

Metodologije razvoja softvera

Scrum - Sprint, Zahtevi, Korisničke priče

dr Milan Stojkov

Katedra za informatiku

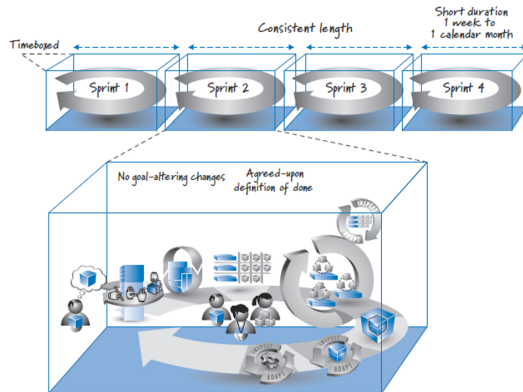
2022.



Fakultet tehničkih nauka
Univerzitet u Novom Sadu

Sprint

- Predstavlja osnovu Scrum okruženja



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Sprint - timebox

- Sprintovi moraju biti vremenski ograničeni, imaju tačan datum početka i kraja
 - Vremenski okvir se naziva **timebox**
- U okviru timebox-a od razvojnog tima se očekuje da održivim tempom završe dogovoreni skup zadataka koji je u skladu sa ciljem sprinta

Sprint - timebox

- Razlozi zbog kojih je bitno vremensko ograničavanje
 - Ograničavanje količine rada koja je u toku (WIP - *work in process*)
 - *Timeboxing* je tehnika za ograničavanje količine rada koja je u toku
 - WIP predstavlja inventar posla koji je započet, ali nije završen
 - Sam tim planira da radi na onim zadacima koje očekuje da će završiti
 - Loše upravljanje i organizacija sprinta može ozbiljno da utiče na projekat
 - Prioritizacija
 - *Timeboxing* prisiljava na prioritizaciju zadataka – ovo podstiče rad na bitnim stvarima
 - Progres
 - Pomaže u demonstriranju progressa time što se završavaju i validiraju bitni delovi do unapred poznatog datuma (kraj sprinta)
 - Smanjuje organizacione rizike pomerajući fokus sa nepouzdatih izveštaja o učinku
 - Takođe pomaže razvojnom timu i klijentima da imaju bolji uvid u to šta je još preostalo do kraja projekta

Sprint - timebox

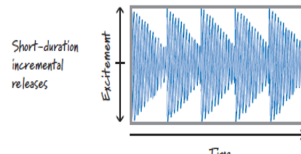
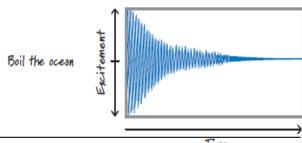
- Razlozi zbog kojih je bitno vremensko ograničavanje
 - Izbegavanje nepotrebnog perfekcionizma
 - Prisiljava razvojni tim da ne mora baš svaku sitnicu uraditi do savršenstva
 - Motivisanje zatvaranja
 - Pokazalo se da se poslovi u većoj meri završavaju kada postoji jasan datum do kada se mora završiti
 - Poboljšavanje prediktivnosti
 - Iako nije moguće precizno predvideti količinu posla koju tim može uraditi u narednoj godini, moguće je predvideti količinu posla u narednom sprintu koja može da se uradi

Sprint - kratko trajanje

- Razlozi zbog kojih je bitno kratko trajanje
 - Lakše planiranje
 - Lakše je planirati nekoliko nedelja unapred nego li nekoliko meseci
 - Sam postupak planiranja zahteva manje truda i mnogo je precizniji kod kraćeg trajanja
 - Brži *feedback*
 - Tokom svakog sprinta kreira se radna verzija softvera koju klijent može da proba
 - U ranoj fazi razvoja moguće je uočiti neke nepoželjne solucije u razvoju softvera i zaobići ih

Sprint - kratko trajanje

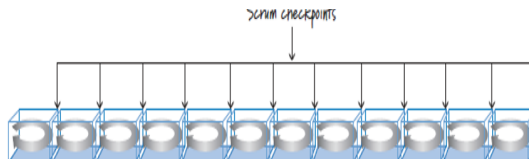
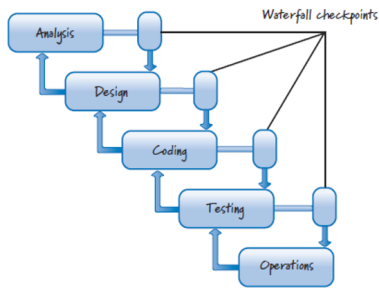
- Razlozi zbog kojih je bitno kratko trajanje
 - Poboljšavaju povratak investicije
 - Omogućuju češće i brže isporuke funkcionalnog softvera – čime klijent ima priliku za ranije ostvarivanje prihoda od tog softvera
 - Kontrolisana/ograničena greška
 - Čak i ako sve što se uradi u sprintu bude pogrešno ne mora bitno da utiče na ishod čitavog projekta – *koliko lošeg može da se uradi u nedelji – dve?*
 - Entuzijazam
 - U prirodi čoveka je da mu entuzijazam opada sa vremenom – na dužim projektima



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Sprint - kratko trajanje

- Razlozi zbog kojih je bitno kratko trajanje
 - Češće kontrolne tačke
 - Mnogo više kontrolnih tačaka nego kod sekvencijalnog rada na projektu
 - U složenom okruženju lakše je funkcionisati ako ima više kontrolnih tačaka u kojima je moguće izvršiti prilagođavanje



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Sprint - konzistentno trajanje

- Trajanje svih sprintova bi trebalo da bude identično osim u specifičnim slučajevima
- Specijalni slučajevi
 - Pomeranje sa npr. četvornedeljnih na dvonedeljne sprintove radi češćeg *feedback*-a
 - Praznici, fiskalna godina, odmori
 - Očekivan period za novu verziju (*release*) – npr. nova verzija izlazi jednom nedeljno, a sprint je na dve nedelje
- Razlog za produženje sprinta ne može biti to što tim ne može da obavi čitav posao dogovoren za sprint
- Lakše se usklađuje ritam i olakšava planiranje

Sprint - nema promene cilja

- Jedno od važnijih pravila za sprint je da kada on započne nema promene cilja koji je definisan
- Šta je cilj sprinta?
 - Svaki sprint bi trebalo da se opiše ciljem sprinta koji predstavlja poslovni smisao i vrednost tog sprinta
 - Tokom faze planiranja tim treba da se dogovori oko toga šta će biti cilj i da se na osnovu toga odaberu iz *product backlog*-a stvari koje će biti urađene u sprintu
- Obostrana posvećenost
 - Razvojni tim se posvećuje da će ispuniti cilj do kraja sprinta
 - *Product Owner* se posvećuje da neće menjati cilj tokom sprinta
 - Balansira potrebu da poslovanje bude adaptivno na izmene, a opet dozvoljava timu da radi na nepromenljivom zahtevu

Sprint - nema promene cilja

- Pragmatičnost
 - Pravilo da nema promene cilja nije i zakon, tj. Scrum predviđa mogućnost da zbog novonastalih okolnosti ipak dođe do promene cilja
 - Npr. konkurencija lansira novi proizvod i u analizi se utvrdi da je potrebno promeniti cilj sprinta kako bi naš proizvod i dalje bio konkurentan
 - Npr. u produkcionom sistemu se desila greška koju tim mora da ispravi

Sprint - nema promene cilja

- Abnormalan prestanak
 - Ako se desi situacija da je cilj sprinta postao potpuno besmislen, tim može odlučiti da sprint više nema smisla i predložiti *Product Owner*-u abnormalan prekid sprinta
 - Tim se sa *Product Owner*-om dogovara oko novog sprinta
 - U praksi ovo je dosta redak slučaj

Sprint - definicija završenosti

- Definicija završenosti – *definition of done*
- Kada se podrazumeva da je nešto završeno
- Obično se definiše lista zadatka (stavki) koje tim treba da uradi da bi se smatralo da je sprint završen
- Ova lista može da zavisi od:
 - Vrste projekta
 - Članova tima
 - Korišćenih tehnologija
 - Potencijalnih prepreka koje mogu da utiču na projekat
- Lista može da evoluiru tokom projekta
 - Dozvoljeno je menjati listu od sprinta do sprinta
 - Obično se ove izmene vrše u nekoliko početnih sprintova

Sprint - definicija završenosti

- Definicija završenosti odnosi se na deo proizvoda koji se razvija u okviru sprinta
- Kako se deo proizvoda u stvari sastoji od stavki iz *product backlog*-a, svaka od ovih stavki mora biti završena u skladu sa definicijom završenosti

Zahtevi

- U Scrum-u zahtevi su nešto o čemu se pregovara, nisu fiksirani
- Na zahteve se gleda kao nešto što ima visok stepen slobode i što može da se menja da bi se što bolje postigao željeni poslovni cilj. Npr.
 - Ako ponestaje novca, neki malo manje bitni zahtevi se mogu odbaciti
 - Ako se tokom razvoja uvide neke nove činjenice koje utiču na smanjenje prioriteta zahteva, taj zahtev se može odbaciti
 - Ako se pojavi novi zahtev sa visokim prioritetom onda treba taj zahtev ubaciti u listu i posvetiti mu se

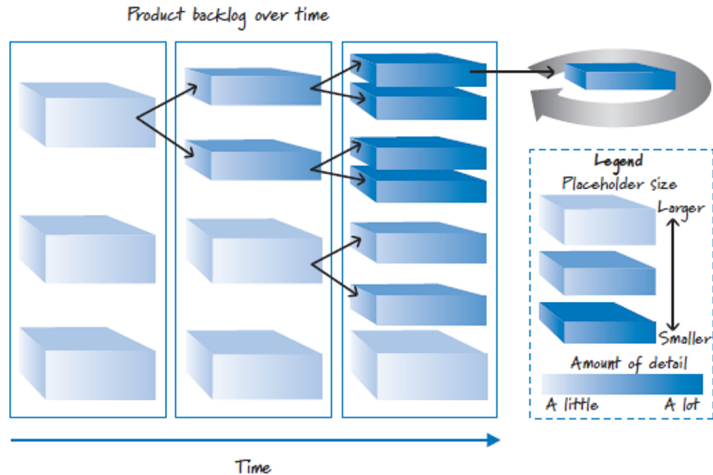
Zahtevi

- Pokazalo se da kada se kreira neki novi proizvod koji ima visok stope inovacija učestalost promena zahteva je jako visoka
 - Nije moguće na samom početku kreirati sve zahteve koliko god se trudili
- Scrum ne zahteva da se svi zahtevi detaljno razrade na samom početku projekta
 - Ovim se štedi na vremenu i novcu
 - Očekuje se da će tokom vremena biti izmena, odnosno da će se neki zahtevi pojednostaviti tokom razvoja

Zahtevi

- Umesto da se u *Product backlog*-u (PB) kreira gomila detaljnih stavki (*items*) na samom početku kreira se manji skup stavki koji će vremenom evoluirati
- Svaka stavka PB-a vezana je za neke poslovne ciljeve (*business value*) koji se žele implementirati

Zahtevi



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Zahtevi

- Inicijalno stavke PB-a su velike
 - Predstavljaju širok opseg poslovnih ciljeva sa jako malo detalja
- Tokom vremena ove stavke će tokom konverzacije između učesnika biti redefinisane u kolekciju manjih i detaljnijih stavki
- Na kraju će PB stavke biti dovoljno male i dovoljno detaljne da mogu da se realizuju u okviru sprinta
 - Čak i tokom sprinta će se neki detalji razraditi u komunikaciji razvojnog tima i *Product Owner-a*

Zahtevi

- Scrum ne definiše standardni format za predstavljanje stavki PB-a
- U praksi mnogi timovi koriste „korisničke priče“ (*user stories*) za reprezentaciju
- Osim korisničkih priča koriste se i slučajevi korišćenja ili neke druge prilagođene tehnike

Zahtevi - Konverzacija

- Jedan od bitnih problema kod sekvencijalnog razvoja nastao je zbog prevelikog oslanjanja isključivo na pisane zahteve
- U Scrum-u konverzacija se koristi kao ključna poluga za razumevanje i pojašnjenje navedenih zahteva
- Konverzacija i dalje ne zamenjuje u potpunosti dokumentaciju
 - Oni koji žele ili moraju da kreiraju specifikaciju zahteva i dalje to mogu uraditi na osnovu stavki PB i detaljnog opisa vezanog za njih

Zahtevi – Progresivno usavršavanje

- Kod sekvencijalnog razvoja svi zahtevi moraju biti na istom nivou detalja u isto vreme, tj. ne bi smelo da se naknadno dodaju neki detalji za zahtev
- Ovo izaziva nekoliko problema:
 - Moraju se predvideti svi detalji u ranoj fazi razvoj kada se, po pravilu, najmanje zna o proizvodu
 - Svi zahtevi se tretiraju jednako bez obzira na prioritet – što znači da će se možda potrošiti vreme na nešto što se možda neće implementirati
 - Kreira se velika baza zahteva koja će se verovatno vrlo često tokom vremena menjati što je dosta skup proces
 - Smanjuje se verovatnoća da se, koristeći konverzaciju, pojašne zahtevi pošto su oni već definisani

Korisničke priče

- Format za predstavljanje poslovnih ciljeva u *Product backlog*-u
- Nastaju na način koji ih čini razumljivim i poslovnim i tehničkim ljudima
- Strukturno su obično jednostavne
- Mogu biti napisane na različitim nivoima granularnosti i mogu progresivno da se modifikuju
- Jedan način za opis/definisanje *user stories* – 3C:
 - *Cards* (kartice)
 - *Conversation* (konverzacija)
 - *Confirmation* (potvrda)

Korisničke priče

- Ideja **kartica** je da korisnici na relativno malom parčetu papira napišu jedan *user story*
- Najčešći šablon za pisanje je da se navede **uloga korisnika** koji želi da postigne određeni **cilj** i šta je **benefit** postizanja tog cilja
- Namena kartice nije da se opišu svi detalji korisničke priče već da iskaže suštinu
- Predstavlja osnovu za neku dalju diskusiju

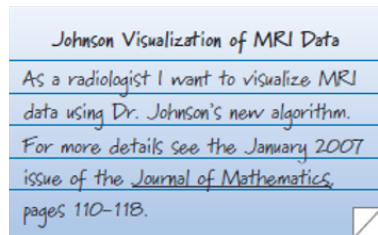
User Story Title
As a <user role> I want to <goal> so that <benefit>.

Find Reviews Near Address
As a typical user I want to see unbiased reviews of a restaurant near an address so that I can decide where to go for dinner.

Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Korisničke priče

- Detalji zahteva se razotkrivaju i razrađuju u **konverzaciji** između razvojnog tima, *Product Owner*-a i klijenata
- Konverzacija po pravilu nije jednokratni događaj već nešto što će da traje
 - Jedna konverzacija tokom definisanja *user story*-a
 - Druga konverzacija tokom razrade *user story*-a
 - Treća tokom procene
 - Četvrta tokom planiranja sprinta
 - Peta tokom implementiranja i testiranja
 - ...
- Iako je konverzacija verbalna može se dopuniti odgovarajućom dokumentacijom



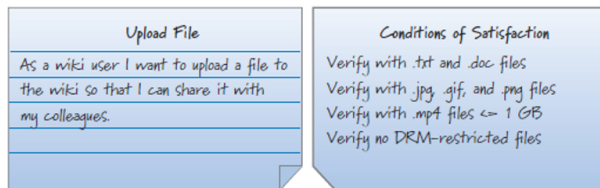
Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Korisničke priče

- *User story* sadrži potvrdu u formi uslova za ispunjenost (zadovoljenje)
- Ovi uslovi predstavljaju kriterijume prihvatljivosti koji pojašnjavaju željeno ponašanje
- Koriste se od strane razvojnog tima radi boljeg razumevanja šta treba da se realizuje i testira i od strane *Product Owner*-a kako bi potvrdilo da je taj *user story* implementiran u skladu sa njegovim zahtevima

Korisničke priče

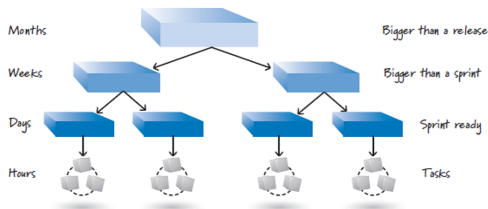
- Uslovi se obično navode na poledini kartice
- Mogu se izraziti kao testovi prihvatljivosti (*acceptance tests*) visokog nivoa
 - Nisu jedini testovi za proveru funkcionalnosti iz *user story*-a
 - Sa strane *Product Owner*-a ovi testovi su važni jer predstavljaju dokaz da je nešto implementirano kako treba
 - Zgodni za definisanje inicijalnih *user story*-ja i za njihovo redefinisanje
 - Pristup baziran na testovima se zove još i **specification by example** ili **acceptance-test-driven development (ATTD)**



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Korisničke priče

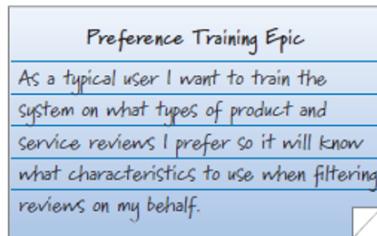
- Korišćenje korisničkih priča iste veličine tokom čitavog procesa razvoja nekog softvera nije pogodno
 - Npr. kratke korisničke priče su pogodne za sprintove, ali ne i za planiranja na visokom nivou
- Po pravilu *user stories* se pišu na različitim nivoima apstrakcije
 - Obično što je nivo apstrakcije viši to je potrebno više vremena za implementaciju - često više meseci i često opisuju jedan čitav *release*



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Korisničke priče

- *User stories* na najvišem nivou apstrakcije se zovu još i epovi (**epics**)
 - Opisuju globalnu sliku ono što se želi postići
 - *Epic* nikad ne bi trebao da bude polazna osnova za sprint, već ga treba posmatrati kao nešto što predstavlja osnovu za kolekciju mnogo manjih i detaljnijih korisničkih priča



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Korisničke priče

- *User stories* u trajanju od nekoliko nedelja zovu se još i **features**
- Najkraća forma se zove još i **stories/sprintable stories/implementable stories**
- Pojedini timovi koriste i pojam **teme (theme)** kojim se predstavlja kolekcija povezanih *user stories*
 - Zgodan način da se vidi šta pripada istoj funkcionalnoj oblasti
- **Zadaci (tasks)** su nivo ispod *user stories*
 - Obično predstavljaju nešto na čemu će raditi jedan ili dva člana tima
 - Njihova realizacija se meri u satima
 - Nisu isto što i *user stories*, tako da ne bi trebalo da se definišu kada i *user stories*
 - *Task* obično opisuje kako nešto napraviti, a ne šta napraviti

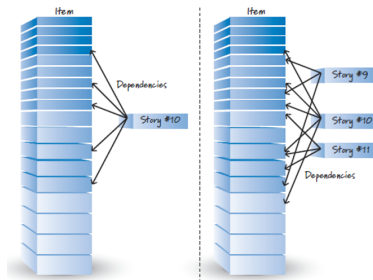
Korisničke priče

- Kako napisati dobre korisničke priče?
- Jedna preporuka je INVEST kriterijum koji se prilikom evaluacije koliko korisničke priče odgovaraju svojoj nameni:
 - **I**ndependent (nezavisnost)
 - **N**egotiable (mogućnost pregovaranja/dogovaranja)
 - **V**aluable (korisnost)
 - **E**stimatable (procenljivost)
 - **S**mall (odgovarajuće dimenzionisan)
 - **T**estable (mogućnost testiranja)

Korisničke priče

● Independent (nezavisnost)

- Koliko god je moguće korisničke priče bi trebalo da su međusobno nezavisne ili bar što labavije povezane
- *User stories* koji imaju visok stepen zavisnosti komplikuju procenu, definisanje prioriteta i planiranje



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

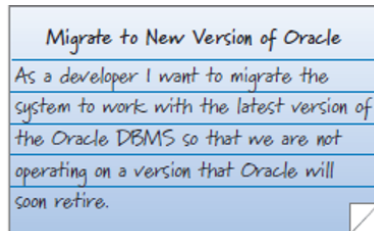
Korisničke priče

- **Negotiable (mogućnost dogovaranja/pregovaranja)**
 - *User stories* nisu pisani ugovor u formi unapred definisanih korisničkih zahteva već predstavljaju osnovu za nešto o čemu će se dalje pregovarati i diskutovati
 - Dobre *user stories* jasno definišu koja poslovna funkcionalnost se želi realizovati i zašto, sa druge strane ostavljaju dovoljno prostora svim učesnicima da se dogovore oko detalja
 - Najčešći slučaj gde se dogovaranje narušava je kada *Product Owner* navede kako treba neka *user story* da se implementira

Korisničke priče

● Valuable (korisnost)

- Svaki *user story* bi trebalo da ima neku vrednost ili klijentu ili krajnjim korisnicima softvera ili oboma
- Ako od nekog *user story*-a korist nema ni klijent ni krajnji korisnik onda on ne bi trebalo da se nađe u *product backlog*-u
- Postavlja se pitanje šta sa *user stories* koje imaju neku vrednost razvojnom timu, ali ne i klijentu i korisnicima – **tehničke korisničke priče**
 - Ključni problem kod ovakvih *user stories* je što treba ubediti *Product Owner*-a u njihov svrsishodnost, tj. objasniti mu zašto treba da finansira i to
 - U praksi tehničke *user stories* se obično ne smeštaju u *product backlog*, već se obično predstavljaju kao *task*-ovi



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Korisničke priče

● Estimatable (procenljivost)

- Svaki *user story* bi trebalo da bude procenljiv od strane razvojnog tima koji će izvršiti dizajn, implementaciju i testiranje
- Procena obezbeđuje indikator o veličini *user story*-a, a time i indikatore o vremenu i trošku potrebnom za realizaciju
- Veličina *user story*-a je jako bitna informacija za Scrum tim
- Ako tim nije u stanju da proceni veličinu *user story*-a onda je ona ili jako velika, ili dvosmislena (neprecizna) ili tim nema dovoljno znanja da bi izvršio adekvatnu procenu
 - Ako je suviše velika tim treba zajedno sa *Product Owner*-om da je dekomponuje na više manjih
 - Ako je neprecizna treba da se rasprave elementi koju je čine nepreciznom/dvosmislenom
 - Ako tim nema dovoljno znanja onda je potrebna dodatna aktivnost po pitanju pojašnjenja tog *user story*-a

Korisničke priče

- **Small (odgovarajuće dimenzionisan)**
 - Veličina *user stories* zavisi od nivoa apstrakcije
 - *User stories* namenjeni za sprintove treba da budu dimenzionisani u skladu sa dogovorenim trajanjem sprintova
 - Poželjno je da *user story* traje kraće od samog sprinta
 - Uvek je bolje imati u jednom sprintu više *user stories* koji traju kraće, nego li imati jedan *user story* koji traje jednako kao i sam sprint pošto je tada rizik nezavršetka *user story* veći

Korisničke priče

- **Testable (mogućnost testiranja)**

- *User stories* bi trebalo da mogu da se testiraju u Bulovom smislu: testovi su ili prošli ili nisu
- *User story* koji može da se testira ima dobar kriterijum prihvatanja (*acceptance criteria*)
- Bez mogućnosti testiranja ne može se znati da li je određeni *user story* realizovan ili ne na kraju sprinta
- Za neke *user stories* ne postoji praktičan način za testiranje
 - Npr. Od sistema se zahteva 99.999% uptime u produkciji

Korisničke priče

- Nefunkcionalni zahtevi predstavljaju ograničenja na nivu čitavog sistema
- Mogu, a ne moraju da se predstavljaju kao *user stories*
- Utiču na dizajn i testiranje većine *user stories* u *product backlog*-u
- Svaki funkcionalni zahtev je nešto što se obično uključuje u definiciju završenosti

Internationalization	Web Browser Support
As a user I want an interface in English, a Romance language, and a complex language so that there is high statistical likelihood that it will work in all 70 required languages.	System must support IEB, IE9, Firefox 6, Firefox 7, Safari 5, and Chrome 15.

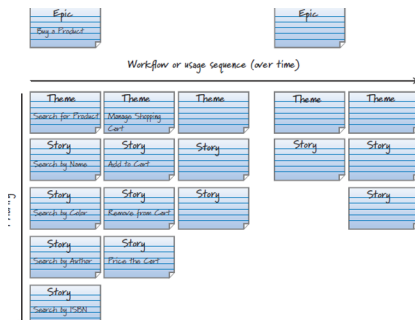
Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Korisničke priče

- Kako se formiraju *user stories*?
- U tradicionalnom pristupu zahtevi se formiraju tako što se korisnici pitaju šta žele – u praksi se ovo pokazalo kao loš pristup
- Drugi pristup je uključiti korisnike u tim koji odlučuje šta treba da se napravi i koji konstantno razmatra šta se pravi
 - Obično se sprovode **user-story-writing workshop**-ovi
 - Na ovim *workshop*-ovima učestvuje i *Product Owner*, *Scrum Master*, razvojni tim, klijenti i neka druga strana (ako treba)
 - Traju od nekoliko sati do nekoliko dana
 - Od *workshop*-a se ne očekuje da se odjednom generiše kompletna lista *user stories*, već obično ima specifični fokus
 - Npr. Na *workshop*-u se razmatra šta treba da ide u sledeći *release*
 - Na prvom *workshop*-u se obično i određuju uloge u organizaciji sa ciljem da se definišu uloge koje se u delu uloge u *user stories* (“As a <user role>, I want to...”)
 - Tokom samog *workshop*-a nema standardnog načina za generisanje *user stories*
 - Koristi se i *top-down* i *bottom-up* pristup

Korisničke priče

- **Story mapping** je tehnika koja se bazira na korisnički orijentisanoj perspektivi za kreiranje *user stories*
- Osnovna ideja je da se korisničke aktivnosti visokog nivoa dekomponuju u *workflow* koji se potom dekomponuje u skup detaljnijih zadataka



Slika preuzeta iz: *Essential Scrum: A practical guide to the most popular Agile process*, Kenneth S. Rubin, Addison-Wesley, 2012.

Korisničke priče

- Osnovna razlika između **user-story-writing workshop**-a i **story mapping**-a je što je tokom *workshop*-a fokus primarno na generisanju *user stories*, a ne na definisanju njihovog prioriteta
- Tehnika *story mapping*-a se može iskoristiti kao komplement *workshop*-u da pomogne u definisanju prioriteta i vizuelizaciji prioriteta *user stories*