

Implementacija metodologije ekstremnog programiranja u nastavni proces visokoobrazovnih institucija

Autori: Tomislav Gligora, Veleučilište Velika Gorica

Davorin Valenčić, Veleučilište Velika Gorica

Aleksander Radovan, Veleučilište Velika Gorica

Sažetak

Ubrzani razvoj softvera (eng. Rapid Application Development) postaje sve prisutniji i potrebniji u projektima tvrtki iz područja informacijskih i komunikacijskih tehnologija, a tim više tvrtki kojima informacijske i komunikacijske tehnologije nisu osnovna djelatnost, a u cilju produktivnijeg razvoja softvera. Primjena agilnih metoda omogućava takav pristup razvoju softvera.

U radu je opisana jedna od najpopularnijih agilnih metoda, metodologija ekstremnog programiranja i njena primjena u nastavnom procesu visokoobrazovnih institucija. U radu je obrađeno planiranje, dizajn, kodiranje i testiranje softvera u ekstremnom programiranju. Kao razvojna platforma za implementaciju ekstremnog programiranja u nastavni proces visokoobrazovnih institucija korišten je Microsoft Visual Studio Lightswitch 2011.

Uvod

U posljednjih 30 godina pokušalo se primijeniti veliki broj različitih pristupa u razvoju softvera.

Agilne metode su se razvijale sredinom 90-tih godina 20. stoljeća. kao djelomična reakcija na tzv. visoko formalne metode (eng. *high ceremony methods*).

Agilne metode razvoja softvera pokušavaju ponuditi odgovor na želju poslovne zajednice za manje opsežnim metodologijama razvoja softvera, koje sa sobom donose brže, žustre procese razvoja softvera [1].

U softverskom inženjerstvu, agilne metode razvoja softvera su manje opsežne metode koje prihvaćaju činjenicu da je softver teško kontrolirati [2]. Primjenom agilnih metoda smanjuje se rizik uspješnosti projekta na način da su razvojni inženjeri fokusirani na male jedinice posla.

Glavne karakteristike agilnih metoda su jednostavnost i brzina. U procesu razvoja softvera, razvojni tim je fokusiran isključivo na funkcije najvišeg prioriteta i na

njihovu implementaciju, potom na brzu isporuku rješenja, dobivanje povratne informacije od korisnika (naručitelja), te reakcije na primljene informacije.

Metodologija ekstremnog programiranja

Ekstremno programiranje je discipliniran pristup razvoju softvera [2]. Ovaj pristup je star petnaestak godina i dokazan je u mnogim kompanijama različitih veličina te u industriji u zemljama širom svijeta [3].

Temeljna vrijednost metodologije ekstremnog programiranja je zadovoljstvo korisnika (naručitelja) softvera rješavanjem ulaznih zahtjeva. Ti ulazni zahtjevi su ispunjeni ukoliko je korisnik zadovoljan sa implementiranim rješenjima. Kao i u svakom projektu, tako i u projektima razvoja softvera, krajnji cilj je zadovoljstvo korisnika, tim više u projektima razvoja softvera jer nakon isporuke softvera projektni timovi ulaze u fazu održavanja.

Primjenom ove metodologije posebno je naglašen timski rad. Programeri, korisnici softvera i menadžeri su dio zajedničkog tima koji je zadužen za isporuku kvalitetnog softvera. Metodologija ekstremnog programiranja implementira jednostavan, ali efikasan način razvoja softvera koji se temelji na grupama (parovima). Programeri rade u parovima i razvijaju testove prije pisanja koda.

Karakteristike modela ekstremnog programiranja su [4]:

1. Igra planiranja (engl. *Planning game*)
2. Malene česte isporuke (engl. *Small/short releases*)
3. Organizacija sustava s metaforama (engl. *Methaphor*)
4. Jednostavan dizajn (engl. *Simple Design*)
5. Testiranje (engl. *Testing*)
6. Korištenje tehnike refaktoriranja (engl. *Refactoring*)
7. Programiranje u paru (engl. *Pair Programming*)
8. Zajedničko dijeljenje kôda (pristup kôdu) (engl. *Collective Ownership*)
9. Kontinuirana integracija (engl. *Continuous integration*)
10. 40 satni radni tjedan (engl. *40-hours week*)
11. Povratna informacija od korisnika (naručitelja) (engl. *On-site customer*)

12. Standardi kodiranja (engl. *Coding Standards*)

Karakteristike metodologije ekstremnog programiranja razvoja softvera razvrstane su u četiri tipične faze razvoja softvera [2]:

1. Planiranje razvoja
2. Dizajn softvera
3. Kodiranje softvera
4. Testiranje softvera

Planiranje razvoja primjenom metodologije ekstremnog programiranja

Planiranje razvoja softvera podrazumijeva definiranje zahtjeva koji se pretvaraju u korisničke priče, planiranje isporuke softverskog rješenja na nivou cijelog projekta, te kreiranje plana iteracije za svaku pojedinu iteraciju u iterativnom razvoju softvera.

Dizajn softvera primjenom metodologije ekstremnog programiranja

Primjenom metodologije ekstremnog programiranja prvenstveno se naglašava korištenje jednostavnog dizajna, korištenje metafore u svrhu pojednostavljivanja slike softvera, korištenje CRC karata (eng. *class-responsability-collaboration*) za timski dizajn sustava. Dodavanje dodatnih funkcionalnosti maksimalno se izbjegava sve do trenutka kad ta funkcionalnost postaje ključna za funkcioniranje cijelog softverskog rješenja.

Kodiranje softvera primjenom metodologije ekstremnog programiranja

Pisanjem programskog kod (kodiranje) u ekstremnom programiranju naglašava se važnost prisutnosti i dostupnost korisnika razvojnom timu, postojanje standarda pisanja programskog koda, implementacija testova prije implementacije samog koda, opis tehnike pisanja koda, te provođenje kontinuirane integracije. Programeri u ovoj fazi stalno proučavaju tuđi kod ukoliko ne postoji standard pisanja koda. Standard pisanja koda ne smije trošiti puno vremena, u tom smislu je jako važan odabir razvojne tehnologije kojom će se razvijati softver.

Testiranje softvera primjenom metodologije ekstremnog programiranja

U ekstremnom programiranju testovi se mogu podijeliti u dvije grupe:

1. Jedinični testovi (eng. *Unit Test*)

Ovom vrstom testiranja provjerava se ispravnost rada sustava. Svakim novim jediničnim testom softver mora raditi. Jedinični testovi daju programerima sigurnost prilikom razvoja novih funkcionalnosti softvera, te objašnjavaju kako se kod koristi. Moraju biti automatizirani kako bi se omogućila jednostavna ponovljivost.

2. Testovi prihvata (eng. *Acceptance*)

Testom prihvata provjerava se da li softver (dio softvera) radi ono što je naručitelj zahtijevao (opisano korisničkim pričama). Definiraju da li sustav zadovoljava kriterij prihvaćenosti i omogućuju korisniku da odluči može li prihvatiti softversko rješenje.

Osnovni nedostaci primjene metodologije ekstremnog programiranja:

- prvenstveno namijenjena malim timovima programera (2-12)
- nije primjenjiva na svim projektima razvoja softvera
- postoji jak otpor ekstremnom načinu programiranja
- neprikladna ukoliko je potrebno generirati velike količine dokumentacije
- ponekad je prekovremeni rad nužan na projektima, ova metodologija takav način rada ne podupire
- na nekim projektima ljudi ne žele komunicirati iz raznih razloga
- nemogućnost dužeg čekanja povratnih informacija
- XP nije pogodan za klase produkata: sigurnosne, dugog života i modularne

Primjena ekstremnog programiranja u nastavnom procesu

Današnje potrebe tržišta informacijskih i komunikacijskih tehnologija, odnosno u vremenu sve produktivniji pristup razvoju softverskih rješenja zahtjeva znanja bržeg i jednostavnijeg razvoja softvera. U tom smislu, primjena agilnih metodologija u nastavni proces visokoobrazovnih institucija nameće se kao nužna potreba.

Metodologija ekstremnog programiranja iznimno je prikladna za primjenu u nastavnom procesu zbog nekoliko ključnih karakteristika:

1. Jednostavnost primjene
2. Brzina razvoja
3. Vizualizacija rješenja

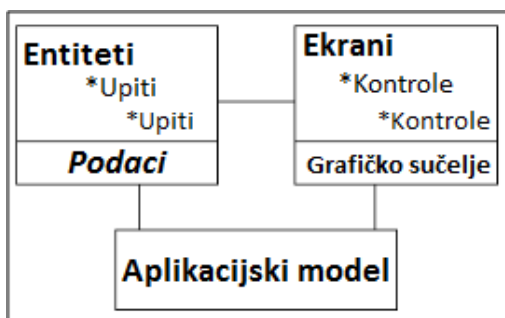
Prilikom odabira razvojne tehnologije potrebno je voditi računa o svim navedenim ključnim karakteristikama metodologije ekstremnog programiranja. Autori ovog rada odabrali su razvojnu tehnologiju Microsoft Visual Studio Lightswitch 2011 jer u potpunosti zadovoljava osnovne karakteristike metodologije ekstremnog programiranja.

Microsoft Visual Studio Lightswitch 2011

Microsoft Visual Studio Lightswitch 2011(*Lightswitch*) je razvojni alat za primjenu ubranog razvoja softvera (eng. *Rapid Application Development*) koji se temelji na nekoliko dokazanih Microsoft tehnologija, poput ASP.NET, Microsoft Silverlight, WCF RIA, ADO.NET, MVVM, VPF [5]. Lightswitch je zapravo ekstenzija Visual Studia, koja može biti korištena kao zaseban razvojni alat ili kao dodatak na Visual Studio. Primjenom ovog razvojnog alata na brz i jednostavan način moguće je razviti tri vrste profesionalnih poslovnih softvera:

1. Softver za Windows okruženje
2. Internet softver
3. Softver u oblaku

Neovisno kakva vrsta softvera se razvija pristup razvoju i razvojno okruženje je identično za programere. Lightswitch podržava pisanje programske logike u C# i Visual Basic programskim jezicima i korištenje .NET frameworka. Arhitektura Lightswitcha zasnovana je na aplikacijskom modelu koji predstavlja centralnu jedinicu aplikacije koja se razvija[6]. Lightswitch razvoj softvera se u osnovi sastoji od definiranja entiteta i ekrana nad entitetima. Aplikacijski model se kreira korištenjem aplikacijskih blokova. Entiteti se kreiraju iz tipova entiteta (eng. *Entity Type*), a upiti (eng. *Query*) izvlače podatke iz entiteta. Ekрани se sastoje od niza kontrola.



Slika 1. Arhitektura Lightswitch softvera

Primjena Microsoft Visual Studio Lightswitcha u nastavnom procesu

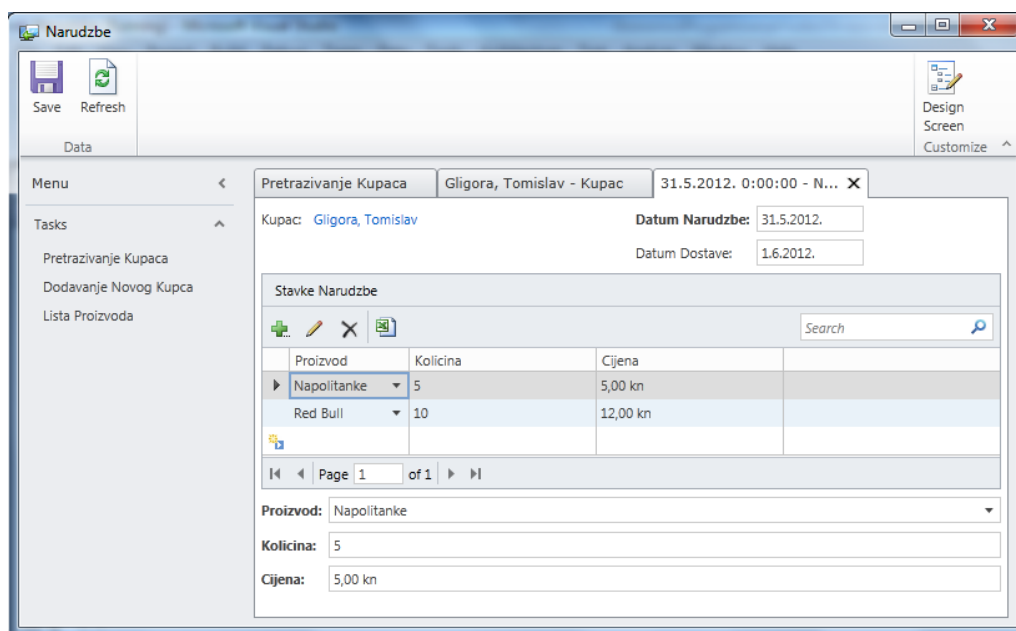
Prilikom razvoja Lightswitch softvera sve karakteristične faze ekstremnog programiranja primjenjuju se na vrlo jednostavan način.

Faza planiranja je vrlo jednostavna jer svi slojevi aplikacije dostupni na jednom mjestu. Planiranje razvoja svih slojeva softvera i budućeg testiranja je centralizirano i veoma brzo. Dizajn softvera je također jednostavan, jer Lightswitch ima posebnu ljusku (eng. *Shell*) koja je dostupna kroz Entity Designer i Screen Designer, a sastoji se od niza gotovih predložaka ekrana koji se automatizmom spajaju na moguće izvore podataka (eng. *Data Source*). Nakon kreiranog podatkovnog modela i razvijenog ekrana softvera, može se pristupiti fazi kodiranja. Lightswitch u potpunosti podržava razvoj softvera troslojne arhitekture (eng. *Three-tier Architecture*), te izgradnjom potrebnih entiteta i ekrana automatski kreira veliki broj klasa (eng. *Class*) i događaja (eng. *Event*). U kombinaciji sa .NET frameworkom, ovakav pristup uvelike standardizira način pisanja programskog koda. Gotovo softversko rješenje se prema unaprijed definiranom planu testiranja iznimno jednostavno testira. Jedinični testovi su djelomično pokriveni kontrolnim mehanizmom Lightswitcha (tzv. *Debugger*), a djelomično podatkovnim testiranjem ekrana, dok su testovi prihvata u potpunosti pokriveni kontrolom željene funkcionalnosti koja je ugrađena u softver.

Iskustva u korištenju Microsoft Visual Studio Lightswitcha u nastavnom procesu

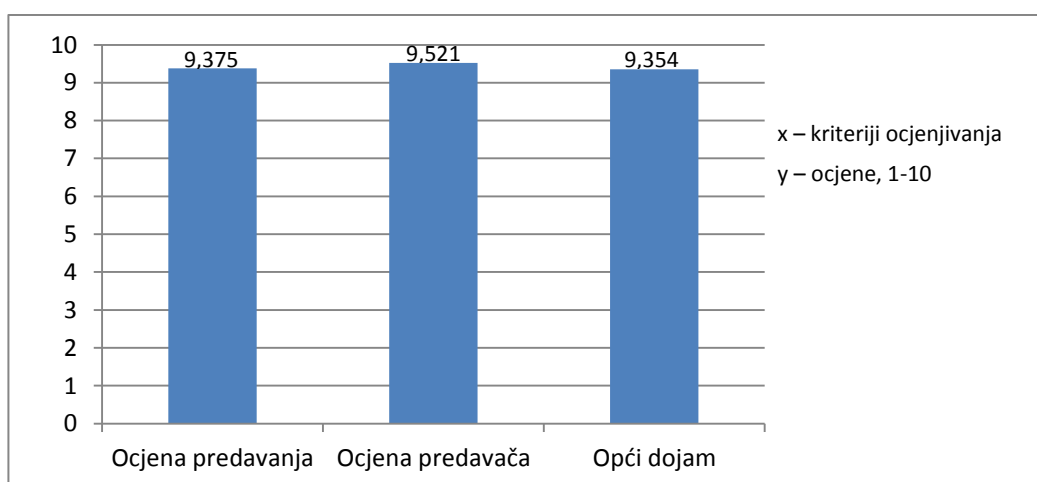
Autori su u suradnji sa studentima proveli analizu prihvata ovakvog pristupa razvoja softvera u nastavnom procesu. Način provedbe analize je bio kreiranje

softvera za evidenciju kupaca i proizvoda, te ažuriranje narudžbi proizvoda od strane kupaca. Zadano vrijeme za izgradnju ovakvog softvera bilo je 90 minuta.



Slika 2. Izgled ekrana kreiranog softvera

Nakon kreiranje softvera provedena je anketa među studentima. Anketi je pristupilo 70 studenata. Studenti su na sljedeći način ocijenili primjenu metode ekstremnog programiranja u nastavni proces korištenjem Microsoft Visual Studio Lightswitch razvojnog alata:



Slika 3. Rezultati ankete – Microsoft Visual Studio Lightswitch

Zaključak

Primjena metodologije ekstremnog programiranja u nastavni proces visokoobrazovnih institucija postaje nužna potreba u procesu osposobljavanja budućih programskih inženjera zbog zahtjeva tržišta informacijskih i komunikacijskih tehnologija.

Korištenjem alata Microsoft Visual Studio Lightswitch 2011 u potpunosti su pokrivenе sve karakteristike metodologije ekstremnog programiranja na najjednostavniji mogući način. Jednostavnost i brzina najveće su prednosti ove metodologije, a samim time ovaj razvojni alat nameće se kao jedno od najboljih rješenja. Također, potrebna predznanja su prikladna za studentski nivo, brzina rada je iznimno velika te se veoma brzo vizualizira softvera koji se razvija.

Prilikom primjene metodologije ekstremnog programiranja korištenjem razvojnog alata Microsoft Visual Studio Lightswitch 2011, poželjna, ali ne i nužna, su predznanja objektno orijentiranog programiranja, bazi podataka i programiranja bazi podataka, te programskog jezika C# ili Visual Basic. Reverznim procesom je moguće ta znanja usvajati.

Popis literature

1. Philip G. Armour: The Laws of Software Process: A New Model for the Production and Management of Software, Auerbach Publications, 2004.
2. Pekka Abrahamsson, Outi Salo, Jussi Ronkainen & Juhani Warsta: Agile software development methods (<http://www.inf.vtt.fi/pdf/publications/2002/P478.pdf>)
3. Michele Marchesi, Giancarlo Succi, DonWells, LaurieWilliams: Extreme Programming Perspectives, Addison-Wesley Pub Co, 2002.
4. William C. Wake: Extreme Programming Explored, Addison Wesley Professional, srpanj 2001.
5. Jayaram Krishnaswamy, A jump-start guide to application development with Microsoft's Visual Studio LightSwitch, Packt Publishing, 2011
6. Tim Leung, Yann Duran, Pro Visual Studio Lightswitch 2011 Development, Apress, 2011