zahtev s primeri uporabe in uporabniške zgodbe. Komponenta metode analiziranje sistema izvaja preslikavo specifikacij sistema v obliko primerno za izgradnjo načrta. Vhodi v komponento metode so predvsem funkcionalne specifikacije in model primerov uporabe kot rezultati komponente metode specificiranje zahtev, pa tudi model organizacije in modeli poslovnih procesov, če so ti na razpolago. Glavni izdelek je model analize, ki vključuje zgolj najpomembnejše elemente sistema, njihova podrobnejša opredelitev je v domeni različnih komponent metod načrtovanja.

Cilj komponente metode načrtovanje funkcionalnosti je podrobnejša specifikacija programske rešitve upoštevajoč nefunkcionalne specifikacije in podane omejitve ciljne izvedbene tehnologije. Načrtovanje funkcionalnosti temeljni na več izdelkih (funkcionalne specifikacije, model primerov uporabe, model analize in model organizacije), ki so nastali kot rezultat drugih komponent in so večinoma vsaj v grobem izdelani pred izdelavo modela načrta funkcionalnosti. Med potekom projekta nastajajo tudi drugi modeli (navigacije, uporabniškega vmesnika itd.), ki lahko povratno vplivajo na model načrta funkcionalnosti, ki ga je treba ustrezno prilagajati. Glavni namen komponente metode načrtovanje arhitekture je izdelava arhitekture na fizični in logični ravni. Načrtovanje ar-

hitekture temelji na arhitekturnih specifikacijah in modelu analize, rezultati komponente metode pa so logični in fizični model arhitekture ter model vzorca, ki ga arhitektura uporablja. Logična arhitektura specificira programsko logiko brez opredelitve njene fizične porazdelitve po različnih računalniških sistemih, medtem ko fizična arhitektura specificira podrobnosti o dodeljevanju izdelkov programske opreme (komponent, modulov itd.) fizičnim vozliščem. Arhitekturo modeliramo z uporabo splošnih diagramskih tehnik jezika UML (razredni, komponentni diagram itd.) ali specializiranih profilov za področje spletnih programskih rešitev kot je WebSA [9].

Komponenta metode načrtovanje navigacije specificira možne navigacijske poti skozi spletno programsko rešitev. Vhod v metodo je opis funkcionalnih specifikacij in njihova podrobna opredelitev v obliki modela načrta. Če navigacijske poti sledijo določenemu poslovnemu procesu, predstavlja njegov model podlago za začetno definicijo poti. Za modeliranje se lahko uporabi ena od številnih diagramskih tehnik specializiranih metod za načrtovanje spletnih programskih rešitev, definiranih večinoma kot razširitve diagramov jezika UML ali modela ER (UWE – model navigacijskega prostora, strukture [10], profil WAE – model UX [11] itd). Komponenta metode načrtovanje uporabniškega vmesnika specificira uporabniški vmesnik programske rešitve na abstraktni ravni. Rezultata komponente metode sta model uporabniškega vmesnika in njegov prototip. Načrtovanje abstraktnega uporabniškega vmesnika je podprto z več tehnikami, vključenimi v specializirane metode za razvoj spletnih programskih rešitev (UWE – model predstavitve, profil WAE itd.).

Cilj komponente metode načrtovanje prilagajanje je prilagoditev spletne programske rešitve specifičnim zahtevam posameznih skupin uporabnikov, individualnim uporabnikom ali kontekstu, v katerem programska rešitev deluje. Prilagoditev spletne programske rešitve je lahko statična ali dinamična [11]. V primeru statične prilagoditve se prilagoditev izvede na podlagi vnaprej definirane rešitve, ki natančno opredeljuje nabor uporabljenih gradnikov (npr. navigacijskih poti, elementov uporabniškega vmesnika itd.), v primeru dinamične prilagoditve pa se nabor gradnikov zgradi dinamično na podlagi vzpostavljenih pravil prilagajanja.

Komponenta metode oblikovanje grafične podobe specificira in izvede grafično podobo spletne programske rešitve. Oblikovanje grafične podobe se

začne z opredelitvijo standardov, ki jim bo sledila grafična podoba. Na podlagi sprejetih standardov se izdelava nabor oblikovnih predlog, ki celovito opredelijo grafično podobo v rešitev vključenih spletnih strani, obenem pa se za vsak gradnik uporabniškega vmesnika opredeli njegova individualna oblika. Podoba uporabniškega vmesnika se preveri z izdelavo prototipa. Komponenta metode načrtovanje vsebine ima za cilj načrtovanje in ustvarjanje različnih vrst vsebin, ki jih bo vsebovala spletna programska rešitev. Glavna vhoda sta specifikacija vsebine in model podatkov (opredeljuje interno organizacijo vsebine), rezultati pa definirane predloge vsebin, opredeljene ključne podatkovne komponente (besedilna, številčna, avdio, video in grafična vsebina) in izdelani opisi (metapodatki) vsebine.

Komponenta metode načrtovanje relacijske podatkovne baze ima za cilj izdelavo načrta relacijske podatkovne baze v obliki normaliziranega logičnega podatkovnega modela in fizičnega podatkovnega modela. Vhoda v komponento sta model načrta funkcionalnosti in specifikacije programske rešitve. Postopek načrtovanja relacijske podatkovne baze se začne s preslikavo konceptualnega podatkovnega modela, realiziranega po navadi v obliki razrednega diagrama jezika UML ali modela ER in vključenega v model načrta funkcionalnosti, v logični (relacijski) podatkovni model. Ta se nato normalizira in dopolni z elementi specifičnimi za relacijske podatkovne baze (indeksi, ključi itd.). Logični model je podlaga za samodejno izdelavo objektov izbranega sistema za upravljanje podatkovnih baz, ki se izvede z uporabo na podlagi elementov logičnega modela izdelanih ukazov v jeziku SQL. Fizični model podatkovne baze po drugi strani omogoča boljši pregled nad fizičnimi objekti sistema za upravljanje podatkovnih baz, kot so sheme, tabele, prostori tabel itd. Za opis logičnega in fizičnega relacijskega podatkovnega modela uporabimo enega od profilov jezika UML za načrtovanje podatkovnih baz [12][13].

Komponenta načrtovanje podatkov XML ima za cilj opredelitev v spletni programski rešitvi uporabljenih shem XML za potrebe hranjenja podatkov (baze dokumentov XML) ali besednjakov za izmenjavo podatkov tako znotraj organizacije kot tudi navzven. Načrtovanja podatkov XML se lahko lotimo na več načinov, odvisno od postavljenih ciljev in obstoječih izdelkov. Po eni strategiji, vzamemo za podlago konceptualni podatkovni model, vključen v model

načrta funkcionalnosti in ga z uporabo tehnik za preslikavo konceptov razrednega diagrama UML v dokumente in sheme jezika XML preslikamo neposredno v dokumente ali sheme XML (DTD, XML Shema) [14]. Druga strategija priporoča uporabo enega izmed profilov za modeliranje shem XML [15][16], s katerim na razširjenem razrednem diagramu natančno opredelimo vse gradnike (elemente, attribute, povezave itd.), ki bodo v končni fazi tvorili posamezno shemo XML.

Komponenta načrtovanje podatkovnih skladišč se ukvarja z načrtovanjem podatkovnih skladišč in večdimenzionalnih podatkovnih baz kot možnim pristopom k realizaciji področnih podatkovnih skladišč. Načrtovanje podatkovnih skladišč temelji na konceptih relacijske podatkovne baze, dimenzijskega modela (tabela dimenzij, tabela dejstev, hierarhija itd.) in večdimenzionalne baze podatkov (kocka, dimenzija, hierarhija, celica itd.). Načrtovanje podatkovnega skladišča se začne z aktivnostjo izdelave logičnega modela podatkovnega skladišča, pri čemer lahko uporabimo klasični ali dimenzijski pristop, katerih rezultat je denormaliziran model ER ali dimenzijski model. Če so področna podatkovna skladišča podprta z večdimenzionalnimi podatkovnimi bazami, izdelamo še njihove konceptualne modele. Zadnja aktivnost je načrtovanje postopka ETL (Extract Transform Load) za migracijo podatkov, v katerem specificiramo postopke in pravila za prepis podatkov iz transakcijskega sistema v podatkovno skladišče.

Komponenta metode izvajanje ima za glavni cilj izvedbo spletne programske rešitve na podlagi opredeljenih specifikacij in zgrajenih modelov. Cilj komponente metode testiranje je ocena kakovosti programske rešitve, ki vključuje načrtovanje, izvedbo in vrednotenje različnih testov (konfiguracije, funkcionalnosti, integracije, sistema itd.). Cilj komponente metode nameščanje je namestitev in testiranje izdelane programske rešitve v okolju uporabnika, pri čemer predstavlja izdaja programske rešitve njen glavni koncept. Komponenta metode upravljanje zagotavlja učinkovito upravljanje projekta, izpolnjevanje najpomembnejših pričakovanj naročnika, njegovih zahtev in ključnih ciljev, vključuje pa planiranje in upravljanje faz ter iteracij, izdelkov, aktivnosti, virov, stroškov in drugih elementov projekta.

Odvisnosti med komponentami metode so pogojene z njihovimi vmesniki oziroma izdelki, ki predstavljajo vhode ali rezultate izvajanja aktivnosti komponente metode. V splošnem velja, da obstaja razmerje

odvisnosti med dvema komponentama metode v primeru, da ena komponenta metode uporablja izdelke druge komponente metode. Kakršne koli spremembe izdelkov osnovne komponente metode vplivajo na izdelke odvisnih komponent metode, kar je treba upoštevati pri načrtovanju še tako majhnih sprememb programske rešitve. Stvar je seveda toliko bolj pomembna, če spremembe posegajo v samo arhitekturo programske rešitve, model organizacije ali posamezne poslovne procese, ki predstavljajo najpomembnejše vhode v druge komponente metode.

Prilaganje metode

Za razliko od procesa RUP (Rational Unified Process), pri katerem se kot podlaga za novo situacijsko metodo vzame celoten proces, kateremu se nato odvzamejo nepotrebne aktivnosti (pristop od zgoraj navzdol) in ogrodja OPF (OPEN Process Framework), pri katerem se metoda gradi s sestavljanjem elementarnih komponent procesa (pristop od spodaj navzgor), predlagana metoda zagovarja kombinacijo obeh pristopov. Podlaga za prilaganje metode lastnostim posamezne organizacije ali projekta je nabor preverjenih komponent metode na višji ravni, namenjenih opisu ali izgradnji posameznega vidika programske rešitve. Od tu naprej obstajata dve možni poti prilaganja: prilaganje navzgor pomeni sestavljanje komponent metode v novo situacijsko metodo, prilaganje navzdol pa prilaganje posameznih elementov (aktivnosti, izdelkov, vlog itd.) izbranih komponent metode. Če tako npr. projekt ne predvideva analize poslovnih procesov, ker je bila ta že predhodno izvedena, projektu prilagojena situacijska metoda ne bo vsebovala komponente metode modeliranje poslovnih procesov. Za prilaganje posameznih elementov komponent metode se uporablja tehnika konfiguracije komponente metode [17], ki temeljni na konceptih konfiguracijskega paketa in konfiguracijske predloge, pri čemer je sam postopek prilaganja pogojen z identificiranimi vrednostmi posameznih karakteristik, ki opisujejo konkretno razvojno situacijo oziroma projekt.

Objektni in iterativni pristop

Objektni pristop k načrtovanju in izvedbi predstavlja temelj predlagane metode za izgradnjo spletnih programskih rešitev. Jezik UML je bil izbran kot standardni jezik za modeliranje, pri čemer večina v komponente metode vključenih modelov uporablja spe-

cifične, posameznemu področju prilagojene diagramske tehnike in profile jezika UML. Tak pristop v primerjavi z uporabo klasičnih diagramskih tehnik jezika UML omogoča izdelavo vsebinsko bogatejših modelov, posledica česar je lažji prehod iz načrtovanja v izvedbo. Predlagana metoda poleg objektnih omogoča in celo priporoča uporabo tudi drugih tehnik za modeliranje nekaterih specifičnih vidikov programske rešitve, predvsem poslovnih procesov in relacijskih podatkovnih baz. Izbor ustrezne tehnike je del aktivnosti prilaganja metode specifičnemu projektu in je odvisen od številnih dejavnikov, kot so npr. vrsta spletne programske rešitve, poznavanje tehnik, ciljna tehnologija itd. Druga razsežnost objektnega pristopa predlagane metode se kaže v sami zasnovi metode, ki je zgrajena kot sestav neodvisnih komponent metode. Vsaka komponenta metode predstavlja samostojno celoto, njena struktura je navzven skrita, povezava z drugimi komponentami metode pa je izvedena prek vmesnikov. In ne nazadnje, sama programska rešitev je realizirana kot sestav komponent in programskih modulov na različnih arhitekturnih ravneh z uporabo kombinacije več objektnih in objektno orientiranih programskih oziroma skriptnih jezikov ter tehnologij.

Predlagana metoda za izgradnjo spletnih programskih rešitev poleg objektnega pristopa poudarja predvsem iterativno izvajanje vseh v metodo vključenih aktivnosti. Iterativni pristop ima za posledico inkrementalni razvoj izdelkov, pri čemer je napredek lahko aditiven (npr. v primeru izgradnje novega modula ali modula) ali pa predstavlja zgolj dopolnitev oziroma nadgradnjo že obstoječih izdelkov. Razdelitev faz na iteracije je pogojena z različnimi dejavniki, kot so kompleksnost programske rešitve v izgradnji, razpoložljivost virov, ocena tveganja, podanih časovnih okvirjev celotnega projekta, ravni znanja in izkušenj v projekt vključenih ljudi na strani naročnika in izvajalca itd. Za spletne programske rešitve je v splošnem značilen krajši razvojni cikel od klasičnih programskih rešitev, kar posledično implicira manjše število iteracij v posameznih razvojnih fazah. Z večanjem kompleksnosti programske rešitve se povečuje tudi število iteracij, na kar vpliva tudi izbor časovnega okvira posamezne načrtovane iteracije. S tega vidika je priporočljivo čas izvajanja posameznih iteracij v okviru projekta poenotiti, npr. skladno s prakso 30-dnevnega sprinta, ki jo priporoča agilna metodologija Scrum.

Ocena predlagane metode

Ocena predlagane metode z vidika njene primernosti za gradnjo spletnih programskih rešitev temelji na enakem naboru kriterijev kot predhodno predstavljena analiza obstoječih pristopov. Razvojni cikel metode vsebuje iste obvezne faze kot proces RUP in dodatno še fazo produkcije, kar je lahko problematično z vidika zahteve po kratkih razvojnih ciklih. Rešitev je v prilagoditvi posameznih komponent metode v smeri poenostavitve njihovega izvajanja in zmanjšanja števila obveznih izdelkov. Upravljanje spreminjajočih se zahtev je v metodi obravnavano v okviru plana izdelka in plana iteracije. Pri pripravi plana gre v prvi vrsti za spekulacijo, pri čemer je obseg dovoljenih odstopanj odvisen od več dejavnikov, kot so npr. stopnja definiranih zahtev, kompleksnost in kritičnost projekta itd. Problem vpliva spremenjenih zahtev na druge izdelke lahko zmanjšamo s prilagoditvijo komponent metode posameznim projektom (vključimo zgolj ključne aktivnosti in izdelke). Podpora sočasemu razvoju več različic (poteka predvsem v fazi konstrukcije) je v metodo vključena posredno prek uveljavljanja načel sodelovanja in neposredne komunikacije, visoko stopnjo ponovne uporabljivosti izdelkov pa zagotavlja sama objektna usmerjenost metode, pri čemer je ponovna uporabljivost teže dosegljiva pri izdelkih v začetnih fazah razvojnega procesa, ki opisujejo specifičnosti posamezne organizacije. Sposobnost prilagajanja metode okoliščinam projekta temelji na sposobnosti prilagajanja posameznih njenih komponent, podlaga pa je definirano ogrodje za inženiring situacijskih metod. Metoda nadalje vključuje vseh pet ključnih razvojnih faz: začetek, zbiranje informacij, konstrukcijo, prevzem in produkcijo, s čimer izpolnjuje zahtevo po celovitem pokrivanju življenjskega cikla programske rešitve. Zahteva po vključenosti tehnik za modeliranje je podprta z definiranimi komponentami metode, ki modelirajo posamezne vidike spletne programske rešitve (poslovni proces, arhitekturo, navigacijske poti, uporabniški vmesnik itd.) in vključujejo vrsto specializiranih tehnik. Skupna ocena predlagane metode je, da ta v kar največji meri izpolnjuje postavljene zahteve za razvoj spletnih programskih rešitev, vključuje pa tudi obsežno paleto tehnik za vse opredeljene vidike modeliranja.

Predlagani pristop k razvoju spletnih programskih rešitev, ki temelji na konceptih discipline situacijskega inženiringa metod, pa prinaša tudi nekaj pasti, tveganj in z njimi povezanih problemov, ki se jih je treba zavedati, upoštevati in reševati med izvajanjem

projekta. Izgradnja in prilagajanje metode posameznim organizacijam in projektom povzroči nastanek dodatnih stroškov že na samem začetku projekta, prav tako pa so dodatni resursi potrebni za verifikacijo prilagojene metode in ne nazadnje izobraževanja njenih končnih uporabnikov. Navzlic uporabi vnaprej definiranega nabora komponent metode še vedno lahko nastopi tveganje glede primernosti končnega rezultata (prilagojene metode), če v postopku izgradnje in prilagajanja ne upoštevamo vseh relevantnih zahtev in dejavnikov, problematična pa je lahko tudi sama uporabnost metode, ki je odvisna od njene razumljivosti vsem v projekt vključenim akterjem.

Sklep

Usklajevanje in nadzor različnih aktivnosti, tehnologij, orodij in ljudi, ki s svojim delom in znanjem ustvarjajo izdelke, predstavlja pereč problem pri projektih razvoja spletnih programskih rešitev. Brez ustrezno definirane metodološkega pristopa k načrtovanju in izgradnji obstaja velika verjetnost, da bo projekt zašel v napačne vode, posledica tega pa bo vsaj povečanje stroškov ali podaljšanje roka dokončanja izdelka, če ne kar propad projekta. Da bi se izognili takim črnim scenarijem, smo v članku predlagali izviren pristop k izgradnji spletnih programskih rešitev, ki temelji na splošni metodi z vključenimi ključnimi aktivnostmi, izdelki in splošnimi ter specializiranimi tehnikami, ki se danes najpogosteje uporabljajo za modeliranje posameznih vidikov spletnih programskih rešitev. Predlagana metoda podaja dobre temelje za izgradnjo posameznim organizacijam in projektom prilagojenih metod, glede na njihove specifične karakteristike in okolje delovanja. Nadaljnje delo bo šlo v smeri prenosa predlaganega pristopa (ogrodja in metode) iz teorije v prakso, kar bo zahtevalo najprej razvoj orodja v obliki spletne programske rešitve za obvladovanje komponent metode in gradnjo ter prilagajanje situacijskih metod, nato pa še preizkus opredeljenih situacijskih metod na konkretnih projektih razvoja spletnih programskih rešitev.

Literatura

- [1] Conallen J.: Building Web Applications with UML 2nd edition. Boston: Addison-Wesley, 2003. 468 str.
- [2] Kiely G., Fitzgerald B.: An investigation of the information systems development environment: the nature of development life cycles and the use of methods. Eighth Americas Conference on Information Systems, Baylor, USA, 2002, str. 1289-1296.

- [3] Powel T.: Web Designm: The Complete Reference Second Edition. New York: Osborne/McGraw-Hill, 2002. 901 str.
- [4] Lahajnar S.: Iterativni objektni pristop h gradnji spletnih in mobilnih programskih rešitev. Doktorska disertacija, Ljubljana EF, 2008, 313 str.
- [5] Brinkkemper S.: Method engineering: Engineering of information systems development methods and tools. *Information and Software Technology*, 38(4), 1996, str. 275–280.
- [6] Wistrand K, Karlsson F.: Method Components - Rationale Revealed. The 16th International Conference on Advanced Information Systems Engineering (CAISE 2004), Riga, Latvia, 2004.
- [7] Eriksson H.-E, Penker M.: Business modeling with UML: business patterns at work. New York: John Wiley & Sons, 2000. 459 str.
- [8] List B., Korherr B.: A UML 2 Profile for Business Process Modelling. Proceedings of the 1st International Workshop on Best Practices of UML (BP-UML 2005), 24th International Conference on Conceptual Modeling (ER 2005), Springer, 2005.
- [9] Melia S., Gómez J.: Applying WebSA to a case study: A travel agency system. Proceedings of the Workshop on Model-driven Web Engineering (MDWE 2005), Sydney, Australia, 2005.
- [10] Koch N., Kraus A.: The expressive Power of UML-based Web Engineering. Second International Workshop on Web-oriented Software Technology (IWWOST 02), 2002, str. 105-119.
- [11] Finkelstein A. C. W.: Ubiquitous Web Application Development – A Framework for Understanding. The 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida, USA, 2002.
- [12] Naiburg J. E., Maksimchuk A. R.: UML for Database Design. Boston: Addison-Wesley, 2001. 300 str.
- [13] Ambler W. S.: A UML Profile for Data Modeling. 2002. [URL:<http://www.agiledata.org/essays/umlDataModelingProfile.html>]
- [14] Carlson D. A.: Modeling XML applications with UML: practical e-business applications. Boston: Addison-Wesley, 2001. 333 str.
- [15] Bernauer M., Kappel G., Kramler G.: A UML Profile for XML Schema, Technical Report. 2003. [URL: <http://www.big.tuwien.ac.at/research/publications/2003/1303.pdf>]
- [16] Routledge N., Bird L., Goodchild A.: UML and XML Schema. Proceedings of the Thirteenth Australasian Database Conference, 2002, str. 157-166.
- [17] Karlsson F., Agerfalk P.: Multi-Grounded Action Research in Method Engineering: The MMC Case. IFIP Situational Method Engineering: Fundamentals and Experiences. Springer Boston, 2007, str. 19-32.

Sebastian Lahajnar je leta 1997 diplomiral na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Po diplomi je vpisal podiplomski študij na Ekonomski fakulteti, smer informacijsko upravljavske vede, in leta 1999 zagovarjal magistrsko delo z mentorico prof. dr. Borko Jerman Blažič. Leta 2008 je pod vodstvom prof. dr. Andreja Kovačiča doktoriral na Ekonomski fakulteti z delom Iterativni objektni pristop h gradnji spletnih in mobilnih programskih rešitev. Zaposlen je kot projektant v podjetju PRIS Inženiring, kjer se ukvarja z načrtovanjem in razvojem poslovnih informacijskih sistemov.

Andrej Kovačič je redni profesor in predstojnik Inštituta za poslovno informatiko na Ekonomski fakulteti Univerze v Ljubljani. Je predavatelj predmetov s področja prenove poslovnih procesov in informatizacije poslovanja. Pred tem je bil več let direktor svetovalnega podjetja, projektant in svetovalec pri projektih strateške prenove in informatizacije poslovanja. Vodil ali izvajal je večje število projektov s tega področja v gospodarstvu in v javni upravi. Je veščak Zveze ekonomistov Slovenije na področju upravljanja, pooblaščen revizor informacijskih sistemov ter svetovalec pri mednarodnih projektih PHARE. Je odgovorni urednik revije Uporabna informatika. V zadnjih dveh letih je soorganiziral in predsedoval mednarodni poslovni konferenci Management poslovnih procesov.