Primena agilnih metoda razvoja softvera u mobilnim tehnologijama

Stevan Milovanović
student drugog ciklusa studija
Fakultet organizacionih nauka
Beograd, Srbija
stevanm993@gmail.com

Sažetak—**Tema ovog rada je analiza agilnih metoda razvoja softvera koje se primenjuju prilikom razvoja mobilnih aplikacija. Kako se projektovanje mobilnih aplikacija suočava sa mnogobrojnim ograničenjima specifičnim za razvoj ove vrste aplikacija samim tim je ceo proces usmeren više na unapređenje funkcionalnosti koje se neprestano razvijaju nego na pisanje dokumentacije o softveru. Metodologije koje su se do sada dokazale kao najuspešnije, a to su Ekstremno programiranje, Scrum, Razvoj vođen funkcionalnostima, Mobile-D i Kanban biće predstavljene i analizirane u radu. Cilj ovog rada je predstavljanje komparativnog prikaza agilnih metoda kako bi se njihove prednosti i mane uporedile te kako bi se na osnovu toga identifikovali idealni uslovi za primenu svake od njih.**

Ključne riječi - projektovanje softvera; agilne metode razvoja softvera; mobilne tehnologije;

I. UVOD

Razvoj mobilnih aplikacija suočava se sa mnogobrojnim ograničenjima kako usled ograničenja hardvera tako i usled neprestanog razvoja tržišta mobilnih aplikacija. Tehnološka ograničenja ogledaju se u veoma ograničenom memorijskom prostoru, bezbednosti uređaja, pokrivenošću mrežom, ograničenom veličinom ekrana, ograničenom sposobnošću unosa podataka – usled male tastature, jačinom procesora uređaja, ograničenjima baterije i tako dalje. Što se tiče ograničenja koje nameće tržište ističe se dinamičnost tržišta moblinih aplikacija koja uslovljava primenu agilnih metoda prilikom samog razvoja softvera, takođe klijenti zahtevaju prototipove aplikacija pa se ustalio princip iterativnog isporučivanja softvera tokom razvoja[1]. Ovaj način razvoja uslovljava male i multidisciplinarne timove bez hijerarhije kako bi se poboljšala komunikacija među članovima.

Kada je reč o ulogama u timu, one zavise od metode koja se koristi prilikom razvoja softvera ali svaki tim mora imati projektanta softvera, grafičkog dizajnera, programere koji razvijaju korisnički interfejs (frontend developeri) i programere koji razvijaju serverski deo aplikacije (backend developeri). U svakoj od iteracija razvoja softvera implementira se određeni broj funkcionalnosti kako bi korisnik imao vremena da svaku od implementiranih funkcionalnosti testira nezavisno od drugih. Prva verzija softvera naziva se i MVP proizvod (Minimum Viable Product), odnosno softver koji minimalno zadovoljava funkcionalnu specifikaciju[2]. Ukoliko ovaj proizvod prilikom testiranja osigura validaciju poslovnog modela, dalji razvoj ide u smeru proširenja postojećih i implementacije preostalih funkcionalnosti.

Specifičnost razvoja softvera u mobilnim tehnologijama ogleda se kako u ograničenjima tako i u zahtevima koji teže da u potpunosti iskoriste mogućnosti mobilnih uređaja. Neki do tih zahteva su i sledeći [3] :

- Interakcija među aplikacijama – nasuprot aplikacijama na desktop računarima koje su uglavnom nezavisne od drugih aplikacija instaliranih na računaru, mobilne aplikacije često interaguju jedne sa drugima.
- Obrada senzora – sve je više senzora na mobilnim telefonima (akcelerometar, žiroskop, NFC senzor, barometar, senzor svetlosti, senzor blizine, geomagnetni senzor). Na novijim telefonima (Galaxy S5) dodati su i senzori za merenje brzine otkucaja srca kao i senzori za očitavanje otiska prsta. Zajedno sa ovim senzorima pojavile su se i nove funkcionalnosti kao što su otključavanje uz pomoć otiska, korišćenje geomagnetnog senzora kao kompasa i tako dalje.
- Kreiranje native i hybrid mobilnih aplikacija – u zavisnosti od potreba kao i od budžeta razvijaju se posebne (native) aplikacije za svaku od podržanih platformi ili jedna integralna (hibridna) aplikacija
- Podržavanje ekrana različite veličine i rezolucije kao i podržavanje različitih orijentacija ekrana

II. EKSTREMNO PROGRAMIRANJE

Metodologija ekstremnog programiranja razvijena je u cilju poboljšanja kvaliteta softvera kao i responzivnosti projekata na promene u korisničkim zahtevima. Kao i svaka druga agilna metodologija zasniva se na iterativno inkrementanom razvoju sa jasno definisanim ciljevima svake iteracije. Ova metodologija agilnog razvoja softvera koristi se ukoliko su sledeći preduslovi ispunjeni:

- Projekat, odnosno softver koji je potrebno razviti, nije previše kompleksan
- Softver se razvija za više platformi (Android, iOS, Windows)
- Ravna menadžment struktura u timu koji razvija softver

Glavna odlika ove metode razvoja softvera je programiranje u paru (Pair programming) odnosno međusobno proveravanje izvornog koda između programera koji implementiraju iste funkcionalnosti na različitim platformama[4]. Tom prilikom oba programera rade za istom mašinom, jedan u ulozi programera koji piše kod a drugi u ulozi posmatrača ili supervizora koji nadgleda napisani kod i donosi odluke prilikom implementacije funkcionalnosti. Na taj način posmatrač se može izolovati od implementacionih problema i stvari sagledati sa višeg, konceptualnog nivoa dok implemetacione probleme ostavlja programeru. Akcenat ove metodologije je na jednostavnosti i jasnoći napisanog koda kako bi se omogućila skalabilnost softvera koji se razvija.

Takođe, jedna od odlika ove metodologije je pisanje automatizovanih testova prilikom proveravanja funkcionalnosti sistema. Na kraju svake iteracije klijent testira implementiranu celinu uzimajući u obzir poslovna pravila sistema. Na taj način potvrđuje se završetak svake od iteracija. Nakon uspešno validirane iteracije, funkcionalnosti implementirane u toj iteraciji spajaju se sa ostatkom sistema i nakon toga prolaze integralni test. Ukoliko se prilikom integralnog testa pojavi neka greška, odbacuje se cela iteracija[5].

Prilikom razvoja u timu postoji princip kolektivne odgovornosti, odnosno svaki član tima može menjati bilo koji deo koda u cilju njegovog poboljšanja. Stoga je neophodno da članovi tima na početku projekta ustanove konvencije koje će poštovati prilikom programiranja kako ne bi došlo do nekonzistentnosti u kodu.

Potencijalni problemi usled primene ove metodologije razvoja softvera mogu biti nedostatak multidisciplinarnih stručnjaka unutar tima kao i kompromisi usled nejasnih ili nepotpunih specifikacija klijenata koje dovode do različite implementacije na različitim platformama.

III. SCRUM

Za ovu metodu razvoja softvera najčešće se vezuju složeniji projekti sa većim timovima kako bi se mogle adekvatno podeliti uloge u timu i kako bi posao mogao da se delegira na pravi način[6]. Glavna karakteristika ove metodologije je pretpostavka da klijent, tokom razvoja softvera, može izmeniti unapred definisane zahteve. Tokom razvoja tim se svakodnevno sastaje kako bi jedni drugima predstavili rezultate rada ali i kako bi postavili nove dnevne ciljeve. Prilikom korišćenja Scrum metodologije postoje sledeće uloge u timu:

- Scrum Master – projektant softvera
- Product owner – najčešće klijent
- Team – jedna ili više grupa zadužena za sprovođenje ideja

Zadatak Product Ownera je specifikacija zahteva zajedno sa definisanjem stepena prioreta za svaki od njih. Zadatak tima je da specifikaciju dobijenu od Product Ownera razloži na skupove funkcionalnosti koje se dodeljuju iteracijama. Takođe, neophodno je da tim odredi način implementacije defnisanih funkcionalnosti i na osnovu toga proceni koliko je vremena potrebno za svaku od iteracija. Naposletku, zadatak Scrum Mastera je da rukovodi timom i posreduje između tima i Product Ownera.

Proces razvoja podeljen je po iteracijama koje se nazivaju Sprint-ovi. Iteracije teže da imaju istu dužinu kako bi se obezbedio kontinuitet razvoja. Pri svakoj od iteracija razvijaju se prethodno definisane funkcionalnosti tako da se posle svake iteracije proizvod može isporučiti klijentu. Funkcionalnosti su predefinisane za svaku od iteracija kako bi se što jasnije mogao pratiti napredak i uspešnost svake od iteracija.

Ova metodologija je pogodna za razvoj softvera u mobilnim tehnologijama jer zahvaljujući svojoj prilagodljivosti promenama prema zahtevima klijenata ima sposobnost prilagođavanja na česte promene zahteva platformi (razvoj operativnih sistema, dodavanje novih funkcionalnosti kao što su otisak prsta, NFC sensor i tako dalje). Ukoliko se softver razvija za više platformi (iOS i Android) za svaku platformu se kreiraju posebni timovi.

Glavni nedostatak ove metodologije predstavlja loše skaliranje u odnosu na kompleksnost projekta. Ukoliko projekat postane isuviše kompleksan može doći do problema u koordinaciji, podeli posla po iteracijama, kao i u struktuiranju samog tima. Takođe, usled česte promene zahteva projekti razvijani uz pomoć ove metodologije ne mogu predefinisati datum finalizacije proizvoda.

IV. RAZVOJ VOĐEN FUNKCIONALNOSTIMA

Ova metoda razvoja softvera predstavlja iterativni, inkrementalni proces razvoja softvera koji objedinjuje najbolje prakse iz industrije. Sastoji se iz pet aktivnosti [7]

1) *Razvoj modela*
2) *Kreiranje detaljne specifikacije funkcionalnosti*
3) *Planiranje na osnovu specifikacije funkcionalnosti*
4) *Dizajniranje na osnovu specifikacije funkcionalnosti*
5) *Razvoj na osnovu specifikacije funkcionalnosti*

Tokom prve dve aktivnosti razvija se celokupan model. Kroz preostale aktivnosti se prolazi iterativno za svaku od funkcionalnosti. Prilikom primene ove metodologije u timu se definišu sledeće uloge:

- Project Manager – Lider projekta
- Chief Arhitect– Projektant softvera
- Development Manager – Glavni odgovorni za implementaciju isprojektovanog rešenja
- Chief Programmer – definiše funkcionalnosti koje će biti implementirane u svakoj iteraciji razvoja. Kreira sekvencijalni dijagram za svaku od aktivnosti koja treba biti implementirana kako bi Class Owner izvršio zadatak prema unapred definisanoj, detaljnoj specifikaciji
- Class Owner – programer koji izvršava unapred defnisane zadatke (razvija funkcionalnosti samo u okviru svoje klase)

- Domain Expert – osoba upoznata sa prirodom problema, najčešće klijent. Učestvuje samo u prvoj fazi projekta

Na ovaj način aktivnosti su delegirane kroz više nivoa a kako se svaka od funkcionalnosti razvija zasebno olakšano je testiranje i identifikovanje eventualnih nepravilnosti. Mehanizam ove metodologije umnogome podseća na verzionisanje koda. Svaka od funkcionalnosti predstavlja zasebnu granu tako da je ceo sistem decentralizovan prilikom razvoja da bi se na kraju sve grane sjedinile kako bi se kreirao finalni proizvod.

Prednosti ove metodologije ogledaju se u jasnoj dekompoziciji zadataka tako da je smanjena potreba za intenzivnom komunikacijom pa se samim tim štedi na vremenu neophodnom za finalizaciju projekta. Takođe, svaka od funkcionalnosti se testira odmah nakon implementacije nezavisno od ostalih tako da se eventualne greške mogu uočiti i ispraviti na vreme.

Najveći nedostatak ove metodologije ogleda se u njenoj zahtevnosti. Naime, podela posla nije rezultat sastanaka i zajedničkog odlučivanja kao kod nekih drugih metodologija, već o tome odlučuje osoba koja projektuje sistem. Stoga, uspešnost celog projekta može zavisiti od stručnosti i iskustva jedne osobe.

V. MOBILE – D

Mobile – D metodologija razvoja softvera predstavlja metodologiju razvoja specijalizovanu za razvoj mobilnih aplikacija i predstavljena je u radu [8]. Ova metodologija bazirana je na Extreme Programming metodologiji kao i na Crystal metodologijama. Principi ove metodologije su razvoj vođen testiranjem, programiranje u paru i iterativni razvoj i integracija. Sastoji se od pet faza razvoja softvera:

1) *Istraživanje*
2) *Kreiranje*
3) *Proizvodnja*
4) *Stabilizacija*
5) *Testiranje i ispravka sistema*

U prvoj fazi neophodno je napraviti plan razvoja, postaviti generalne karakteristike sistema i definisati procese razvoja funkcionalnosti. Nakon toga, u fazi kreiranja obezbeđuju se ključni resursi neophodni za razvoj sistema kao što su tehnološki, komunikacioni ili infrastruktura sistema. Ova faza podeljena je na tri dela, postavljanje projekta, inicijalno planiranje i dan za ispitivanje [9]. Sledeća faza, faza proizvodnje, obuhvata implementacione aktivnosti. Ova faza je podeljena na dane za planiranje, dane za proizvodnju i dane za distribuciju (release). Dani za planiranje usmereni su na poboljšanje procesa razvoja u smislu definisanja prioriteta i analiziranja funkcionalnosti. Takođe, u ovom periodu trebalo bi osmisliti testove koji će biti sprovedeni kako bi se utvrdila validnost implementiranih funkcionalnosti. U danima za proizvodnju, korišćenjem razvoja vođenog testiranjem, implementiraju se funkcionalnosti na osnovu prethodno ustanovljenog plana. Na kraju, u danima za distribuciju, uz pomoć već osmišljenih testova, validiraju se implementirane funkcionalnosti. Poslednje dve faze, stabilizacija i testiranje i ispravka sistema, koriste se za finalizaciju proizvoda i pojedinačno testiranje svake od funkcionalnosti.

Glavna prednost ove metodologije je to što je nastala za potrebe razvoja softvera u mobinim tehnologijama tako da unutar svojih faza i procesa inkorporira sve specifičnosti mobilnih platformi.

Neki od problema sa kojima je moguće suočiti se prilikom korišćenja ove metodologije su nedostatak adekvatnih primera primene kao i prekoračenje definisanih rokova za finalizaciju projekta usled raznih faktora kao što su izmena zahteva, unapređenje platformi ili samih uređaja.

VI. KANBAN

Kanban metodologija nastala je u Toyota kompaniji kako bi se optimizovali proizvodni procesi ali je našla primenu i pri razvoju softvera između ostalog zbog svog principa dostavljanja proizvoda u pravo vreme („just in time“). Prilikom razvoja timovi su usredsređeni samo na zadatke koji su u toku. Na taj način omogućeno je fokusiranje na tekuće, konkretne probleme i izolovanje od kompleksne celine. Samim tim zadaci planiranja i implementacije se usmeravaju na konkretne probleme. Ova metodologija posebno je primamljiva klijentima zbog svoje transparentnosti. Naime, na japanskom pojam „kanban“ označava vizuelni signal [10] stoga prilikom razvoja tim koristi panel za vizuelizaciju posla sa sekcijama za predstojeće, zadatke kojima se trenutno bave i već urađene zadatke. Na panelu su pojedinačni zadaci, koji predstavljaju zasebne celine, prikazani kao kartice. Panel pomaže razvojnom timu kako bi bolje planirao i organizovao svoje obaveze, dok klijentima pruža uvid u sveukupni napredak projekta kako bi u svakom trenutku bili upućeni u tok razvoja. Samim tim se odstranjuje potreba za održavanjem sastanaka između tima i klijenata (ili menadžmenta kompanije). Iako neke kompanije koriste fizičke panele sve više se kao rešenje nameću razni web alati kao na primer Trello, JIRA, Kanbanize i tako dalje.

Prednosti ove metodologije su fleksibilnost prilikom planiranja jer je tim usmeren samo na tekuće zadatke. Takođe, menadžment raspoređivanjem kartica omogućava izvršavanje zadataka po prioritetu stoga nema potrebe za iteracijama prilikom razvoja kao u ostalim agilnim metodologijama. Ipak vizuelizacija podataka predstavlja glavnu prednost ove metodologije. Predstavljanjem napretka na panelu lakše se mogu uočiti procesi koji izazivaju poteškoće prilikom razvoja. Mogu se identifikovati uzroci zagušenja i nakon toga se blagovremeno otkloniti[11].

Jedan od glavnih problema sa kojima je moguće suočiti se prilikom korišćenja ove metodologije je zagušenje procesa razvoja ukoliko timovi nisu dovoljno raznovrsni. Na primer, ukoliko je projekat kompleksan u pogledu grafičkog korisničkog interfejsa a tim raspolaže sa jednim grafičkim dizajnerom ceo razvoj se može usporiti. Takođe, teško je predvideti rokove jer nisu predefinisani. Koncept metodologije

se zasniva na kompletiranju zadataka bez vremenskih ograničenja. Naposletu, ukoliko se paneli kreiraju za više timova unutar projekta (tim dizajnera, programera i tako dalje) koordinacija panela može predstavljati ozbiljan izazov.

VII. ZAKLJUČAK

U ovom radu predstavljene su najzastupljenije agilne metode razvoja softvera sa osvrtom na njihovu primenu prilikom razvoja softvera specifično za mobilne uređaje. Prilikom odabira metodologija neophodno je bilo uzeti u obzir ograničenja mobilnih uređaja, broj različitih platformi koje podržavaju mobilne aplikacije kao i veliki raspon verzija operativnih sistema podržanih od strane pomenutih platformi. Pored ograničenja, konstantan rast i razvoj tržišta ovih aplikacija uticao je kako na modifikovanje postojećih tako i na razvoj novih metodologija specifičnih za ovu vrstu tehnologija. Prilikom analiziranja metodologija naglašene su prednosti i nedostaci svake od njih kao i situacije u kojima pomenuti pristupi daju najbolje rezultate. U tabeli I. predstavljen je uporedni prikaz prednosti i nedostataka analiziranih metodologija.

TABELA I. UPOREDNI PRIKAZ AGILNIH METODA RAZVOJA SOFTVERA U MOBILNIM TEHNOLOGIJAMA

Metodologija	Prednosti	Nedostaci
Ekstremno programiranje	• Robustna metodologija otporna na promene zahteva korisnika • Istovremena implementacija i testiranje uz pomoć automatizovanih testova • Manji rizik pri implementaciji usled programiranja u paru	• Nedostatak multidisciplinarnih stručnjaka unutar tima • Nekonzistentnost implementacije na različitim platformama
Scrum	• Podela posla u manje jedinice – Sprintove • Jasno praćenje napretka projekta kroz Scrum sastanke • Usvaja povratne informacije klijenata tokom sastanaka	• Usled česte promene zahteva problem sa poštovanjem defnisanih rokova • Koordinacija timova postaje izazovna na velikim projektima (problem skalabilnosti) • Česti sastanci zahtevaju mnogo vremena
Razvoj vođen funkcionalnostima	• Jasna dekompozicija zadataka smanjuje potrebu za intenzivnom komunikacijom • Olakšano testiranje usled validacije svake funkcionalnosti nezavisno od ostalih • Rezultat svake iteracije je skup funkcionalnosti koji predstavlja proizvod isporučiv klijentu	• Uspešnost celog projekta može zavisiti od stručnosti i iskustva jedne osobe • Usled dekompozicije projekta otežana je koordinacija nezavisnih celina
Mobile – D	• Inkorporira sve specifičnosti moblinih platformi • Konstantan napredak i učenje prilikom razvoja • Mogućnost kreiranja platformski nezavisnog rešenja	• Nedostatak adekvatnih primera primene • Prekoračenje definisanih rokova za finalizaciju projekta • Definisanje testova u ranoj fazi planiranja zahteva sagledavanje problema sa konceptualnog nivoa
Kanban	• Fleksibilnost prilikom planiranja jer je tim usmeren samo na tekuće zadatke • Mogućnost definisanja prioritetnih zadataka • Vizuelizacija toka projekta	• Zagušenje procesa razvoja usled nedovoljne multidisciplinarnosti u timu • Teško je predvideti rokove jer nisu predefinisani. Akcenat nije na vremenskom intervalu već na ispunjenju zadataka

Na osnovu predstavljene tabele možemo zaključiti da su značajni faktori pri izboru metodologije konzistentnost korisničkih zahteva i kompleksnost celokupnog projekta. Ukoliko se može računati na konzistentnost dokumentacije prilikom rada na kompleksnom projektu najbolji izbor bila bi metodologija razvoja vođenog funkcionalnostima zbog dekompozicije kompleksnosti i jasno definisanih procesa razvoja. Nasuprot tome, ako su zahtevi klijenata nestabilni (skloni čestim izmenama) ali i ukoliko projektni tim poseduje dovoljno iskustva u koordinaciji iteracija idealno rešenje predstavlja Scrum metodologija.

Sledeći skup identifikovanih faktora prilikom izbora metodologije predstavlja broj platformi na kojima isprojektovano rešenje treba biti implementirano i iskustvo tima. Ukoliko je cilj podržavanje svih dostupnih platformi sa timom koji poseduje potrebna znanja i iskustvo kao najbolje rešenje nameće se metodologija Ekstremnog programiranja. Ukoliko je projektni tim sačinjen od mladih i ambicioznih članova, željnih novih znanja, korišćenje jedne nove metodologije kao što je Mobile – D predstavlja sjajnu priliku za razvoj inovativnog, platformski nezavisnog, sistema.

Na kraju, kao još jedan bitan faktor ističe se uključenost klijenata, odnosno menadžmenta u proces razvoja. Ukoliko je neophodna svakodnevna interakcija razvojnog tima sa klijentima kao najbolji pristup predstavljena je Scrum metodologija zbog svakodnevnih sastanaka radi predstavljanja sveukupnog napretka projekta. U suprotnom, najbolje rešenje predstavlja Kanban metodologija jer uz pomoć vizuelizacije toka projekta menadžment i razvojni tim mogu funkcionisati gotovo nezavisno.

ZAHVALNICA

Ovaj rad je napisan kao seminarski rad u okviru predmeta „Izabrana poglavlja iz elektronskog poslovanja" katedre za elektronsko poslovanje na Fakultetu organizacionih nauka u Beogradu. Zahvaljujem nastavnom osoblju katedre na pomoći pri izradi rada.

LITERATURA

[1] Asra, K., Sobia, Z., & Muhammad Fahad, K. (2014). Suitability and Contribution of Agile Methods in Mobile Software Development. International Journal Of Modern Education And Computer Science, 2014, Vol 6, Iss 2, Pp 56-62 (2014), (2), 56. doi:10.5815/ijmecs.2014.02.08

[2] Codington-Lacerte, C. Agile software development. Salem Press Encyclopedia, 2016.

[3] Corral, L., Sillitti, A., & Succi, G. Software assurance practices for mobile applications. Computing, 2015, 97(10), 1001-1022. doi:10.1007/s00607-014-0395-8

[4] Concas, G. Agile Processes in Software Engineering and EXtreme Programming : 8th International Conference, XP 2007, Como, Italy, June 18-22, 2007: Proceedings. Berlin: Springer.

[5] R. C. Martin, Agile Software Development Principles, Patterns and Practices, Prentice Hall, Oct 25 2002.

[6] K. Schwaber, Agile project management with Scrum. Redmond, WA: Microsoft Press, 2004.

[7] S. R. Palmer and J. M. Felsing, A practical guide to feature-driven development. Upper Saddle River, NJ: Prentice Hall PTR, 2002.

[8] P. Abrahamsson, A. Hanhineva, H. Hulkko, T. Ihme, J. Jäälinoja, M. Korkala, J. Koskela, P. Kyllönen, O. Salo, Mobile-D: an agile approach for mobile application development, Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications, October 24-28, 2004, Vancouver, BC, CANADA [doi>10.1145/1028664.1028736]

[9] A. C. Spataru, Agile Development Methods, Edinburgh, 2010.

[10] M. O. Ahmad, J. Markkula and M. Oivo, "Kanban in software development: A systematic literature review," in *EUROMICRO Conference*, Oulu, 2013.

[11] Lei, H., Ganjeizadeh, F., Jayachandran, P. K., & Ozcan, P. Full length Article: A statistical analysis of the effects of Scrum and Kanban on software development projects. Robotics And Computer Integrated Manufacturing, February 1, 2017, 43(Special Issue: Extended Papers Selected from FAIM 2014), 59-67. doi:10.1016/j.rcim.2015.12.001

ABSTRACT

The theme of this work is the analysis of agile software development methods that are applied in the development of mobile applications. Because the design of mobile applications cope with the numerous constraints specific to the development of these types of applications, and therefore the whole process is aimed more at improving the functionality that is constantly evolving rather than on writing documentation about the software. The methods that have so far proven to be the most successful, such as Extreme Programming, Scrum, Feature Driven Development and Mobile – D, will be presented and briefly analyzed in this paper. The aim of this paper is to present a comparative view of methods presented in order to compare the advantages and disadvantages of each method.

Keywords - software development, agile methods of software development, mobile technologies;

PAPER TITLE

The application of agile methods of software development in mobile technologies

Author name(s)

Stevan Milovanović