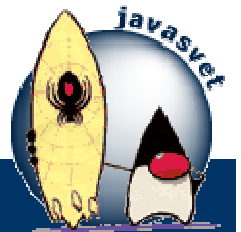


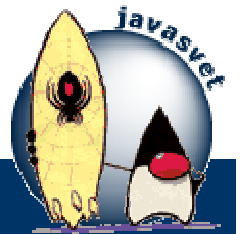
Agilne metodologije primer iz prakse

Predravač: Predrag Spasojević
Sportska kladionica Mozart



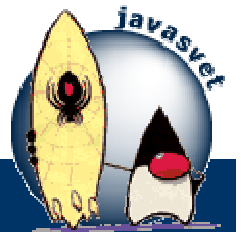
Cilj prezentacije

Prenošenje iskustva u praktičnoj primeni
agilnih metodologija



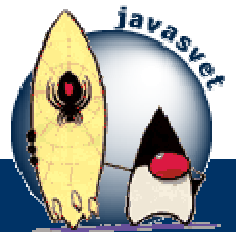
Agenda

- Zašto agilno?
- Iterativnost i evolutivnost
- Uloge
- Proces
- Planiranje i praćenje projekta
- Problemi



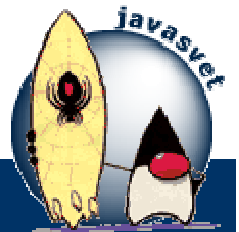
Metodologije

- Waterfall
- Rational Unified Process (RUP)
- Agile :XP, Scrum,...



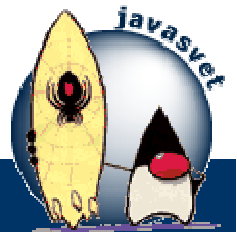
Agile manifesto

- Pojedinci i interakcije pre nego procesi i alati
- Software koji radi pre nego obimna dokumentacija
- Saradnja sa klijentima pre nego pregovori
- Odgovor na promenu pre nego praćenje plana

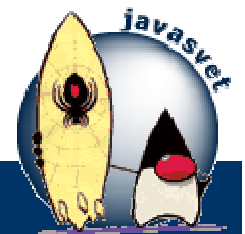
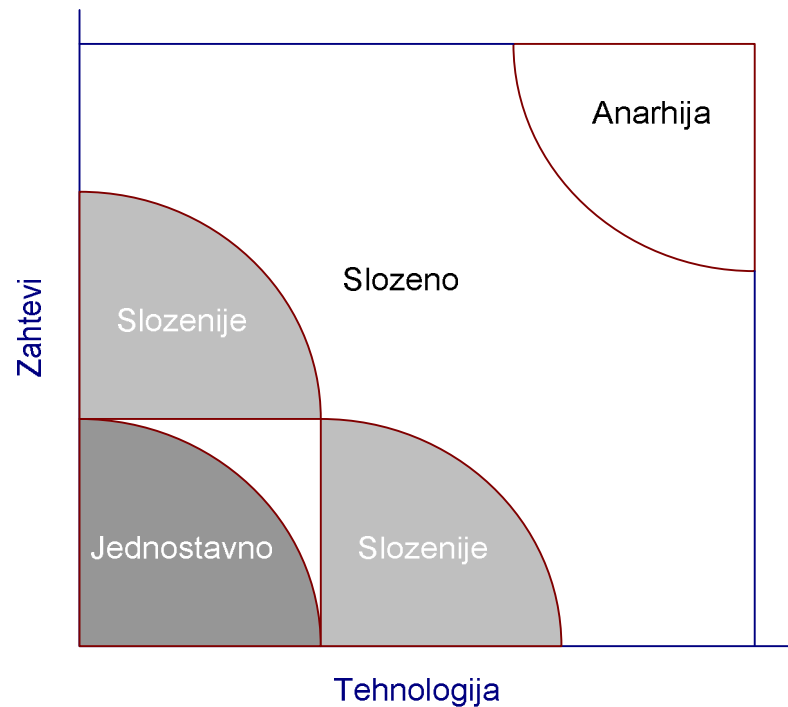


Agile

- Pokušaj da se razvojni proces prilagodi klijentu a ne klijent razvojnem procesu
 - Klijent nije vizionar da u startu može da predvidi sve moguće funkcije i probleme
 - Klijent je živ čovek i može da promeni mišljenje
 - Poželjno je da klijent menja mišljenje
- Upravljanje prema ciljevima

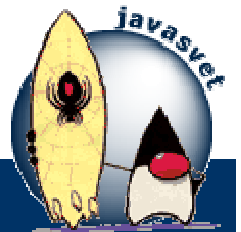


Promenljivost i neizvesnost



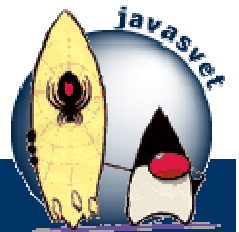
Građevina

- Materijali za izradu se godinama i vekovima ne menjaju
- Velika ograničenja
 - Zidovi su ravni i vertikalni
 - Podovi su ravni i vertikalni
 - Svaka soba mora da ima prozore i vrata i oni se nalaze na zidovima
 - Nameštaj mora da se nalazi na podu
 - Malo tipova nameštaja
- Mala promenljivost zahteva
 - Usled velikih ograničenja
 - Projekat svima jasan
 - U fazi projektovanja svi problemi mogu da se reše
- Jako skupo ispravljanje grešaka
- Velika kreativnost u projektovanju i mala u realizaciji
- Velika sličnosti između projekata



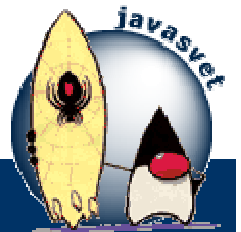
Građevina kao software

- Materijali bi se svakih par meseci menjali, kao i tehnologija gradnje
- Mala ograničenja
 - Zidovi bi mogli biti iskošeni i krivudavi
 - Podovi ne bi morali biti horizontalni
 - Sobe bi mogle imati prozore i vrata i na podu i na plafonu
 - Nameštaj bi mogao biti po zidovima ili na plafonu
- Velika promenljivost zahteva
 - Usled velikih ograničenja
 - Zbog velike slobode
 - Tek kada bi klijent počeo da koristi zgradu mogao bi da zaključi da mu zgrada odgovara
- Relativno jeftino i jednostavno ispravljenje grešaka
- Velika kreativnost i u projektovanju i u realizaciji
- Veliki stepen razlike između projekata

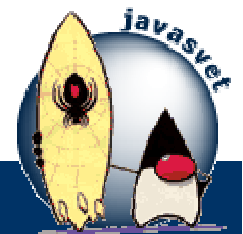
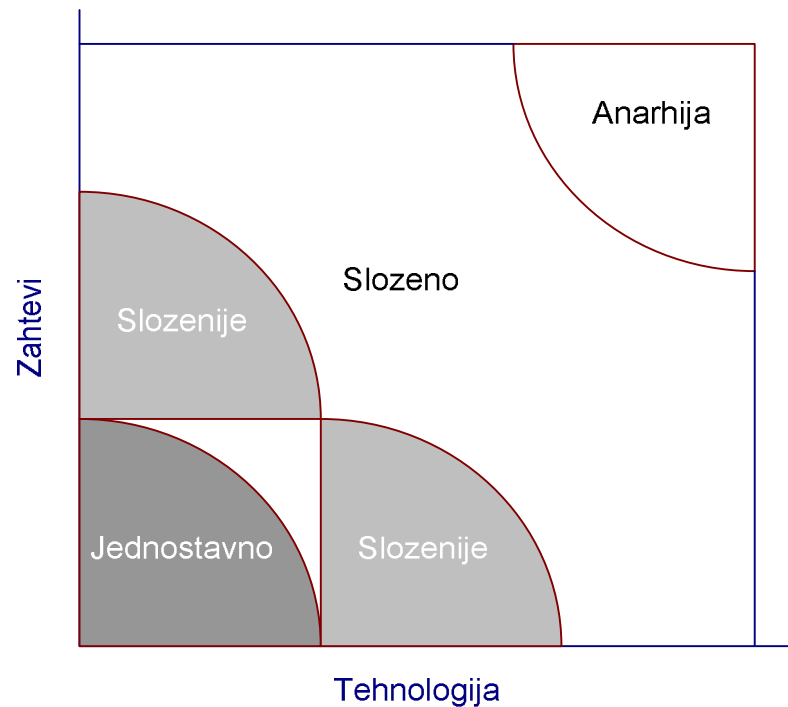


Software

- Jedina ograničenja su
 - Zahtevi i želje klijenata
 - Kreativnost programera
 - Budžet
- Svaki projekat može da se realizuje na bezbroj načina i da pri tome svaki odgovara korisniku
- Velika promenljivost tehnologije
 - Programski jezik
 - Framework
 - Koncepti dizajna kako GUI tako i aplikacije
- Korisnik u startu ne može da sagleda sve mogućnosti da finalno definiše zahtev.

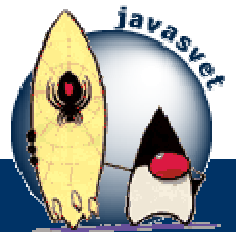


Promenljivost i neizvesnost



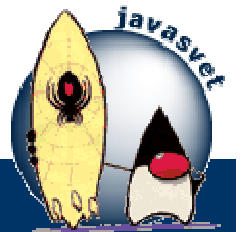
Kada primeniti agilne metodologije

- U uslovima velike dinamočnosti okruženja, metodologija razvoja kao i organizacija mora biti dinamična i fleksibilna
- Ako nema dinamičnog okruženja (npr na malim projektima) ne treba ni primenjivati agilne metodologije



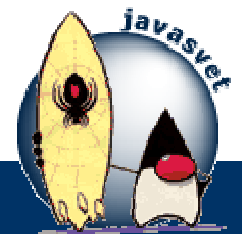
Agenda

- Zašto agilno?
- Iterativnost i evolutivnost
- Uloge
- Proces
- Planiranje i praćenje projekta
- Problemi



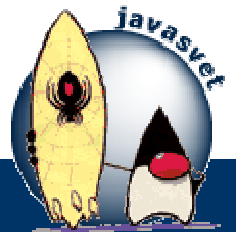
Iterativnost

- Razvoj software-a u više sekvencijalnih iteracija
- Svaka iteracija je mini projekat i sadrži sve faze od pisanja korisničkog zahteva do testiranja
- Rezultat iteracije je “nešto” što radi i što se klijentu može prezentovati
- Završena iteracije podrazumeva skup funkcija za koje se zna tačno šta rade i koje su detaljno **istestirane**.
- Testiranje obično oduzme 25-30% vremena razvoja



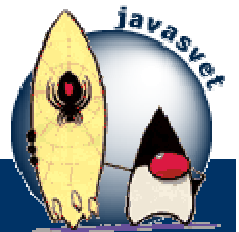
Iterativnost

- Najpre se razvijaju klijentu najbitnije funkcije
- Rana povratna informacija od klijenta
- Smanjenje rizika. Brzo se uočavaju promašaji.
- Velika zamka da iteracije prerastu u waterfall



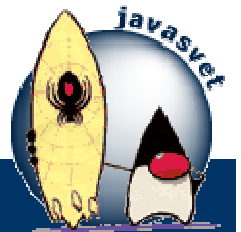
Iteracije i trajanje

- Trajanje iteracije obično od 1 do 6 nedelja, najčešće 2-3 nedelje
- Kraća iteracija - greška manje košta
- Duža iteracija - manji overhead vezan za planiranje
- Obično svaka iteracija podjednako traje
- Nema produžavanja iteracije(time boxing)
- Nema dodavanja novih zahteva u toku iteracije
- Svi timovi započinju i završavaju iteraciju istog dana



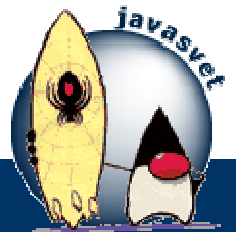
Agenda

- Zašto agilno?
- Iterativnost i evolutivnost
- Uloge
- Proces
- Planiranje i praćenje projekta
- Problemi



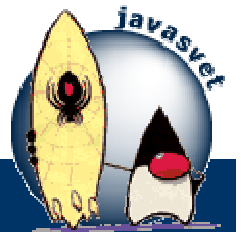
Uloge u Scrum metodologiji

- Tim
- Scrum master
- Product owner



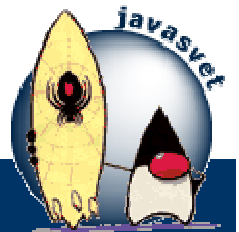
Tim

- Atomska organizaciona jedinica
- Tim snosi odgovornost, ne pojedinac
- Samoorganizujući timovi
- Tim planira i procenjuje vreme
- Jačanje timskog duha i saradnje među članovima
- Tim uključen u sve faze projekta



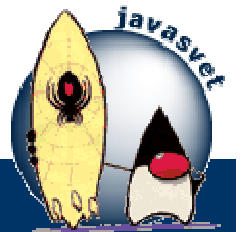
Scrum master

- “Tim lider”
- On služi timu ne tim njemu
- Pre trener i koordinator nego šef
- Zadužen da obezbedi timu sve što mu je neophodno za uspešan rad
- Svaki problem rešava u najkraćem roku – tim ne sme da čeka
- Zadužen da obezbedi poštovanje metodologije
- Povezuje tim i product owner-a
- Zadužen da obezbedi da nema promena u zahtevima tokom iteracije



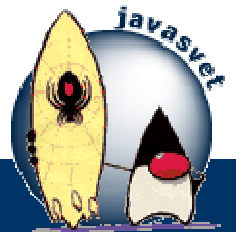
Product owner

- Predstavnik klijenta
- Neko ko odlično poznaje klijenta i njegove zahteve
- Poznaje i tehnologiju.
- Zadužen za definisanje korisničkog zahteva
- Aktivni učesnik realizacije projekta
- Stalno dostupan timu



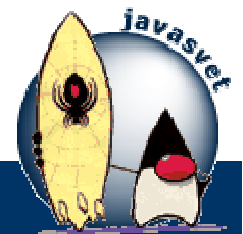
Agenda

- Zašto agilno?
- Iterativnost i evolutivnost
- Uloge
- Proces
- Planiranje i praćenje projekta
- Problemi



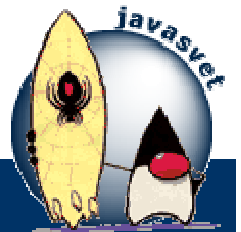
Proces razvoja

- Rečnik podataka
- Korisnički zahtev
- Igra planiranja
 - Planiranje projekta
 - Planiranje iteracija
- Dnevni “standup” sastanci
- Praćenje iteracije i projekta
- Analiza iteracije
- Promena plana



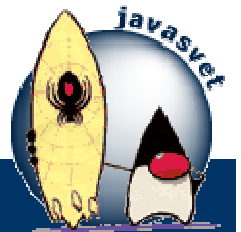
Rečnik podataka

- Veoma bitno da svi učesnici projekta iste stvari nazivaju istim imenima
- Nepostojanje rečnika dovodi do nesporazuma i nerazumevanja



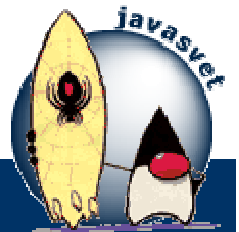
Korisnički zahtev

- Use case (RUP)
- Story card(XP)
- Specifikacija(Mozzart)



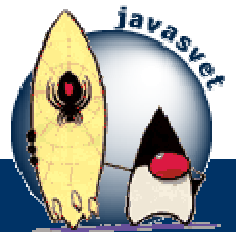
Specifikacija

- Ne pretrano detaljna
- Dovoljno informacija za procenu vremena
- Bez tehničkih detalja
- Opisuje kompletan proces ne samo software-ski deo
- Opsije određene procese a ne funkcije
- Jasna i krajnjem korisniku
- Mora da ima priču koja “teče”
- Naglašene sve nedoumice koje postoje.
- Wiki – sjajno mesto za čuvanje specifikacije
- Nikada nije završena ni savršena
- Konstantna promena i dorada kroz iteracije



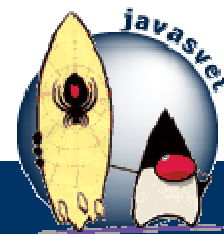
Specifikacija: naša iskustva

- Kvalitetna specifikacija = kvalitetan projekat
- Kritične procese, pogotovu nove, one koji još uvek ne postoje, najpre opisujemo kroz dijagram aktivnosti.
- Zbog bolje procene vremena potrebno je dizajnirati kritične forme ili još bolje uraditi prototip aplikacije.
- Najveće greške u proceni vremena su nam upravo bile na funkcijama koje imaju složen UI



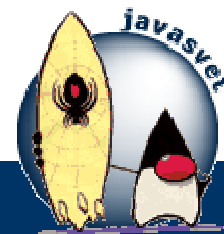
Specifikacija primer: Uplata tiketa

Igrač po ulasku u kladionicu obično kupuje listu. Na osnovu ponude takmičenja i igara iz liste odlučuje na šta će da se kladi. Popunjava listić sa tipovima na koje se kladi i predaje ga šalterskom radniku. On prekucava tipove u računar. Unosi i iznos uplate. Jako je bitno obezbediti da brzina kucanja jednog tiketa bude velika, kao bi se izbeglo stvaranje redova. Sistem odlučuje da li će da prihvati opkladu. Ako je prihvati štampa se tiket i radnik predaje tiket igraču. Tiket služi igraču kao dokaz o uplati. Prilikom predaje tiketa radnik je dužan da kaže "Izvolite Vaš tiket". Igrač može da proveri da li je radnik ispravno ukucao njegovo klađenje. Ako je bilo greške može zahtevati storniranje tiketa.



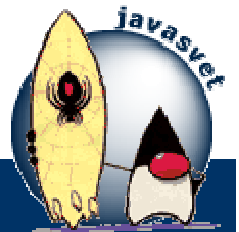
Specifikacija primer: klađenje

Igrač po ulasku u kladionicu obično kupuje listu. Na osnovu ponude takmičenja i igara iz liste odlučuje na šta će da se kladi. Popunjava listić sa tipovima na koje se kladi i predaje ga šalterskom radniku. On **prekucava tipove u računar**, unosi i iznos uplate. Jako je bitno obezbediti da brzina kucanja jednog tiketa bude velika, kao bi se izbeglo stvaranje redova. **Sistem odlučuje da li će da prihvati opkladu**. Ako je prihvati štampa se tiket i radnik predaje tiket igraču. Tiket služi igraču kao dokaz o uplati. Prilikom predajke tiketa radnik je dužan da kaže "Izvolite Vaš tiket". Igrač može da proveri da li je radnik ispravno ukucao njegovo klađenje. Ako je bilo greške može zahtevati storniranje tiketa.



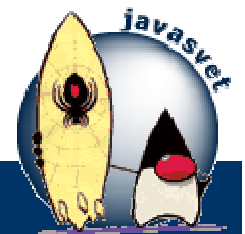
Planiranje projekta (Release planing)

- Tim:
 - Podela specifikacije na funkcije (use case-ove, priče...)
 - Procena vremena za svaku funkciju
 - Vreme implementacije jedne funkcije ne sme prelaziti trajanje iteracije
- Product owner:
 - Definisavanje prioriteta funkcija



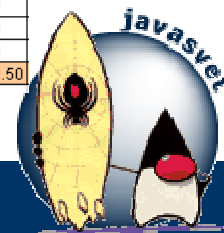
Procena vremena (projekat)

- Zbog dosta nejasnoća koristimo Delphi metod
- Svaki član tima nezavisno procenjuje vreme svake funkcije
- Optimističko (o), pesimističko(p) i srednje vreme(s)
- Procena vremena = $(o+p+4*s)/6$
- Upoređivanje individualno procenjenih vremena i diskusija
- Ako ima velikih razlika proces se ponavlja dok se procene ne usaglase
- Mi nismo imali više od 2 iteracije
- Dobijeno vreme se množi ponderom(velocity) npr 1.5 i tako dobijamo finalno vreme
- Ponder je faktor “sigurnosti”. Obuhvata:
 - Obaveze koje će programeri imati a nisu vezane za ovaj projekat
 - Greške usled nedovoljno informacija prilikom planiranja
 - Zastoji usled nepredvidjenih problema

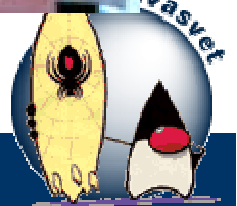


Procena vremena (projekat)

1	UseCase	Zlaja			Adnan			Perica			Prioritet	Status	% završenosti	Prosek sa ili bez pondera	procena sa ponderom 1.5			
		P	S	O	P	S	O	P	S	O								
2																		
3	Komunikacija centrale i uplatnog mesta																	
4	REM - Slanje paketa REM SERVER -> REM LOCATION SERVER	4	3	2	3.0	6	4	2	4.0	4	3	2	3.0	5	finished	100%	3.33	5.00
5	REM - Stizanje paketa na LOCATION SERVER	15	12	8	11.8	10	7	5	7.2	15	10	6	10.2	5	finished	100%	9.72	14.58
6	REM - Slanje paketa REM LOCATION SERVER -> REM CLIENT	4	3	2	3.0	7	5	3	5.0	4	3	2	3.0	4	finished	100%	3.67	5.50
7	REM - Stizanje paketa na REM CLIENT	5	4	3	4.0	7	5	3	5.0	3	2	1	2.0	4	finished	100%	3.67	5.50
8	Ukupno:				21.8				21.2				18.2				20.39	30.58
9	Task-ovi																	
10	Task - provera ispravnosti zadatka koji kopira fajlove	3	2	1	2.0	5	4	2	3.8	4	3	2	3.0	2	finished	100%	2.94	4.42
11	Task - izvršavanje zadatka koji kopira fajlove	5	3	2	3.2	5	4	3	4.0	4	3	2	3.0	5	finished	100%	3.39	5.08
12	Task - paljenje servisa	4	2	1	2.2	5	4	3	4.0	3	2	1	2.0	3	finished	100%	2.72	4.08
13	Task - gašenje servisa	4	2	1	2.2	5	4	3	4.0	2	1.5	1	1.5	3	finished	100%	2.56	3.83
14	Task - aktivacija	4	3	2	3.0	3	2	1	2.0	2	1.5	1	1.5	5	deprecated	100%	2.17	3.25
15	Task - prekidanje internet konekcije													finished	100%			
16	Ukupno:				12.5				17.8				11.0				13.78	20.67
17	Poslovi																	
18	Job - pravljenje posla	5	4	3	4.0	3	2	1	2.0	5	3	2	3.2	3	finished	100%	3.06	4.58
19	Job - pretraživanje poslova	5	4	2	3.8	3	2	1	2.0	7	5	3	5.0	3	finished	100%	3.61	5.42
20	Job - brisanje poslova	2	1	1	1.2	3	2	1	2.0	3	2	1	2.0	1	finished	100%	1.72	2.58
21	Job - editovanje poslova	2	2	1	1.8	3	2	1	2.0	3	2	1	2.0	1	finished	100%	1.94	2.92
22	Job - izbor uplatnih mesta na kojima treba izvršiti posao	5	4	3	4.0	4	3	2	3.0	4	3	2	3.0	3	finished	100%	3.33	5.00
23	Job - izvršavanje posla na uplatnom mestu	7	5	3	5.0	10	8	5	7.8	8	5	4	5.3	5	finished	100%	6.06	9.08
24	Job - patch - provera ispravnosti paketa za patch	4	3	2	3.0	3	2	1	2.0	8	5	3	5.2	2	finished	100%	3.39	5.08
25	Job - aktiviranje posla	4	3	2	3.0	3	2	1	2.0	3	2	1	2.0	5	finished	100%	2.33	3.50
26	Ukupno:				25.8				22.8				27.7				25.44	38.17
27	Lokacije i klijenti																	
28	Nova i editovanje lokacije	1.8	1.8	2	1.8	1.8	1.8	1.8	1.8	1.8	1.8	2	1.8	4	finished	100%	1.75	2.63
29	Spisak lokacija i klijenata	2.3	2.3	2	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2	2.3	4	finished	100%	2.25	3.38
30	Editovanje klijenata	2	2	2	2.0	2	2	2	2.0	2	2	2	2.0	4	finished	100%	2.00	3.00
31	Ukupno:				6.0				6.0				6.0					
32	Evidencija																	
33	Evidence - Dobijanje spiska uplatnih mesta na kojima je izvršavan neki posao	8	4	3	4.5	7	5	3	5.0	4	3	2	3.0	3	deprecated	0%	4.17	6.25
34	Evidence - Dobijanje spiska poslova koji su izvršeni na jednom uplatnom mestu	8	4	3	4.5	5	3	1	3.0	4	3	2	3.0	1	finished	100%	3.50	5.25
35	Ukupno:				9.0				8.0				6.0				7.67	11.50
36	Opšte																	
37	REM – framework	15	8	3	8.3	15	10	8	10.5	15	10	7	10.3	5	finished	100%	9.72	14.58
38	Priprema za deploy	10	10	10	10.0	10	10	10	10.0	10	10	10	10.0		in progress	50%	10.00	15.00
39	Finalno testiranje(integration test+stress test)	20	20	20	20.0	20	20	20	20.0	20	20	20	20.0		in progress	80%	20.00	30.00
40	Ukupno:				38.3				40.5				40.3				39.72	59.58
41	Ukupno za sve use-case-ove:				98.5				102.3				97.2				107.00	160.50

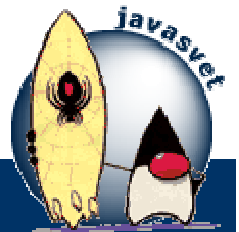


Story board



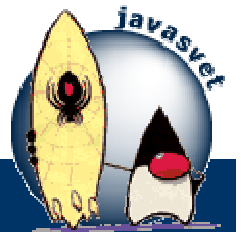
Planiranje iteracije

- Product owner rangira funkcije
- Najbitnije funkcije product owneru se prvo razvijaju
- Često to nisu one koje su programer najbitnije (najsloženije)
- Funkcije se dele na taskove
- Taskovi se detaljno opisuju
- Taskovi ne treba da traju više od 2 dana
- Tim se dogovara ko će koji task da implementira
- Programer koji implementira odredjeni task procenjuje vreme taska
- Tako procenjeno vreme množi se ponderom
- Promena plana projekta



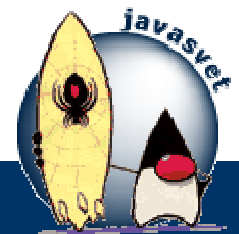
Jučerašnje vreme

- Potrošeni milioni dolara da bi se dobila preciznost procene vremena od 70%
- Samo ako bi se reklo “Sutra će vreme biti isto kao danas” dobila bi se ista preciznost.
- Ovaj princip se koristi kod procene pondera za svakog programera
- Ako je u prethodnoj iteraciji planirao da će posao završiti za 10 dana a završio ga je za 13, njegov ponder u novoj iteraciji će biti 1.3
- Prilagođavanje planiranja pojedincu
- Neko je po prirodi pesimista, neko optimista
- Poboljšanje kroz iteracije

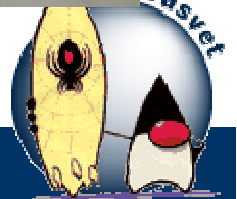
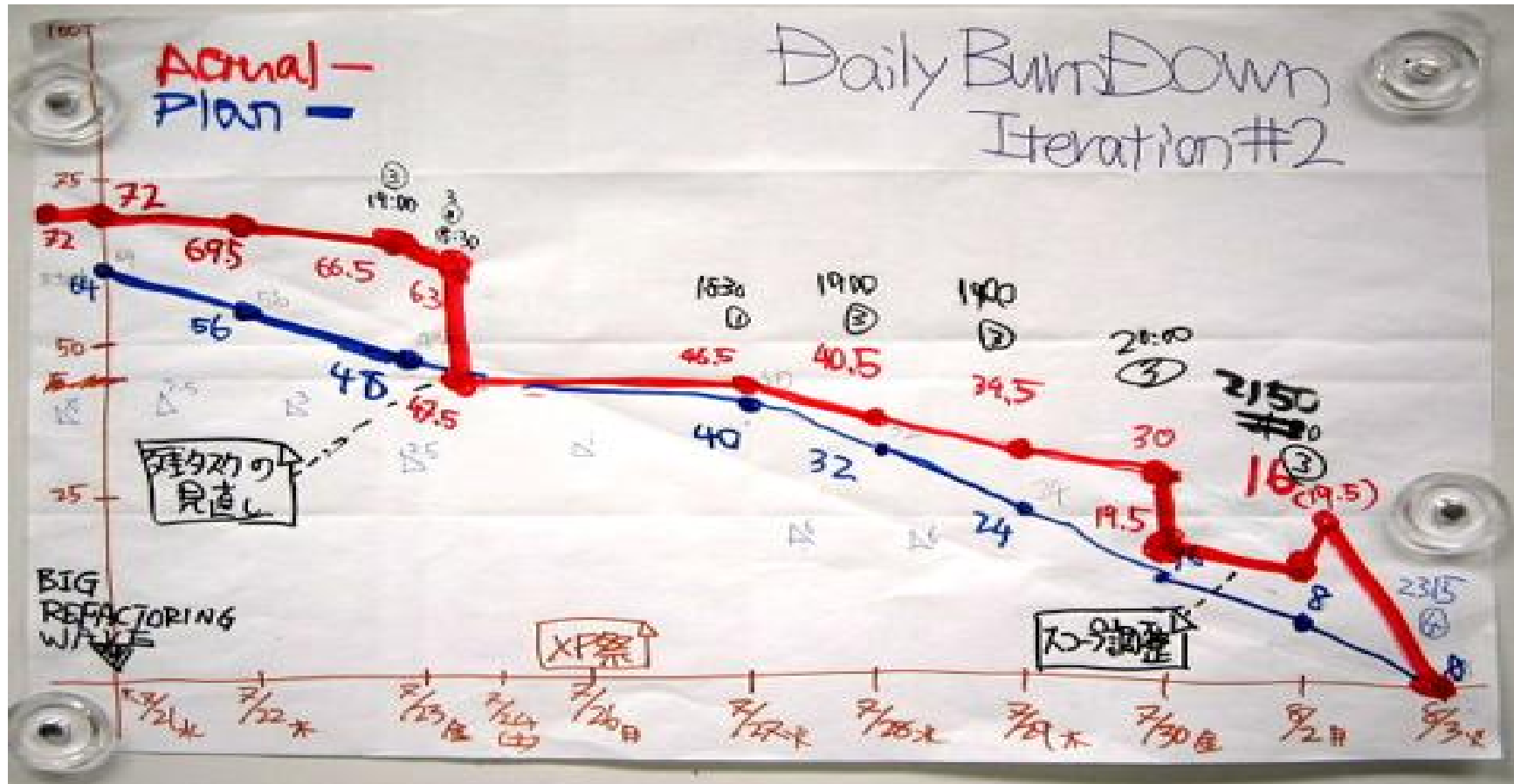


Plan iteracije

	A	B	C	D	E	F
1	Rb	Naziv use-case-a i njegova podela na sitne taskove	Ime	Procenjeno	Realizovano	Ponderisano* 1.5
2		Deployment – priprema za instalaciju				
3	1	Odredjivanje ID za klijente	Z	8	10	12
4		Ukupno vreme za taskove:		8	10	12
5		Job – Pretrazivanje poslova				
6	1	Ispravka bug-ova iz napravljenih u prethodnoj iteraciji (opisani na JIRI) – Drazen	D	10	8	15
7		Ukupno vreme za taskove:		10	8	15
8		Use-case Job - izbor uplatnih mesta na kojima treba izvršiti posao				
9	1	Ispravka bug-ova iz napravljenih u prethodnoj iteraciji (opisani na JIRI) – Adnan	A	4	24	6
10		Ukupno vreme za taskove:		4	24	6
11		Use case Task - izvršavanje zadatka koji kopira fajlove				
12	1	Ispravka bug-ova iz napravljenih u prethodnoj iteraciji (opisani na JIRI) – Adnan	A	3	3	4.5
13		Ukupno vreme za taskove:		3	3	4.5
14		Use-case Job - aktiviranje posla				
15	1	Ispravka bug-ova iz napravljenih u prethodnoj iteraciji (opisani na JIRI) – Adnan	A	3	3	4.5
16		Ukupno vreme za taskove:		3	3	4.5
17		Use-cases Slanje paketa REM LOCATION SERVER i REM				0
18	1	Test case – Ucitavanje podataka o poslu koji se skida nakon restarta	Z	6		9
19		Ukupno vreme za taskove:		6	0	9
20		Job - označavanje da paket treba poslati na izabrana uplatna mesta				
21	1	Kupljenje sa GUI-ja oznacenih klijenata	Z	2	1	3
22	2	Logika koja proverava da li za sve izabrane klijente moze da se oznaci slanje paketa	Z	2	4	3
23	3	Metoda servisa za označavanje slanja paketa klijentima	Z	2	1	3
24	4	Dao metoda za označavanje slanja paketa klijentima	Z	2	1	3
25		Ukupno vreme za taskove:		8	7	12
26		Job - aktiviranje posla				
27	1	Kupljenje sa GUI-ja oznacenih klijenata	Z	2	1	3
28	2	Logika koja proverava da li za sve izabrane klijente moze da se oznaci aktiviranje posla	Z	2	2	3
29	3	Metoda servisa za označavanje aktiviranja posla na klijentima	Z	2	1	3
30	4	Dao metoda za označavanje aktiviranja posla na klijentima	Z	2	1	3
31		Ukupno vreme za taskove:		8	5	12
32		Nova i editovanje lokacije				
33	1	Osmisljavanje i izrada forme	A	4	1	6
34	2	Pravljenje logike koja uzima podatke za lokaciju (novu ili editovanu) sa forme	A	2	1	3
35	3	Metode servisa za kreiranje i editovanje lokacije	A	2	1	3
36	4	Dao metode za kreiranje i editovanje lokacije	A	2	4	3
37	5	Testiranje	A	4	3	6
38		Ukupno vreme za taskove:		14	10	21

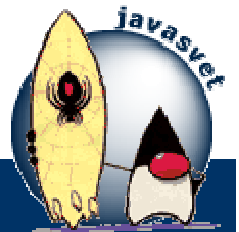


Burn down chart



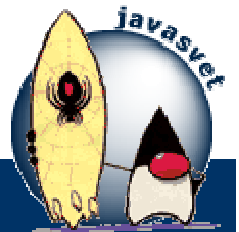
Iteracija

- Svako jutro “stand up” sastanak u 10:15
- Svakodnevno ažuriranje realizovanog vremena
- Ukoliko se proceni da se plan ne može ispuniti product owner određuje koji task se neće raditi
- Nema prekovremenog rada da bi se ispunio plan



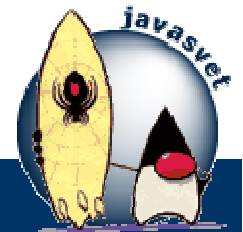
“Stand up” sastanak

- Prisustvuju tim i scrum master
- Mogu i ostali zainteresovani ali samo kao posmatrači
- Scrum master postavlja 3 pitanja svakom članu tima:
 - Šta si radio juče?
 - Šta planiraš danas da radiš?
 - Ima li kakvih problema?
- Zabranjeno je diskutovati o bilo čemu drugom
- Ako se pojavi potreba za diskusiju o nečemu drugom, organizuje se poseban sastanak.
- Traje maksimalno 15-20 minuta
- Daje ritam iteraciji



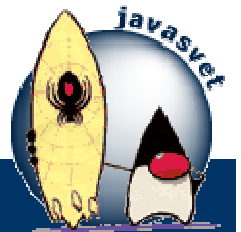
Završetak iteracije

- Dogovoreni taskovi su kompletno završeni i istestirani
- Nema “polugotovih” taskova.
- Prezentacija Product owneru i zainteresovanim klijentima
- Planiranje naredne iteracije
- Promena plana projekta
- Svi timovi završavaju iteraciju istog dana. Daje ritam projektu.
- Ova faza obično traje pola dana po nedelji iteracije. Ako je iteracija trajala 3 nedelje, za ovu fazu će biti potrebno 1.5 dan



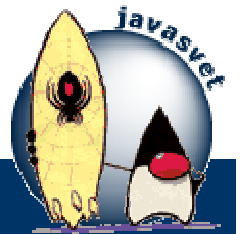
Razvoj

- Konceptualni model, sekvencijalni dijagrami, dijagrami stanja, klasni dijagrami?
- Test first development? Automatsko testiranje? Acceptance testovi?
- Programiranje u parovima?



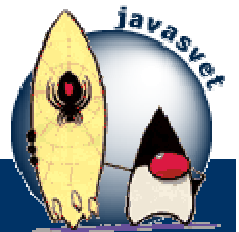
Problemi agilnih metodologija

- Zahtevaju visoko motivisane i kvalitetne kadrove
- Problem kod velikih projekata sa više timova



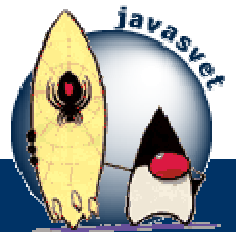
Problemi u primeni

- Agilno \neq nema dokumentacije
- Dosta neodređenosti, pa se zahteva iskustvo onih koji je primenjuju
- Ubediti menadžment u kvalitet agilnih metodologija
- Nedostatak ljudi u Srbiji koji poznaju i imaju iskustva u agilnim metodologijama



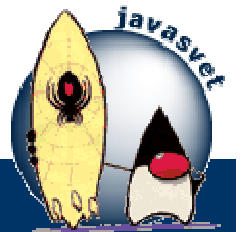
Reference - sajtovi

- www.agilemanifesto.org.
- <http://www.agilealliance.org/>
- http://en.wikipedia.org/wiki/Delphi_method
- [http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))
- <http://www.joelonsoftware.com/>
- <http://www.parleys.com>



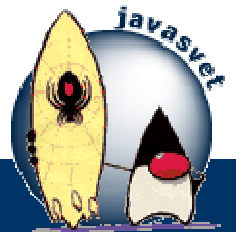
Reference - Knjige

- Agile and Iterative Development (Craig Larman)
- Agile Project Management with Scrum (Ken Schwaber)
- Extreme Programming Explained (Kent Beck)
- Planning Extreme Programming (Kent Beck, Marting Fowler)



Zaključak

PROMENA JE POŽELJNA



Hvala!

Kontakt:

Predrag Spasojevic

predrag.spasojevic@mozzart.co.yu

