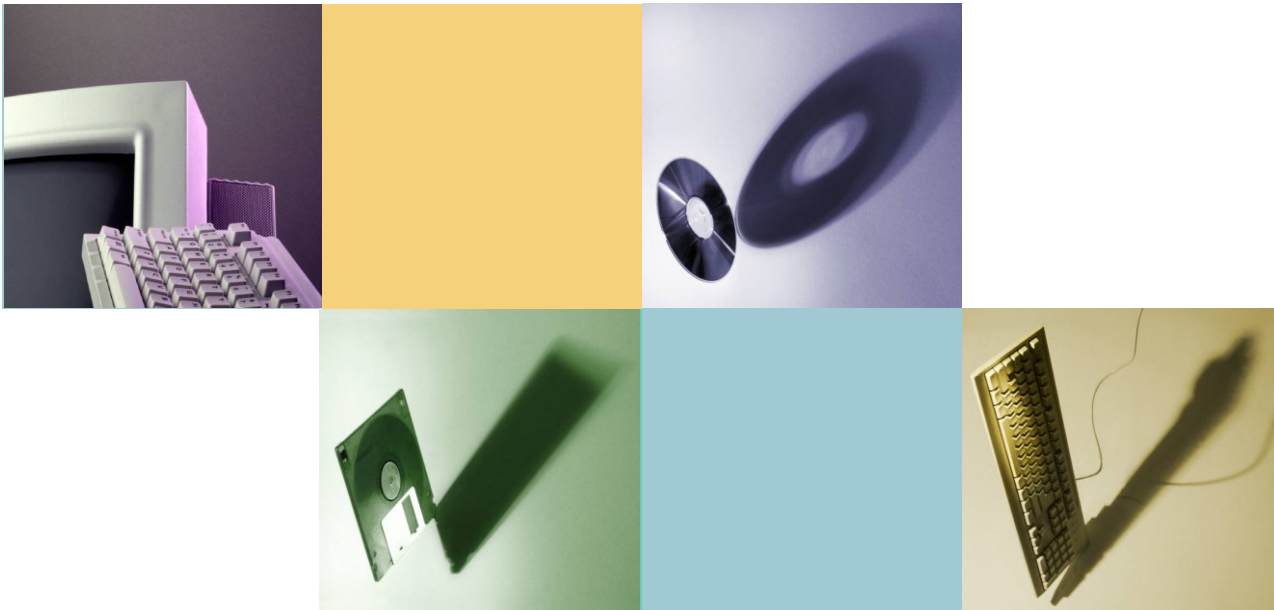


Upravljanje softverskim projektima



Agilne metodologije

Predavač: Dražen Drašković

Agenda

1. XP i Lean

2. Kanban

3. Scrum

4. Pitanja?



Agilne metodologije

- 17 istraživača okupljenih kao „The Agile Alliance“ 2001. godine potpisuju **Agilni manifesto**.
- Agilni mentalitet ima 4 vrednosti koje ga razlikuju od tradicionalnih procesa softverskog razvoja:

1

Individuals and Interactions

over

Processes and Tools

2

Working Software

over

Comprehensive Documentation

3

Customer Collaboration

over

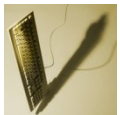
Contract Negotiation

4

Responding to Change

over

Following a Plan



Agilni manifesto (1)

- 1) Naš najveći prioritet je da zadovoljimo kupca ranom ili kontinuiranom isporukom vrednog softvera.
- 2) Zahtevi koji se menjaju dobrodošli su čak i kasno u razvoju softvera.
- 3) Isporučujte softver relativno često, iz manjih delova, u periodima od nekoliko nedelja do nekoliko meseci.
- 4) Programeri, kao i svi učesnici projekta, treba da rade svakodnevno na projektu zajedno, tokom čitavog trajanja.
- 5) Izgraditi projekat oko motivisanih pojedinaca. Dajte im okruženje i podršku koja im je potrebna, i verujte im.
- 6) Najefikasnija i najefektnija razmena informacija ka i u okviru razvojnog tima je „licem u lice“



Agilni manifesto (2)

- 7) Ispravan softver, u produkciji, glavna je mera napredovanja.
- 8) Agilni procesi promovišu održivi razvoj.
- 9) Stalna pažnja usmerena na tehničkoj izvrsnosti i dobrom dizajnu softvera povećava agilnost.
- 10) Jednostavnost, kao veština maksimizovanja količine nezavršenog posla, je od suštinske važnosti.
- 11) Najbolje softverske arhitekture, dizajn i zahtevi dolaze od timova koji su samoorganizovani.
- 12) U redovnim terminima, tim treba da razmišlja o tome kako da bude efikasniji, i da podešava i prilagođava svoje ponašanje u skladu sa tom analizom.



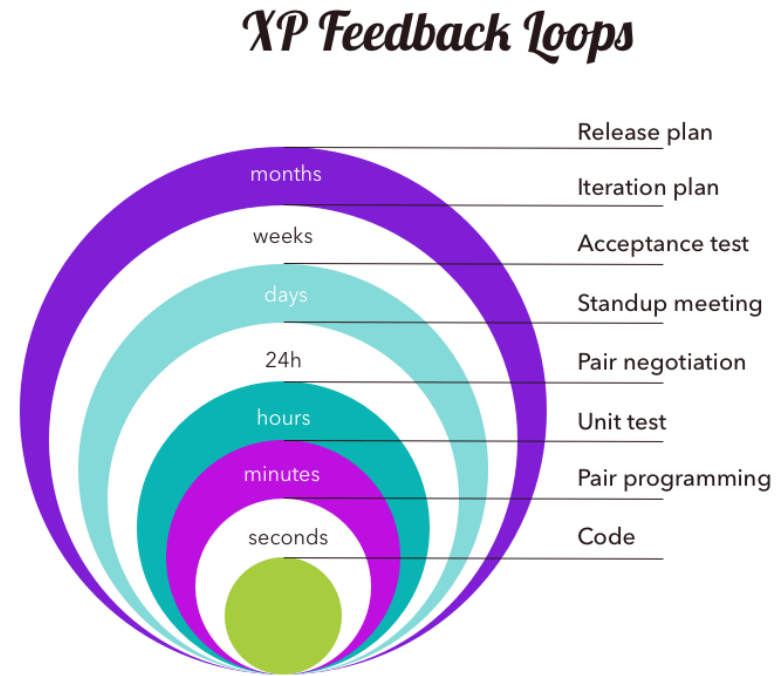
Kent Beck

- Softverski inženjer, rođen 1961. godine, studirao Računarske nauke na Univerzitetu u Oregonu (US)
- Tvorac Ekstremnog programiranja (XP)
- Jedan od 17 potpisnika Agilnog manifesta
- Bio je pionir u dizajnu softvera (*design patterns*) i tvorac pristupa Test Driven Development (TDD), od 2002.
- Razvio je xUnit seriju radnih okvira za testove, iz kojih je proizašao i JUnit (2004)
- Trenutno je zaposlen u kompaniji *Facebook*



Ekstremno programiranje (XP)

- Niz nekoliko važnih elemenata koji vode do uspešnog procesa i projekta
- Izvući maksimum iz potencijala programera
- Kruta pravila unutar tima krenula da se izostavljaju (npr. dnevni testovi integracije)
- Kritike menadžera: stalno prisutan klijent = stres
- Bez detaljnih zahteva i sa dizajniranjem softvera „u letu“ možemo imati veoma neefikasan projekat.



Source: <http://www.extremeprogramming.org>

objectstyle



XP - principi 1 i 2

- Igra planiranja
 - Naručilac definiše svaki zahtev prema vrednosti
 - Korisnici pišu scenarije, kako sistem treba da radi, a projektni tim procenjuje neophodne resurse za realizaciju
 - Korisnici dodeljuju prioritete zahtevima
 - Mora da se postigne konsenzus (šta će da se uradi)
 - Stvaraju se mape verzija (šta koja verzija sadrži i rok isporuke)
- Male verzije
 - Softverski sistem se projektuje da se isporuči što je pre moguće
 - Funkcije se dekomponuju na manje podfunkcionalnosti i karakteristike, kako bi se neke funkcionalnosti isporučile u ranoj fazi
 - Male verzije zahtevaju fazni pristup razvoja, odnosno inkrementalne ili iterativne cikluse



XP - principi 3 i 4

- Metafora (razumevanje od strane svih)
 - Usaglašavanje projektnog tima (programera) sa kupcima i svim drugim učesnicima u razvoju projekta oko zajedničke vizije – kako tretirati ključna pitanja i definisati zajedničku nomenklaturu
- Jednostavan dizajn
 - Sistem zasnovan samo na trenutnim potrebama
 - Ne razmatrati predviđanja i buduće potrebe, jer takav pristup dovodi do nepotrebnih funkcionalnosti



XP - princip 5 i 6

- Razvoj vođen testovima (TDD)
 - Prvo pišemo scenarije testova (za jedinične testove), kao način da naručilac specificira sve korisničke zahteve
 - Po izradi softvera, odmah se vrši testiranje i verifikacija
 - Funkcionalne testove definiše naručilac, a izvršavaju ih članovi projektog tima i krajnji korisnici. Ovi testovi su automatizovani i izvršavaju se svakog dana (idealno).
 - Testovi u kodu se pišu i izvršavaju od strane projektog tima.
- Refaktorisanje / Refaktorizacija (preuređivanje) koda
 - Novi zahtevi vode nas ka preispitivanju postojećeg projektog rešenja
 - Refaktorisanje se odnosi na preispitivanje zahteva i dizajna, i njihovo preformulisanje u skladu sa novim potrebama
 - Često se radi prestrukturisanje koda, bez ikakvih izmena u spoljašnjem ponašanju sistema



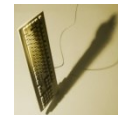
XP - principi 7, 8 i 9

- Programiranje u paru (eng. *Pair Programming*)
 - Dva programera razvijaju system koristeći specifikaciju i definisan dizajn
 - Jedna osoba odgovorna za kodiranje, druga za kontrolisanje
- Kolektivna svojina (eng. *Collective Code Ownership*)
 - U fazi razvoja, svaki učesnik razvojnog tima može da izmeni bilo koji deo sistema, naravno uz obaveštenje drugima šta je uradio
 - Najčešće se softverski sistemi razvijaju na razvojnim serverima, pa se tek na kraju stavljaju na produkcionu server
- Neprekidna integracija (eng. *Continuous Integration*)
 - Brza isporuka funkcionalnosti – isporuka sistema koji funkcioniše može da se obeća naručiocu u roku od odmah (jedan dan ili par sati)
 - Među revizije: mali inkrementi i poboljšanja, nema velikih skokova između revizija



XP - principi 10, 11 i 12

- Održiv tempo (bez prekovremenog posla)
 - Umor uvek proizvede greške!
 - Sugestija je da je cilj da svi članovi projektnog tima rade 40-časova nedeljno
 - Ukoliko tim treba da utroši mnogo više vremena od previđenog, onda znači da su rokovi nerealni ili da postoji veliki problem u nedostatku resursa neophodnih za izvršavanje takvog projekta
- Naručilac raspoloživ na terenu
 - Naručilac radi na definisanju zahteva i obezbeđivanju povratnih informacija od krajnjih korisnika
- Definisani standardi kodiranja (eng. *Coding Standard*)
 - Izgled programskog koda treba da bude kao da ga je pisala jedna osoba
 - Konzistentnost po pitanju imenovanja, strukture koda i paketa,...



Lean razvoj softvera

- 1980: Mary Poppendieck, IT menadžer, u velikoj fabrici video kasete
- Problem: njihovi konkurenti iz Japana sa bržom i jeftinijom izradom kasete
- Šta radi? Čita knjigu koju je napisao menadžer iz fabrike automobila Toyota, primenjuje njihova procesna unapređenja koja su koristili i Meri kao rezultat dobija: produktivnost u pakovanju video kasete raste sa 60% planiranih na 95% ostvarenih.
- Sa suprugom Tomom, Meri izdaje knjigu (2003): **Lean Software Development**
- Veoma sličan kao Kanban pristup



Radni tok u Lean-u

1. Mapiranje procesa

Kreira se vizuelna mapa sa čitavim radnim procesom. Ona mora biti detaljna i transparentna.



2. Set-based razvoj

Koristi se pristup više različitih opcija i odabirati onu opciju koja je najbolja. Neuspeh brzo proceniti. Testirati što je pre moguće (TDD).



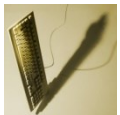
3. MVP isporuka

Odabrati najboljeg kandidata i isporučiti ga kao MVP (*Minimum viable product*)



4. Ispirati i ponavljati

Unapređivati proizvod tokom vremena i nastaviti izgradnju MVP.



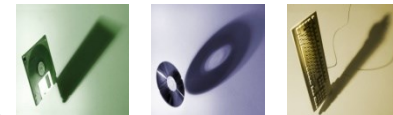
Lean – principi i prakse

PRINCIPI

- Eliminirati gubitke
- Pojačati učenje (stalno učiti)
- Odluke donositi što je kasnije moguće
- Isporuke raditi što je pre moguće
- Osnažiti tim
- Ugraditi integritet
- Gledati celokupno, i optimizovati

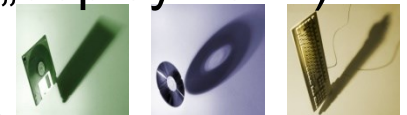
PRAKSE

- Uočiti gubitke
- Mapirati tokove vrednosti (*Value-stream mapping*)
- *Set-based* razvoj
- Sistemi zasnovani na prevlačenju (*Pull systems*)
- Teorija čekanja
- Motivacija
- Merenja (dobre metrike)
- Test Driven Dev. (Toyota)
- Trunk-based Dev.



Lean

- Minimalni održivi proizvod (*Minimal Viable Product*) – verzija softvera koju puštate čim možete pre, često čak i pre pisanja bilo kakve dokumentacije
- Sličnosti sa Kanban-om:
 - smanjiti gubitke i maksimizovati vrednost za kupca
 - ne gubiti vreme na nešto što se nikada neće koristiti, ili započeti iznova
 - koncept prebacivanja zadatka unapred, i verovanje da se vaši radnici trude maksimalno
- Razlike u odnosu na Kanban:
 - Lean ima neke inženjerske prakse (TDD)
 - Lean manje propisuje vremenske okvire isporuke (deadlines), ali je tim potencijalno spreman u svakom trenutku da se postavi rok i isporuči softverski proizvod (uradi „deployment“)



Istraživanja u Lean projektima

- Gubici se postižu zbog:
 - Izrade pogrešnih karakteristika unutar proizvoda ili celog proizvoda
 - Nepravilnog upravljanja
 - Ponovnog rada na nekim delovima (re-work)
 - Nepotrebno kompleksnih softverskih rešenja
 - Čekanja na zadatke ili više poslova istovremeno
 - Neefikasne komunikacije
 - Psihološkog uznemirenja (stresa)



Kanban

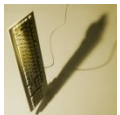
- Na japanskom: *signboard / billboard*
- Cilj vizuelizovati tok rada (upravljanje radom)
- Radni zadaci treba da se prikažu vizuelno kako bi učesnici imali uvid u napredak i proces, od početka do kraja, obično putem table „Kanban“
- Rad se izvršava koliko to kapaciteti dozvoljavaju, a ne da se gura u proces kada se to traži
- Nastao iz *Lean proizvodnje* (Toyota, 1988)
- *Kanban* se vrlo često koristi u proizvodnji softvera, u kombinaciji sa drugim metodama i radnim okvirima, kao što je npr. *Scrum*



Kanban - osnove

- Dve osnovne prakse u Kanban-u:
 - Vizuelizovati vaš rad (*Visualize work*) korišćenjem table
 - Ograničiti max rad na projektu (*limit Work In Progress*) tj. uspostavite vaš „idealni kapacitet“ i održavajte ga
- Četiri dodatna prakse u Kanban-u:
 - Upravljajte radnim tokom projekta (napretkom) – pratiti, meriti i izveštavati o radnom toku, kroz svako stanje
 - Učiniti pravila/smernice eksplicitnim - tačno opišite postupak da biste mogli da ga poboljšate
 - Primeniti povratne veze (tzv. „*feedback loop*“)
 - Unaprediti saradnju, razvijati se eksperimentalno – koristiti modele i naučnu empirijsku metodu za sprovođenje stalnih, inkrementalnih i evolutivnih promena

- *Knjiga: Kanban for software development*



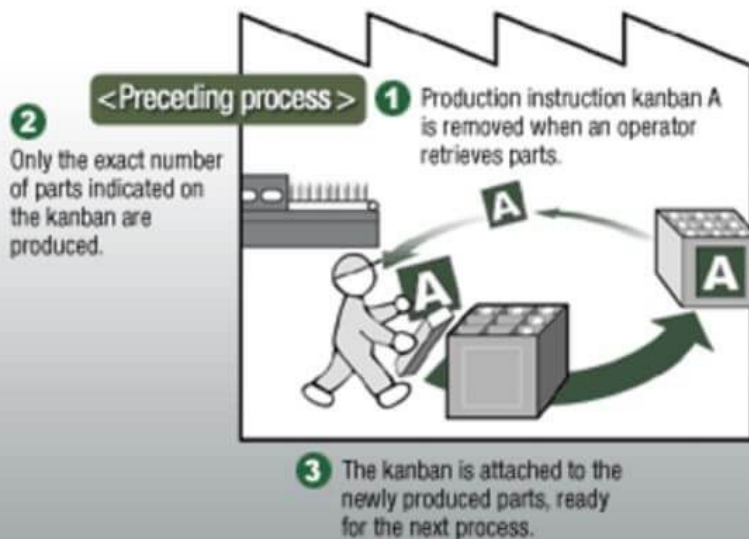
Dijagram Kanban principa

Eliminisanje gubitaka: „Just in time“ princip - „Napraviti samo ono što je potrebno, kada je potrebno i u količini koja je potrebna“

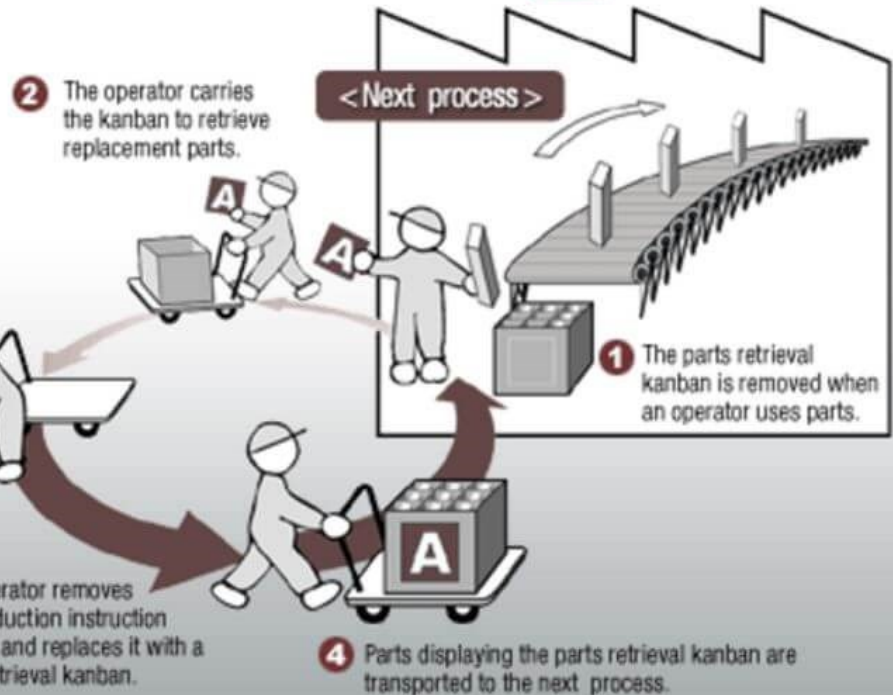
Conceptual Diagram of the Kanban System

Image from Toyota's Official Site: http://www.toyota.co.jp/jpn/company/vision/production_system/just.html

Operational Flow of Production Instruction Kanban **A**



Operational Flow of Parts Retrieval Kanban **A**



Kanban tabla u softverskoj industriji

Kanban table u soft. industriji mogu da se razlikuju u sadržaju (ovde su date karakteristike - *features*, i korisničke priče). Kolone su aktivnosti radnog toka, vrste su kao „plivačke trake“ da bismo grupisali priče po karakteristikama.

Pool of Ideas	Feature Preparation		Feature Selected	User Story Identified	User Story Preparation		User Story Development		Feature Acceptance		Deployment	Delivered
Epic 431	3 - 10 In Progress Ready		2 - 5	30	15 In Progress Ready		15 In Progress Ready (Done)		8 In Progress Ready		5	Epic 294
Epic 478	Epic 444	Epic 662	Epic 602			Story 602-02 Story 602-03	Story 602-06 Story 602-04	Story 602-05 Story 602-01	Epic 401	Epic 609	Epic 694	Epic 386
Epic 562	Epic 589		Epic 302	Story 302-03 Story 302-01	Story 302-07	Story 302-09	Story 303-05	Story 302-04	Epic 468	Epic 577	Epic 276	Epic 419
Epic 439	Epic 651		Epic 335	Story 302-02 Story 302-06	Story 302-08				Epic 362		Epic 339	Epic 388
Epic 329			Epic 335	Story 335-09 Story 335-10 Story 335-04	Story 335-05	Story 335-06					Epic 521	Epic 287
Epic 287			Epic 512	Story 335-08 Story 335-01 Story 335-03	Story 335-02	Story 335-07					Epic 582	Epic 274
Epic 606	Discarded			Story 512-04 Story 512-07 Story 512-02	Story 512-01							
	Epic 511	Epic 213		Story 512-05 Story 512-06 Story 512-03								
	Epic 221											

Policy

Business case showing value, cost of delay, size estimate and design outline.

Policy

Selection at Replenishment meeting chaired by Product Director.

Policy

Small, well-understood, testable, agreed with PD & Team

Policy

As per "Definition of Done" (see...)

Policy

Risk assessed per Continuous Deployment policy (see...)

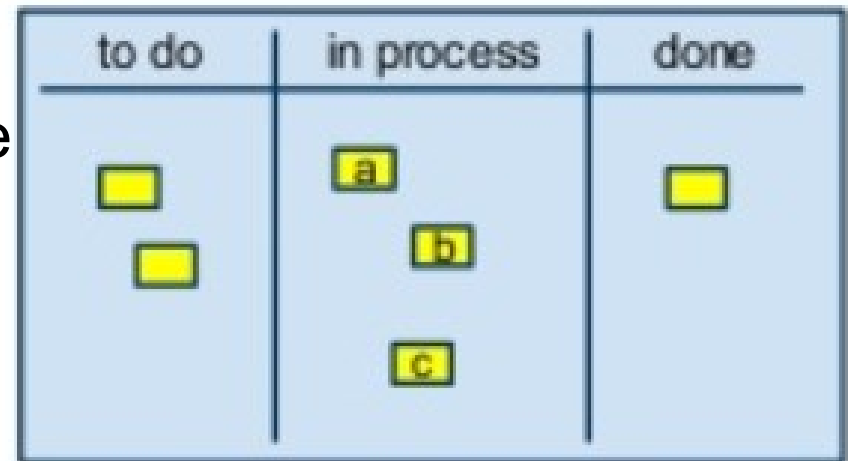
Pravila kada crtamo tablu

- Kanban tabla mora da:
 - vizuelizuje rad kroz karakteristike i korisničke priče
 - obuhvati ograničenja (pogledati prethodni slajd, zaokružene vrednosti – broj radnih stavki u tom koraku radnog toka)
 - dokumentuje smernice / gotova pravila (pogledati prethodni slajd, plavi pravougaonici na dnu)
 - upravljanje Kanban-ovim protokom kroz korake (pogledati: *User Story ident*, *User Story dev*, *Features accept*, imaju potkorake – IN PROGRESS i READY)



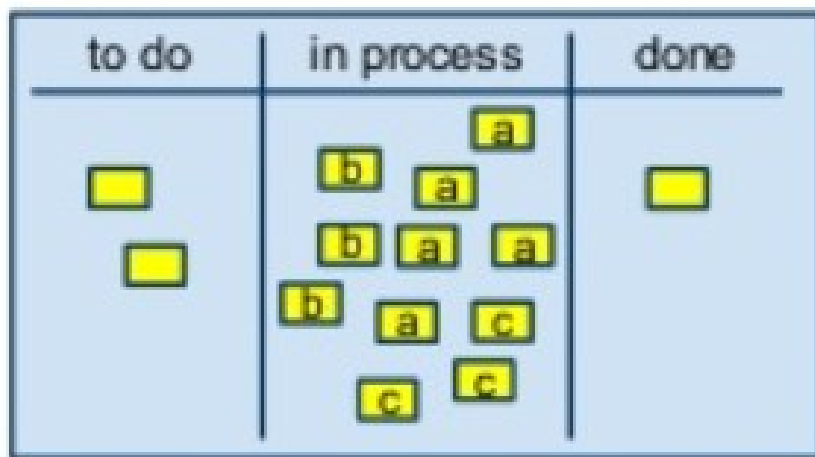
Kanban - jednostavan primer

- Jednostavna table najčešće ima samo tri kolone:
 - zadaci koje je potrebno uraditi (TO DO)
 - zadaci koji se trenutno rade (IN PROCESS/PROGRESS)
 - zadaci koji su već urađeni (DONE / FINISHED)
- Svaka kartica predstavlja stavku posla (zadatak koji radimo), svaka stavka ima svoje ime, a onda se vezuje za određenog programera/e (na slici: a, b, c)
- Cilj je da se na lak način vizuelizuje posao i kreira prostor za interakciju
- Neke table koriste redove da prikažu osobe koje rade



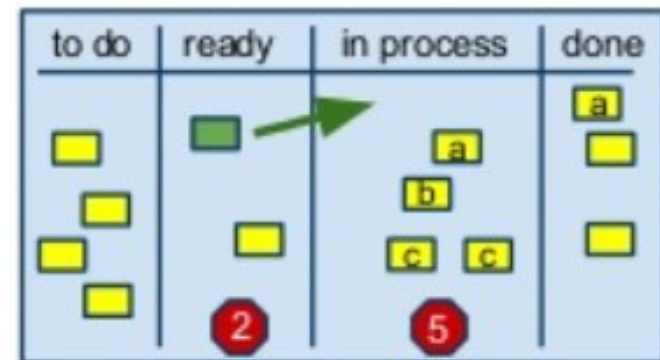
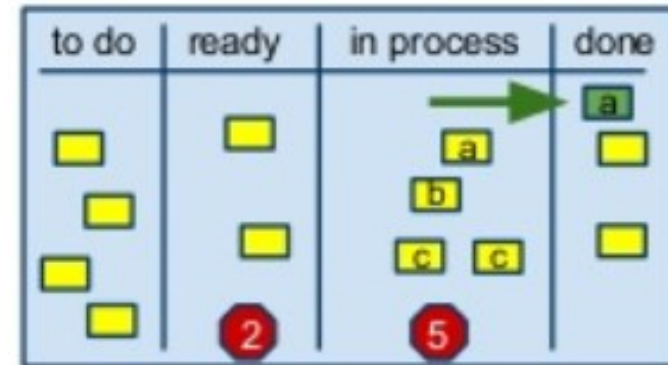
Kanban - mogući problemi

- Previše posla koji se trenutno izvršava (WIP) i previše dodeljenih kartica nekom programeru
- Šta uraditi?
 - Kontrolisati broj kartica na tabli, identifikovati jasne pragove koji su vezani za određene faze
 - Postavljanje ograničenja, npr. primena „multitasking“ u radu kod određenih programera, pa neko može recimo da ima za svoje ime vezane 2 kartice (ne moraju svi programeri, već samo neki)

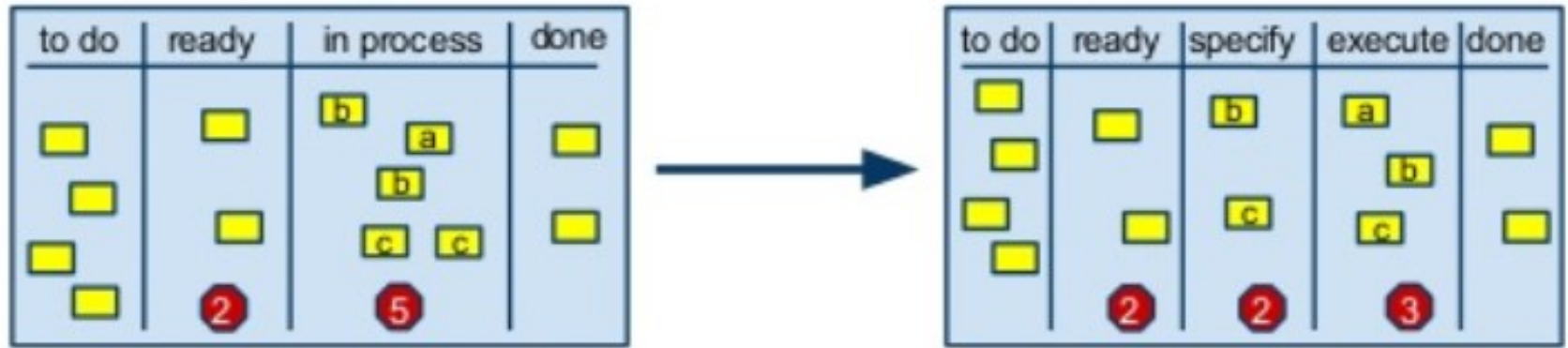


Kanban - tok

- Korak #1:
Član tima A završava zadatak i prebacuje se kartica iz statusa „IN PROGRESS“ u „DONE“
- Korak #2:
Član tima A povlači jednu karticu iz statusa „READY“ u zadatke koje trenutno izvršava i počinje rad na tom zadatku - „IN PROGRESS“
- Korak #3:
Po prioritetu, sledeća kartica se iz statusa „TO DO“ prebacuje i dovlači u kolonu „READY“



Kanban - idealni tok



- Kada se uspostavi kapacitet našeg tima, znamo koliki su nam limiti, i uvek težimo idealnom toku, jer prevlačimo zadatke (kartice) iz jedne u drugu kolonu.



SCRUM

- Radni okvir za razvoj i održavanje velikih softverskih proizvoda
- Scrum ideja: empirijska kontrola procesa, iz iskustva => znanje, iz poznatog => odlučivanje
- Koristi iterativno-inkrementalni pristup, sa ciljem povećanja izvesnosti i upravljanja rizicima
- Scrum čine:
korisničke uloge + artefakti i pravila + događaji



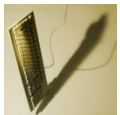
Scrum - korisničke uloge (role)

- 3 korisničke uloge:
 - **Razvojni tim** – od 5 do 9 osoba, zašto?
 - Da li tim da bude više funkcionalan ili specijalizovan?
 - Specijalnosti: programer (*developer*), tester, analitičar,...
 - **Scrum Master** – Kakva je to osoba?
 - Moderator timskog rada
 - Obezbeđuje sve što je potrebno timu
 - Priprema sastanke i nadgleda proces
 - **Vlasnik proizvoda** (*Product Owner*) – Ko treba da piše priče?
 - To je osoba koja reprezentuje naručioca posla (klijente)
 - Odgovorna je za sadržaj i za listu prioriteta u proizvodnom backlog-u



Scrum - Artefakti

- Glavni *Scrum* artefakti:
- 1) Korisnička priča (*User story*)
- 2) Zadatak (*Task*)
- 3) Proizvodni backlog
 - Preostali poslovi proizvoda
- 4) Sprint backlog
 - Preostali poslovi u okviru Sprint-a
- 5) Inkrement proizvoda (*Product increment*)
- 6) Dodaci (*Extensions*)
 - npr. Burndown - grafikon preostalog posla
- A šta je i gde je tu Sprint?



Scrum - Sprint

- **Sprint** - vremenski ograničen period (mesec dana ili kraći) tokom kog se stvara upotrebljiv, potencijalno isporučiv inkrement proizvoda
- Svaki *Sprint* obično ima konstantno trajanje tokom procesa razvoja (termin: *time-boxed*)
- Novi *Sprint* počinje odmah po zaključivanju prethodnog



Scrum - događaji u Sprint-u (1)

- Sprint se sastoji iz 4 događaja:
 - (1) **Planiranje** - Šta treba raditi u narednom periodu?
 - Da li imamo neke ranije iste zadatke i slične procene?
 - Planira se narednih 15-30 dana
 - Biraju se oni poslovi koji će da se rade (isporučiti sa inkrementom koji je rezultat tog Sprint-a)
 - Priprema se Sprint Backlog koji prikazuje vreme koje će biti potrebno da se urade poslovi
 - Na ovim sastancima učestvuje ceo tim, a max trajanje sastanka je obično ograničeno na 1 dan (8 radnih sati)
 - (2) **Dnevni Scrum (Daily)** – 15-minutni sastanak;
 - Da li ceo tim treba da prisustvuje?
 - Šta ste uradili juče? Kakav je plan za danas?
 - Ima li problema?

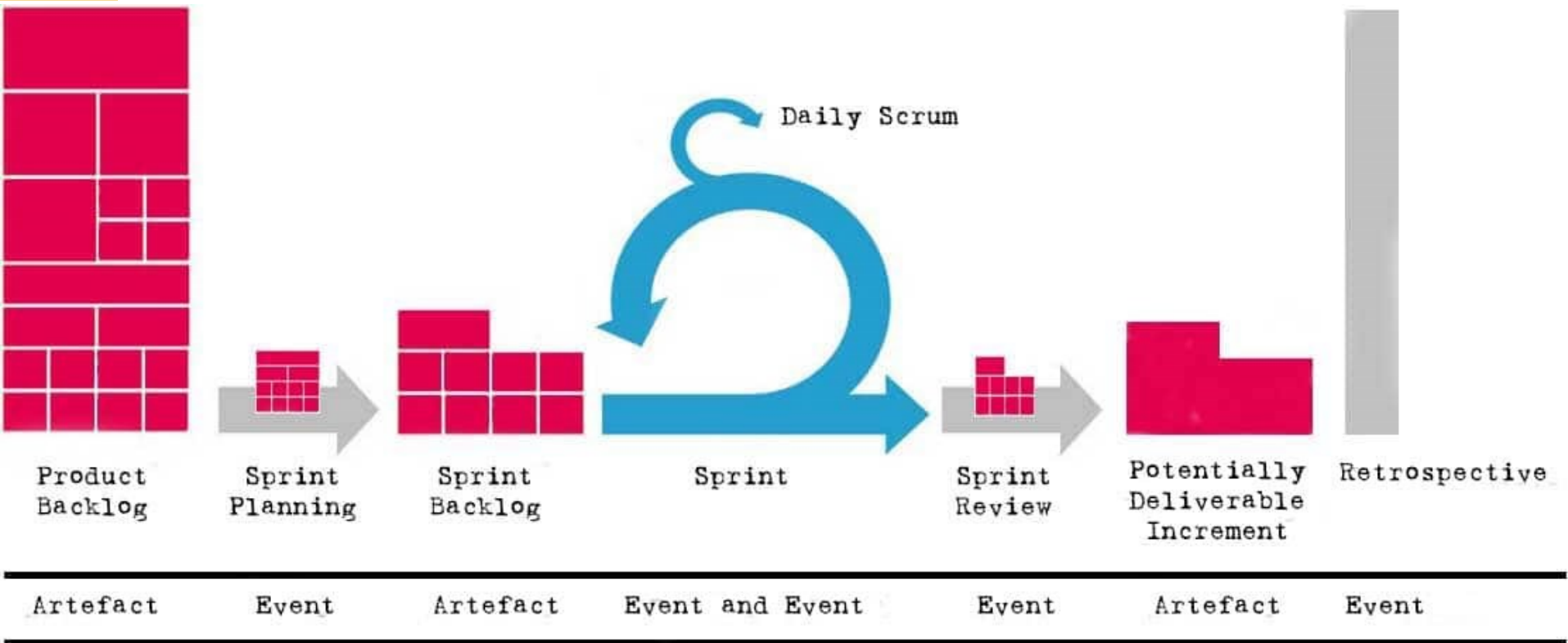


Scrum - događaji u Sprint-u (2)

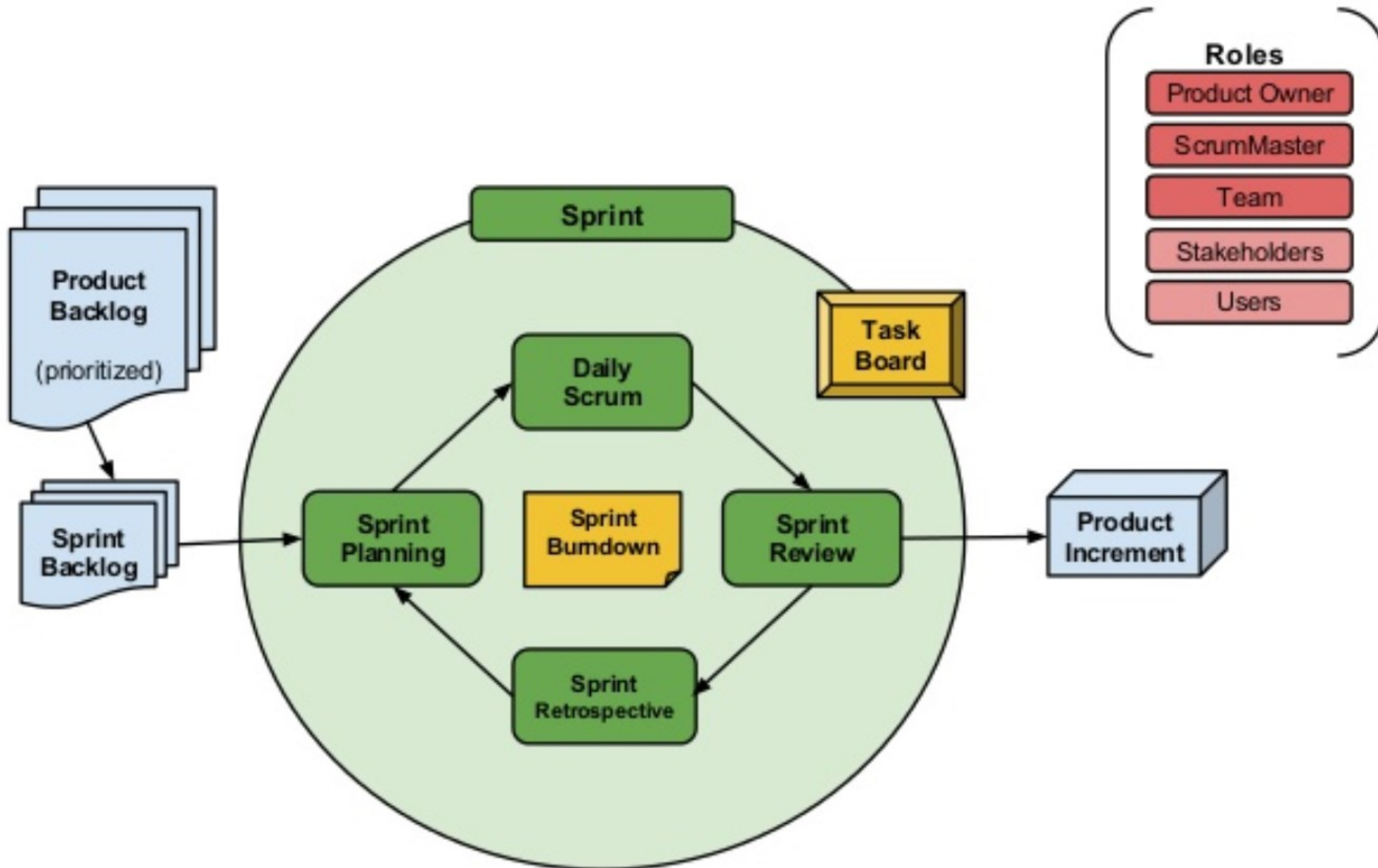
- Sprint se sastoji iz 4 događaja:
 - (3) **Pregledanje** (Revizija, *Sprint Review*)
 - Pregled šta je od posla završeno, a šta nije
 - Po potrebi se ažurira Proizvodni backlog (lista preostalih poslova)
 - Završeni posao se može prezentovati zainteresovanim stranama (demo)
 - Trajanje revizije ograničeno na 4 sata
 - Da li se priče često prenose, iz sprinta u sprint?
 - (4) **Retrospektiva** - svrha: poboljšanje procesa (sa više strana)
 - Osvrtn nakon svake revizije, a pre narednog planiranja Sprint-a. Svi članovi tima daju svoj pogled na protekli Sprint.
 - Da li se odvija svaki Sprint korektno? Šta je bilo dobro tokom Sprint-a, a šta još može da se poboljša u sledećem?
 - Trajanje retrospektive je ograničeno na 3 sata



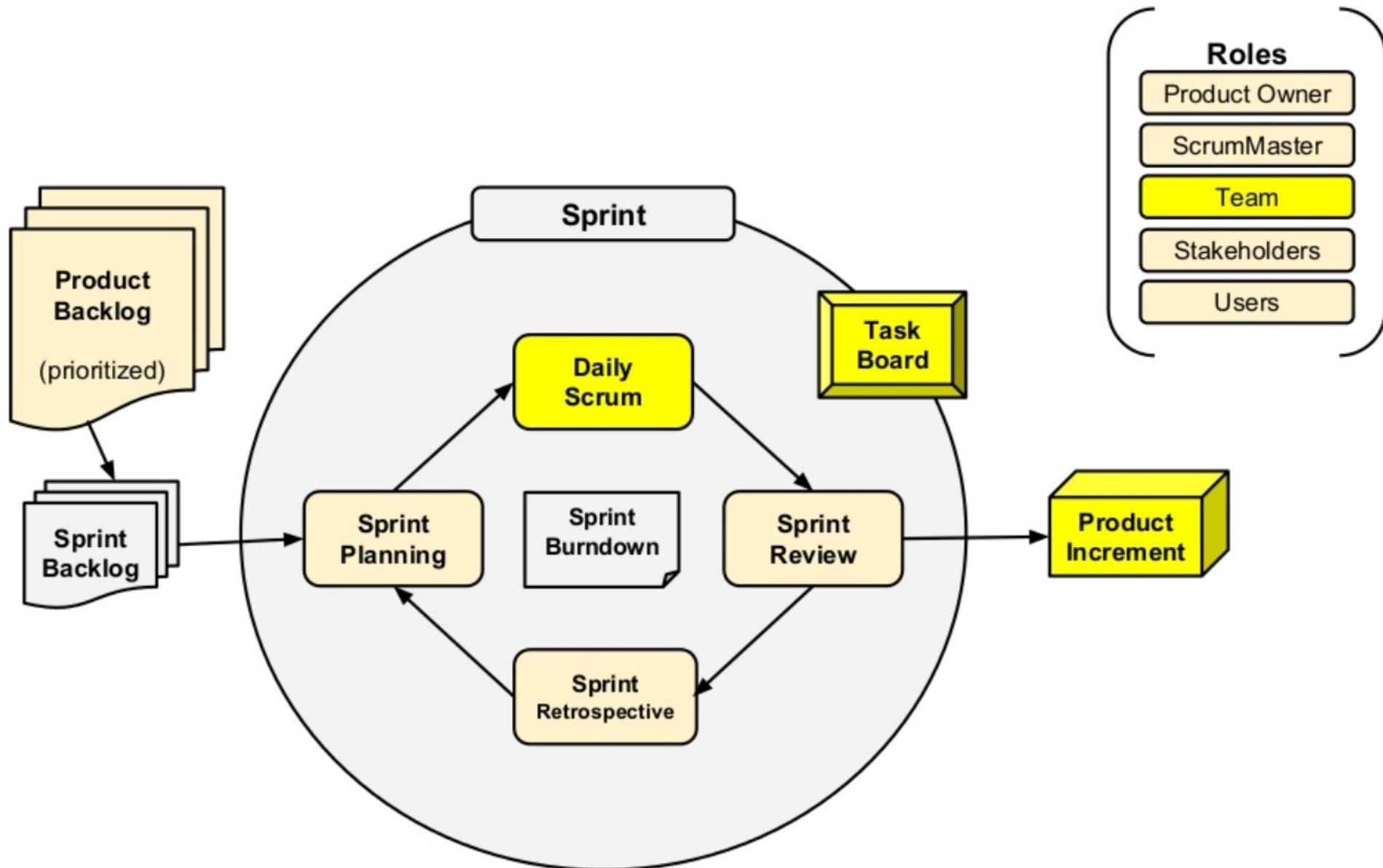
Veza artefakta i događaja



SCRUM - pregled

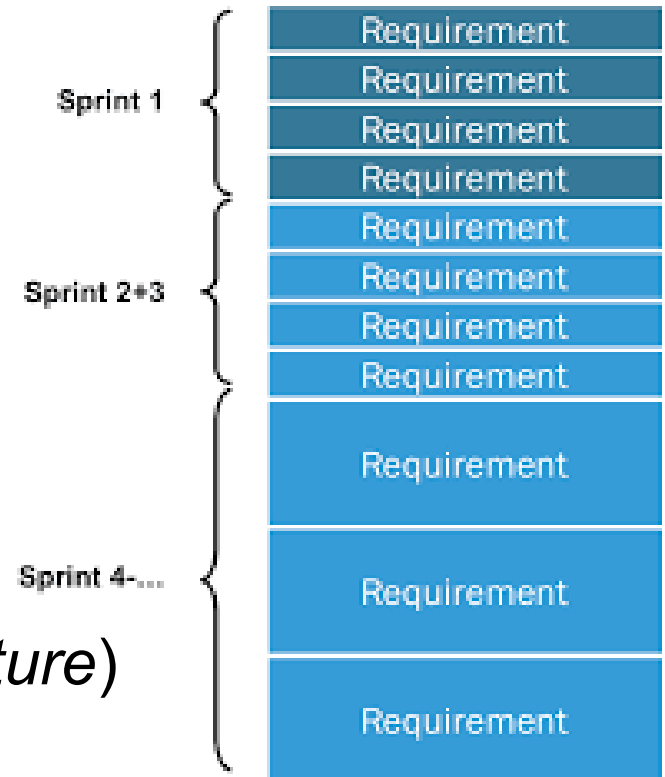


Kanban u okviru Scrum-a



Proizvodni backlog

- Lista svih zahteva, funkcionalnosti, poboljšanja, ispravki, koji zajedno čine promene koje treba implementirati u budućim inkrementima proizvoda
- PB stavke treba da imaju:
 - naziv i opis
 - prioritet
 - procenu uloženog napora
 - (poslovnu) vrednost
- Za proizvodni backlog je odgovoran Vlasnik proizvoda
- Poželjno konstruisati RBS (*Requirements Breakdown Structure*)



Stavke kod PB

- Karakteristike softvera (*Features*), prikazuju se:
 - Korisničke priče (*User stories*)
 - Dijagrami slučajeva korišćenja (*Use cases*)
 - Scenario slučaja upotrebe (slobodnog teksta)
- Defekti u softveru, koje treba da ispravimo
- Tehnički rad (*Technical work*)
 - npr. instalirati neki alat, podesiti okruženja, poboljšati neke performanse unutar softvera, sigurnost, i sl.
- Sticanje znanja (*Knowledge acquisition*)
 - npr. učiti nove tehnologije i alate, učiti neku metodologiju



Važne karakteristike kod PB

- DEEP pravilo: *Detailed appropriately + Emergent + Estimated + Prioritized.*
- Detaljnost, koliko je potrebna
 - Stavke u PB se razlikuju u nivou detalja (one koje radimo ranije, i koje su prioritetnije su i detaljnije, one na kojima ne radimo biće sa manje detalja)
- Hitnost
 - PB je živ dokument, koji se neprestano menja. Zdrav PB treba da ima dodate nove stvari, neke odbačene ili promenjene.
- Procenjenost
 - U određenom trenutku imati procenu veličine PB koja odgovara uloženom naporu potrebnom za razvoj tog proizvoda
 - Velike stavke visokog prioriteta razbijati u manje priče, pre nego što ih proglasimo spremnim za Sprint
- Prioritetnost - davati prioritete prema izdanju (verziji)

Sprint backlog

- Lista stavki koje moraju da se urade u toku sledećeg Sprint-a.
- Lista se pravi izborom najprioritetnijih stavki u PB, tako da se popuni Sprint (dovoljno da se uradi).
- Samoorganizovanje razvojnog tima:
Razvojni tim deli SB stavke na zadatke, a zadaci se ne dodeljuju, već ih preuzimaju članovi tima sami.
- Za predstavljanje zadatka i statusa koristi se tabla sa zadacima (slično kao Kanban tabla)



Tabla kod Scrum-a

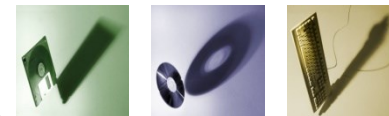
Sprint 1

STORY	TO DO	IN PROGRESS	TO VERIFY	REJECTED	DONE / CLOSED
<p>Sprint Impediments</p> <p>+</p>		<p>37</p> <p>No access to test environment</p>			
<p>Tested</p> <p>Contact form 25</p> <p>Christoph Zierz 1</p>	<p>70</p> <p>Create wireframes</p> <p>Christoph ...</p>	<p>33</p> <p>Select plugin</p> <p>Claire Gull...</p>			<p>54</p> <p>Style-the form</p> <p>Christoph ...</p>
<p>Developed</p> <p>Cookie warning 24</p> <p>Christoph Zierz 3</p>			<p>64</p> <p>Write tests</p> <p>Christoph ...</p>		
<p>In development</p> <p>PDF Export 18</p> <p>John Doe 5</p>			<p>35</p> <p>Integrate export library</p> <p>John Doe</p>		
<p>In development</p> <p>Newsletter registration form 26</p> <p>Peter Nielsen 3</p>		<p>60</p> <p>Test newsletter</p> <p>Birthe Lin... 10</p> <p>50</p> <p>Test integration with</p> <p>Peter Niel... 10</p>			
<p>Specified</p> <p>SEO optimized footer 29</p> <p>Unassigned 1</p>	<p>87</p> <p>Create visuals</p> <p>Birthe Lin...</p>				



Inkrement proizvoda

- Inkrement = skup svih PB stavki koje su završene tokom tekućeg, ali i svih prethodnih Sprint-ova
- Novi inkrement mora biti u upotrebljivom stanju i da ispunjava definiciju završenog
- Inkrement mora biti upotrebljiv bez obzira da li će ga vlasnik proizvoda pustiti u upotrebu ili ne



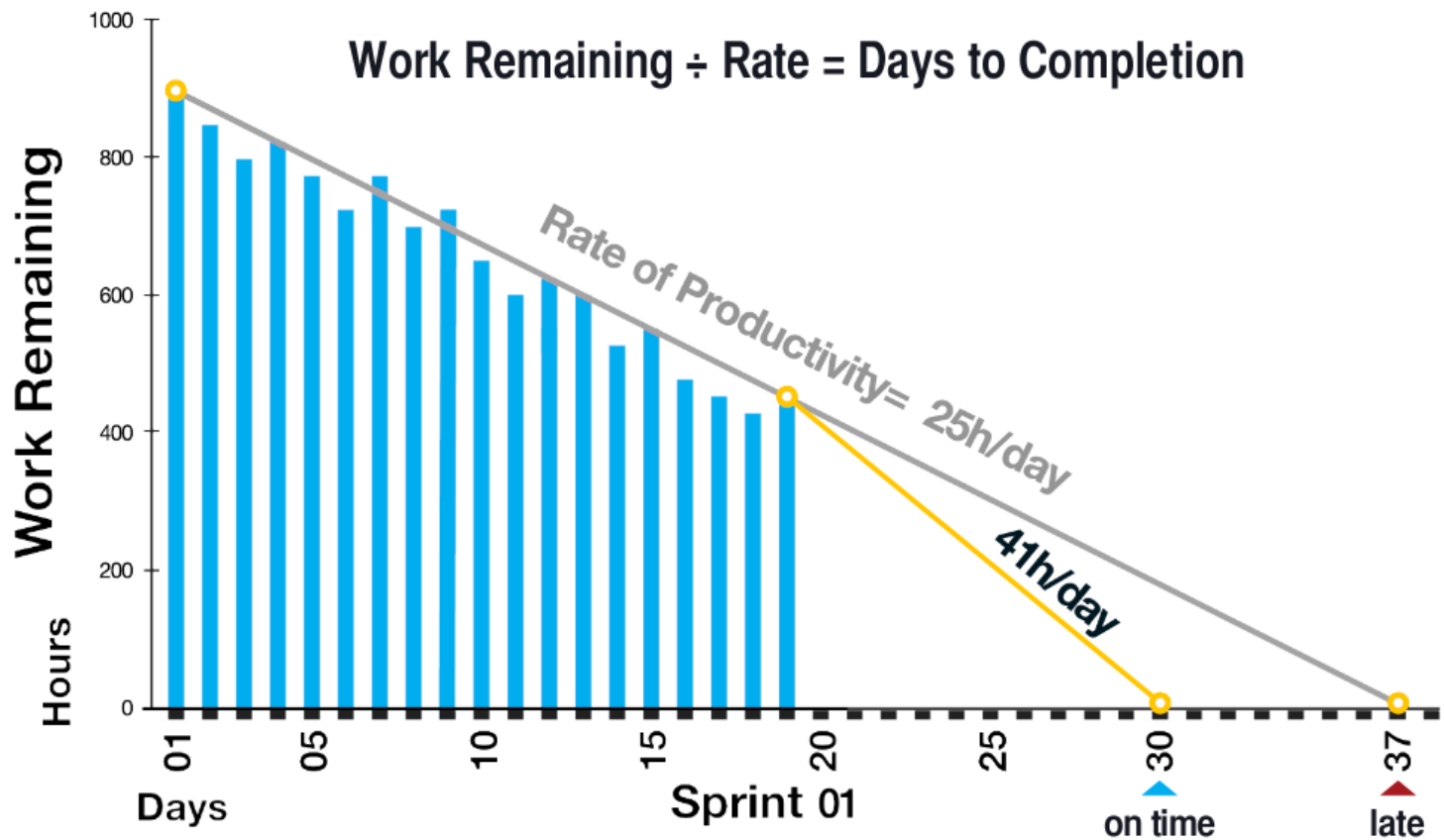
Vremenska ograničenja u Scrum-u

- Priče (vremenski ograničene) moraju biti manje da bi se „uklopile“ u dužinu jednog sprint-a.
- Razbijanje priče tako da postoje neprirodne tačke preseka => poteškoće za testiranje i *deployment*
- Premali broj priča je takođe problem, i na kraju sprint-a možda nećemo dobiti inkrement softverskog proizvoda
- Povećanje zavisnosti između većeg broja priča zahteva više koordinisanja i dobro planiranje posla
- Pojačan je proces testiranja (i ponovnog testiranja) zbog mnogo malih nepotpunih priča
=> zahtevaju se mnogo veći naponi (teže testiranje)



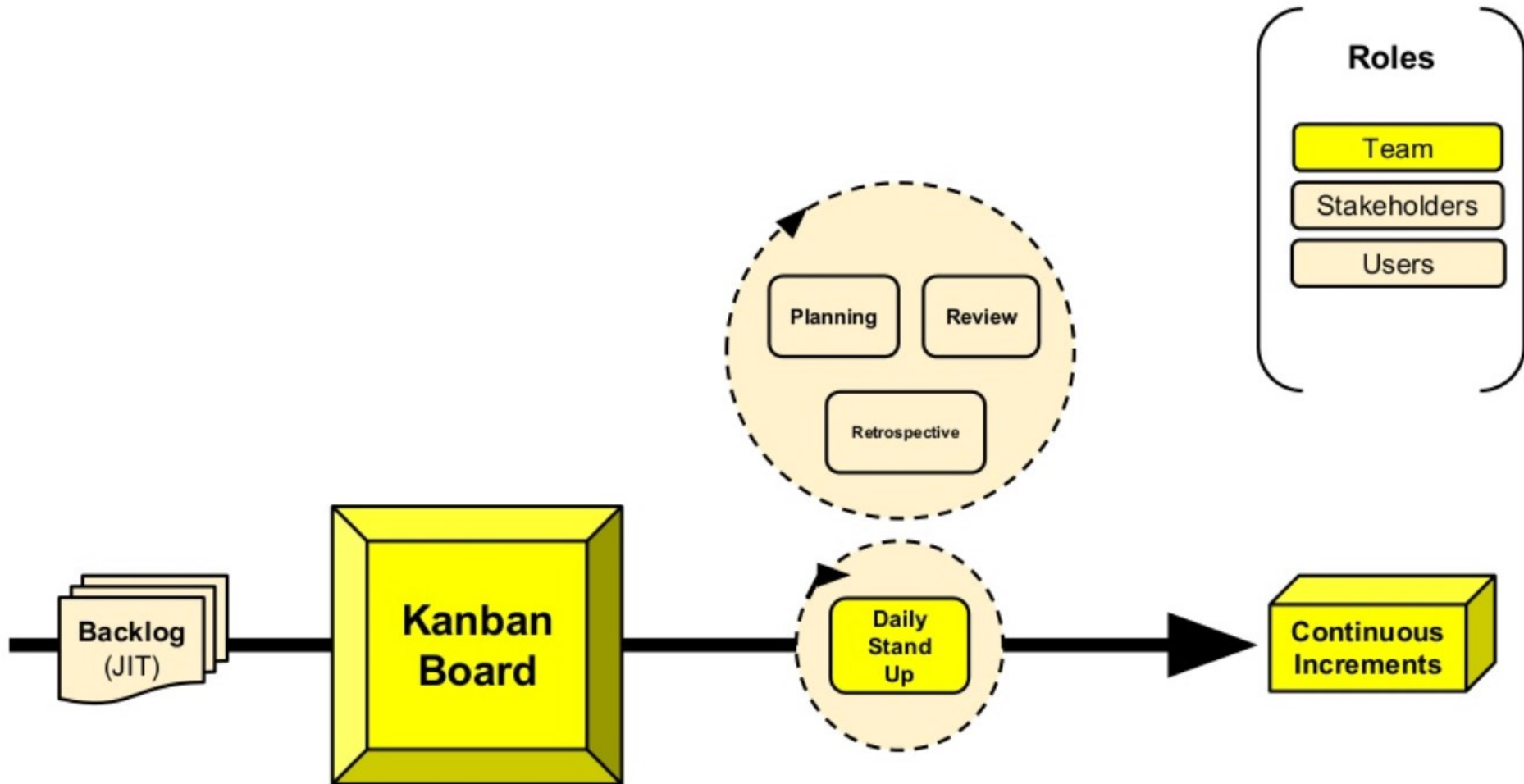
Kako se prati napredak?

- *Burndown* dijagram – dijagram preostalog posla
- X-osa (vreme projekta), Y-osa je napor u časovima
- Nagib *Burndown* dijagrama, naziva se brzina (*velocity*) i predstavlja prosečnu produktivnost po danu



Kanban sa Scrum-om = Scrumban

- Imamo tablu iz Kanban, ali i nešto što liči na Sprint

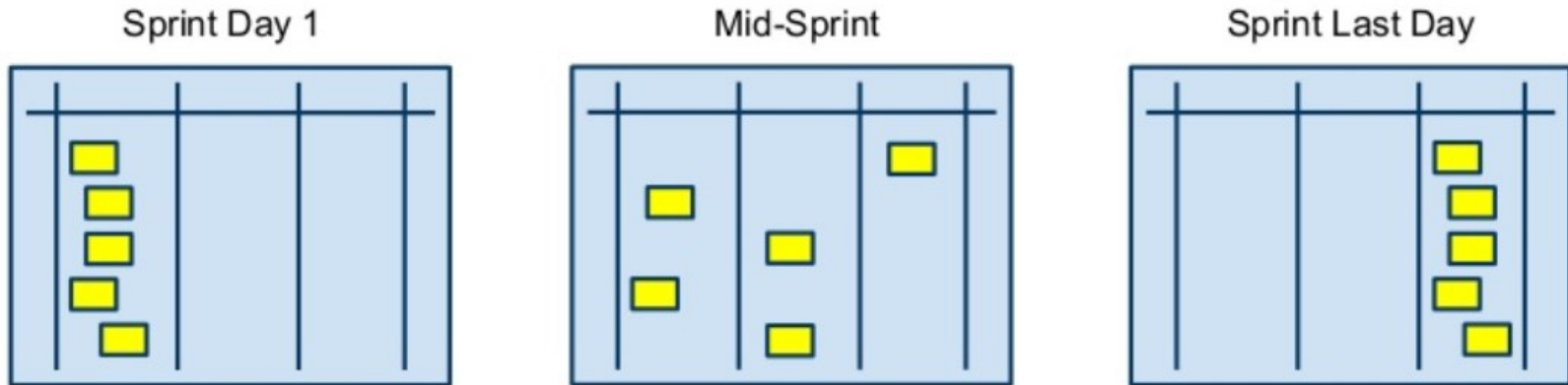


Scrum i Kanban - sličnosti i razlike

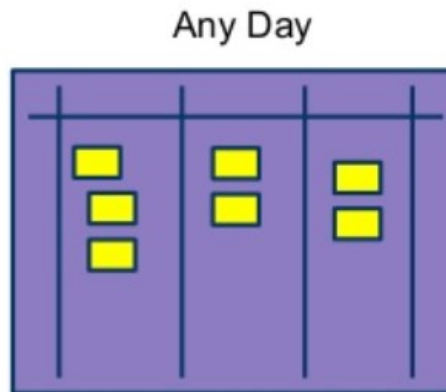
	Scrum	Kanban
Tabla/Artefakti	tabla, backlogs, burndown dijagrami	samo table
Događaji	planiranje sprinta, dnevni scrum, pregled sprinta, retrospektiva sprinta	dnevni scrum, pregled/retrospektiva na grupi frekventnih i planirajućih aktivnosti na redu
Iteracije	DA (sprint)	NE (neprekidan tok)
Procena	DA	NE
Timovi	mora biti višefunkcionalni tim	može biti specijalizovan tim
Korisničke role	Product Owner, Scrum Master, tim	tim, nema posebnih uloga
Timski rad	saradnja može u nekim zadacima	grupisati se, da postignemo cilj
Work in Progr.	kontrolisano od sadržaja sprinta	kontrolisano u stanjima rad.toka
Promene	treba da sačekaju za sledeći sprint	dodaju se na tablu po potrebi
Metrike	brzina	vreme realizacije, vreme ciklusa, WiP
Proizvodni Backlog	spisak prioriternih i procenjenih priča	pristup „tačno na vreme“ (<i>Just in Time</i>)
Prepreke	rešavati ih odmah	izbegavati ih

Scrum vs Kanban (1)

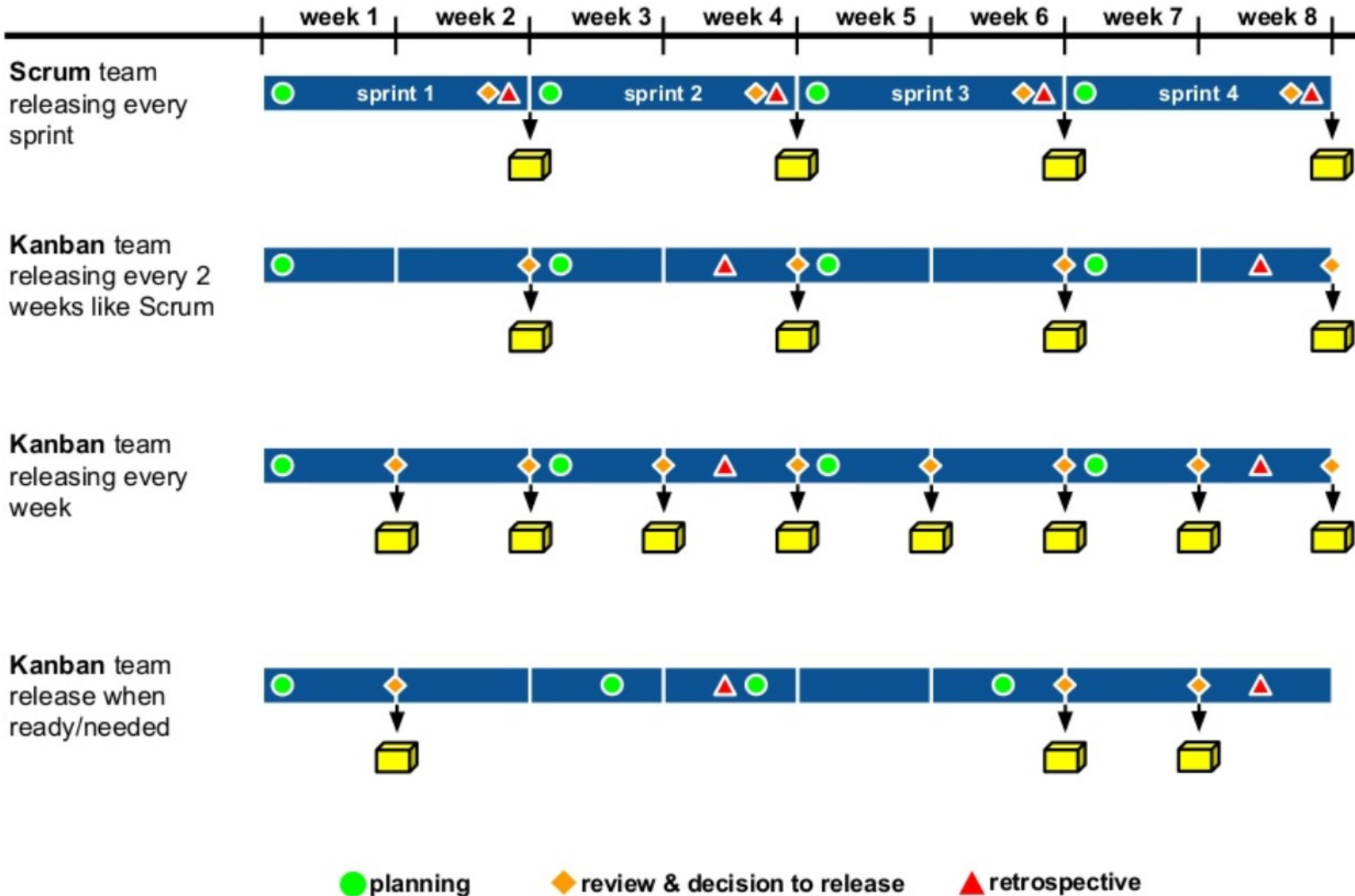
Scrum



Kanban



Scrum vs Kanban (2)



Idealno okruženje za Kanban

- Rad vođen događajima
 - dobar za korišćenje kod help-desk / tehničke podrške
 - kod projekata sa frekventnim i veoma neočekivanim korisničkim pričama ili projekata gde se izazivaju česte greške (recimo samo projekat u kome testiramo softver)
 - kod projekata održavanja softvera
- Kompanije koje su uvele Scrum su se bazirale na razvoju novih softverskih proizvoda
 - rad je ispred razvoja Sprint-a (R&D)
 - rad je nakon razvoja Sprint-a (sistemsko testiranje, izbacivanje produkcione verzije - *deployment*)
- Unapređuje dobru organizaciju unutar kompanija



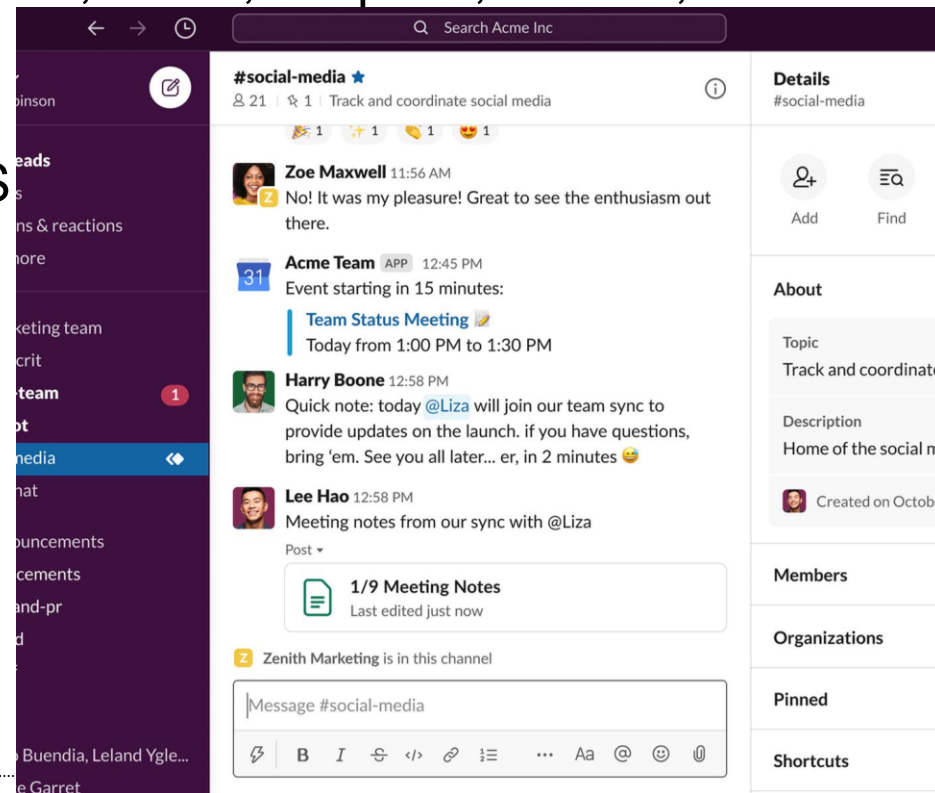
Alati za rad u softverskoj industriji

- Alati za komunikaciju i kolaboraciju:
 - Slack
 - Microsoft Teams
 - Google Hangouts / Google Meet
 - Zoom
 - Cisco Webex
 - Discord
 - Trillian
- Alati za agilni razvoj projekata i upravljanje projektima:
 - Trello
 - Atlassian JIRA
 - Microsoft Project
- Sistemi za verzionisanje i repozitorijumi:
 - Github
 - Atlassian Bitbucket
- Alati za evidenciju bagova (*Bug Tracking* sistemi)
 - Atlassian JIRA
 - Bugzilla
 - Mantis
 - Redmine
 - IBM Rational ClearQuest



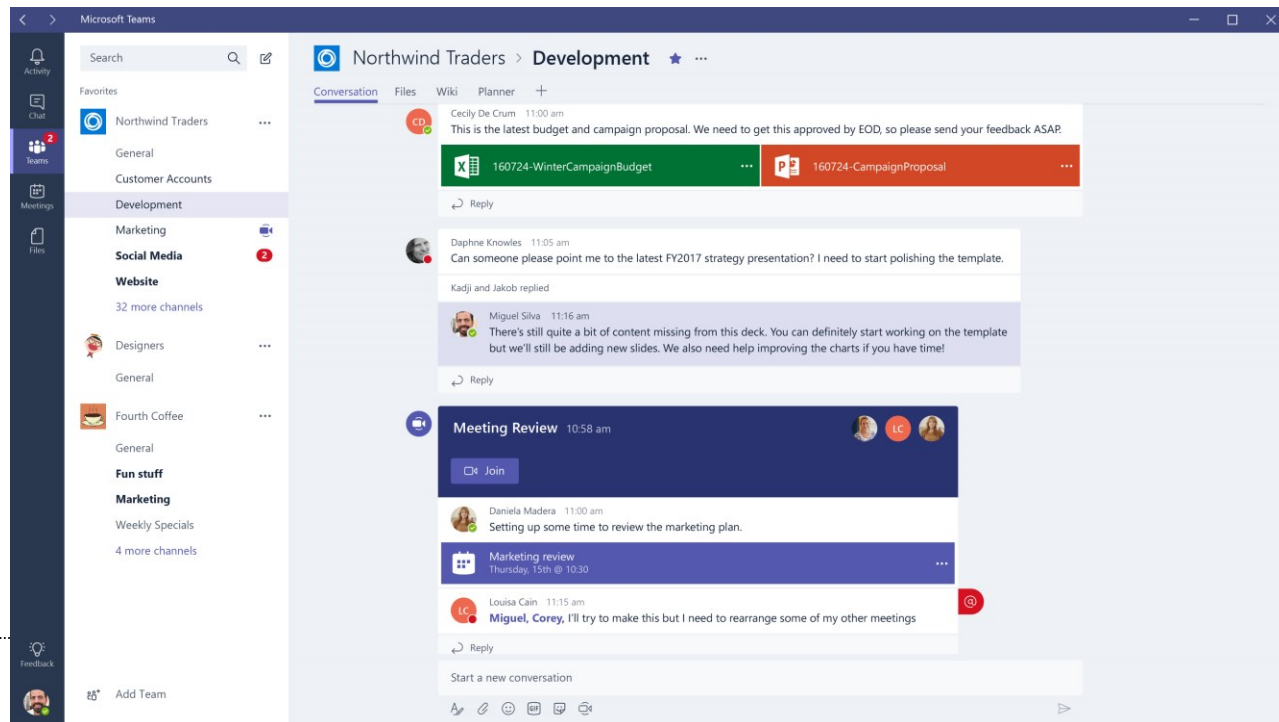
Alati za rad u softverskoj industriji (1)

- Slack:
 - Aplikacija za „ćaskanje“, nudi slične pogodnosti kao raniji IRC-alati (*Internet relay chat*) – sobe za ćaskanje (kanale), organizovane po temama, privatnim grupama ili direktnim dopisivanjem.
 - Integriše se sa: Google Drive, Trello, Dropbox, Heroku, GitHub,...
 - Postoje i mobilne verzije aplikacije za Android i iOS
 - <https://slack.com/>



Alati za rad u softverskoj industriji (2)

- Microsoft Teams:
 - Sistem za komunikaciju i kolaboraciju u softverskim timovima
 - Omogućava ćaskanja, video sastanke, skladište fajlova sa mogućnošću razmene fajlova, i integracija drugih aplikacija
 - Predstavlja nadogradnju aplikacije *Skype* za komunikaciju i konkurent je alatima kao što je *Slack*
 - Servis je integrisan sa Office 365 platformom



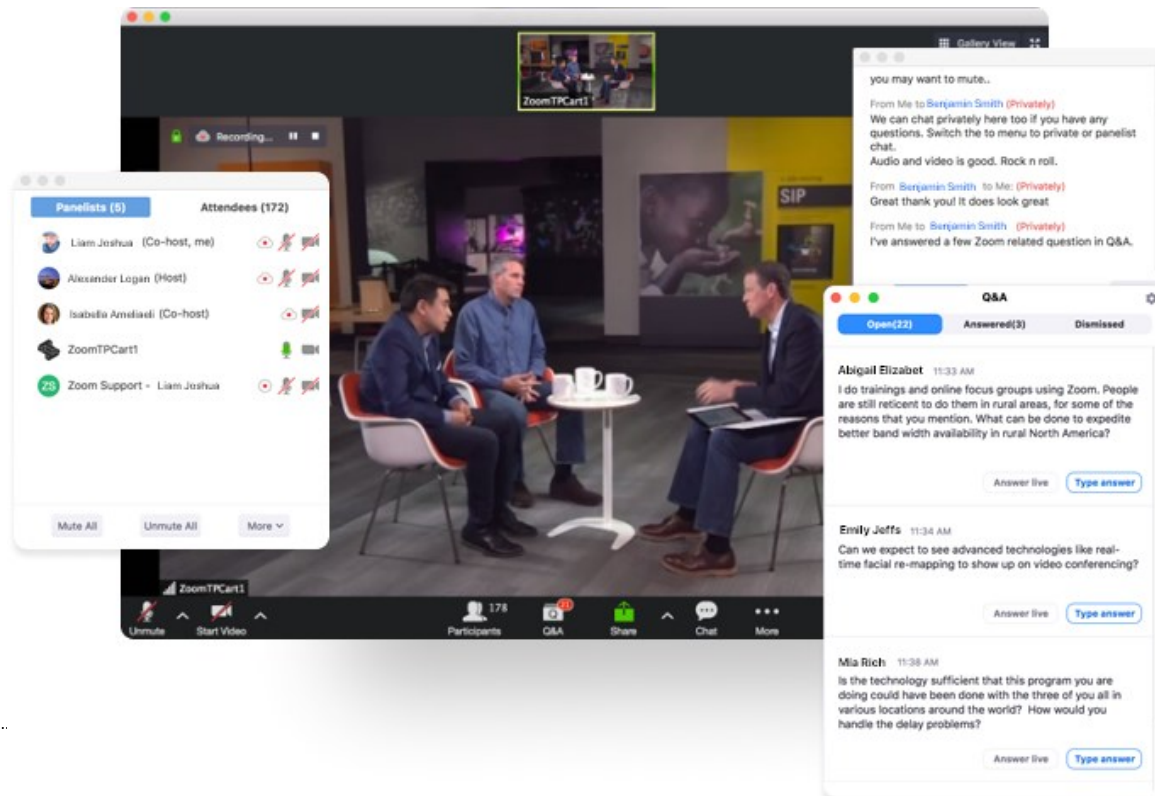
Alati za rad u softverskoj industriji (3)

- Google Hangouts:
 - Još jedan iz grupe softvera za komunikaciju, razvijen 2013. godine kao spoj *Google Messenger* i *Google Talk*.
 - 2017. počeo razvoj *Hangouts*-a sa ciljem da to bude alat za komunikaciju u kompanijama, između 2 i više korisnika
 - *Hangouts* je danas deo G Suite linije proizvoda i čine ga dva proizvoda: *Google Meet* i *Google Chat*.
 - Od aprila 2020. god. Google Meet kao komercijalni alat počinje da se nudi besplatno korisnicima.



Alati za rad u softverskoj industriji (4)

- ZOOM
 - Alat za komunikaciju zasnovan na konferencijskim sobama
 - Lak za upotrebu, ne morate se registrovati, dovoljno je da dobijete link za pristup i kredencijale za pristup (ako želite samo prisustvo)
 - Veoma dobar kvalitet, podrška za pokretanje iz Outlook-a ili Gmail-a
 - Podrška i do 1000 video učesnika u jednom pozivu i prikaz čak 49 ekrana (kamera)
 - <https://zoom.us/>

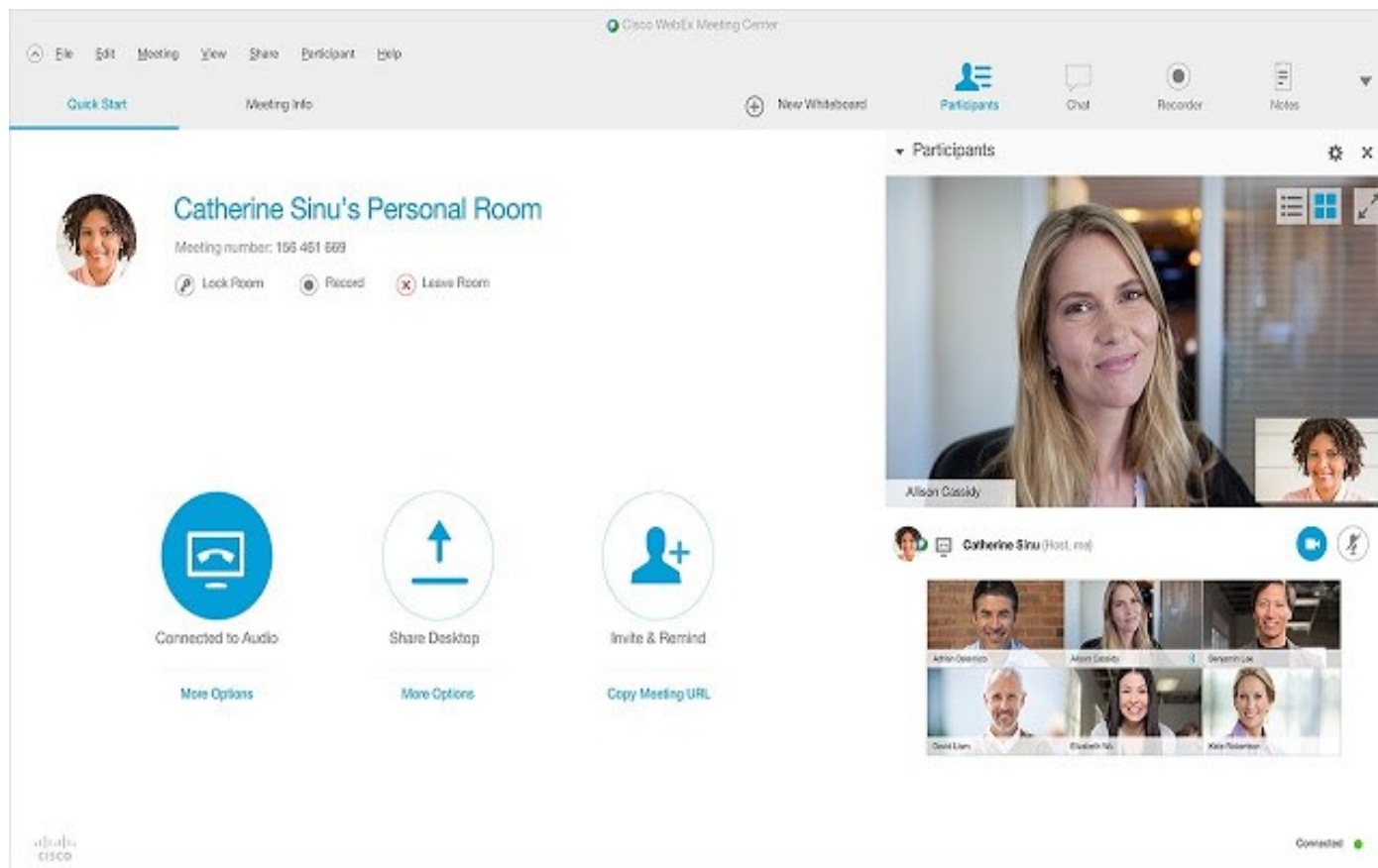


zoom

Alati za rad u softverskoj industriji (5)

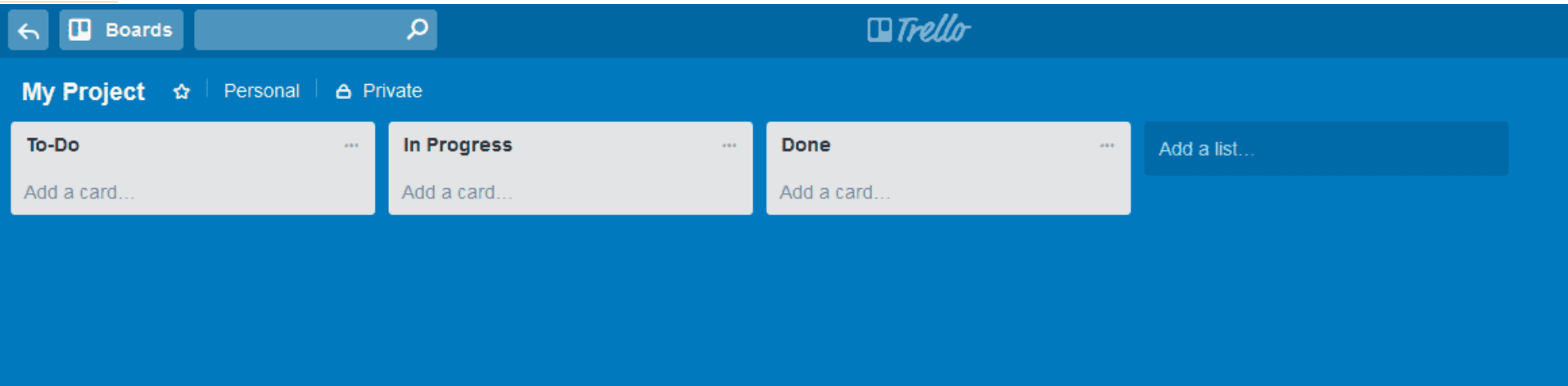
- Cisco Webex:

- Nastao 2007. godine kada je kompanija Cisco kupila Webex
- Onlajn sastanci i webinar, onlajn kursevi i treninzi



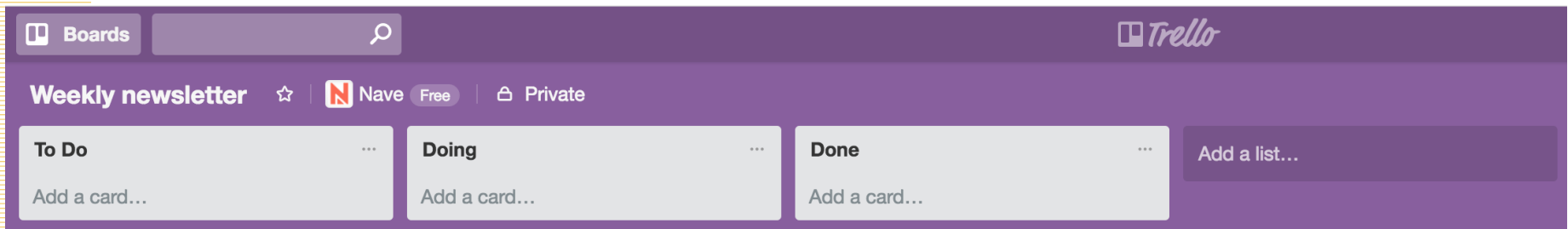
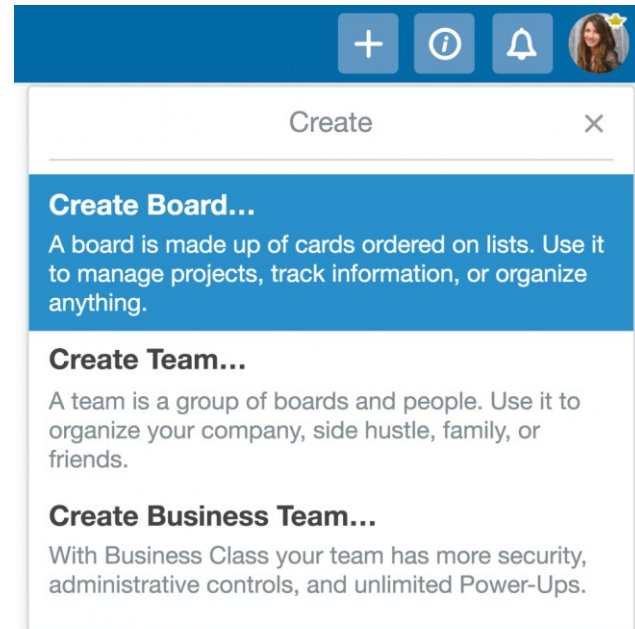
Alati za rad u softverskoj industriji (6)

- Trello:
 - Jednostavan i efikasan alat za upravljanje manjim projektima
 - Dobro definisanim radnim tokom, može da se koristi i kod složenijih projekata
 - Besplatan *online* alat koji omogućava crtanje Gantt dijagrama i praćenje projekta kroz vreme, korišćenje elektronske Kanban table
 - <https://trello.com/>



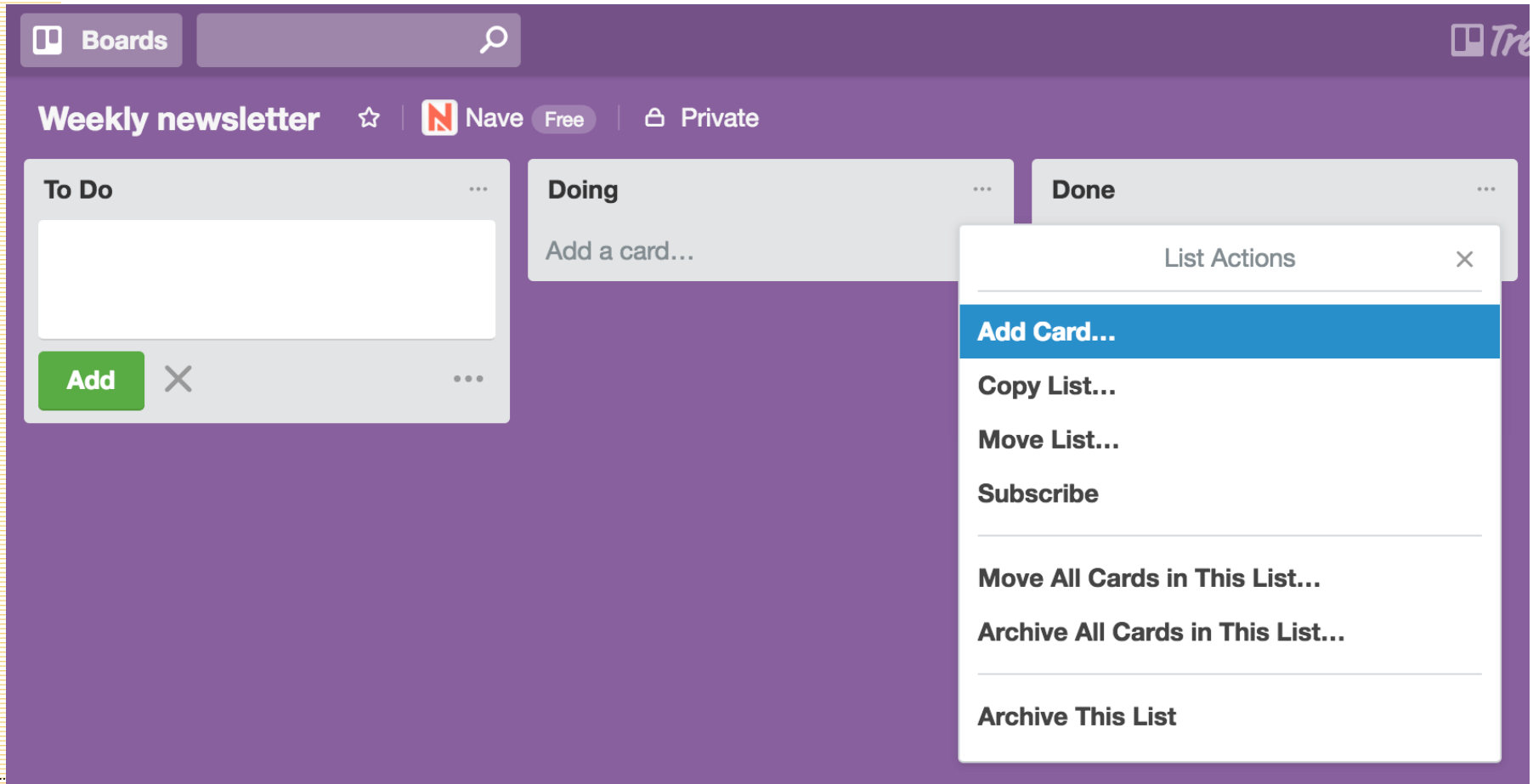
Trello (1)

- Kreiranje table, tima, i kako izgleda tabla
- Osim standardnih listi (To Do, Doing, Done) mogu se dodati i nove



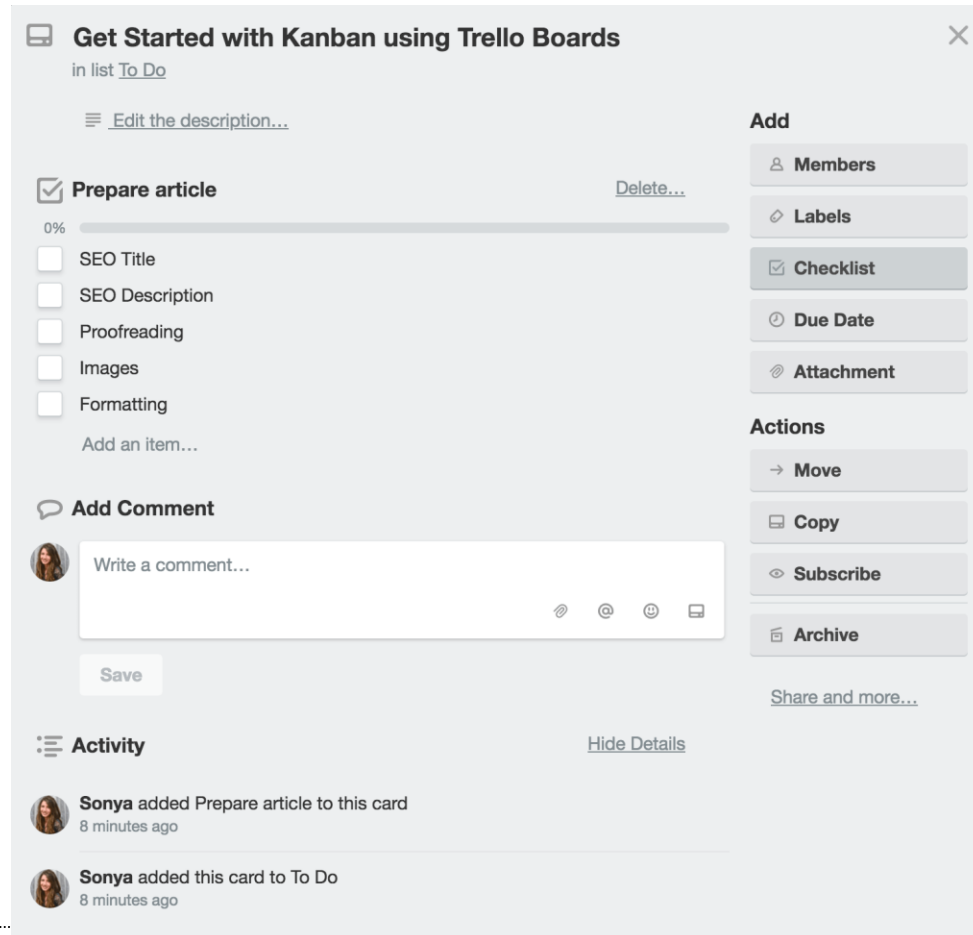
Trello (2)

- Mogućnost dodavanja kartica u okviru table i određene liste



Trello (3)

- Zadatak može da se izdela na manje
- Unosi se opis posla, članovi, labele, čekliste
- Mogu se primeniti različite akcije (pomeranje, kopiranje, arhiviranje)



The screenshot shows a Trello card interface. At the top, the card title is "Get Started with Kanban using Trello Boards" with a close button (X) in the top right corner. Below the title, it says "in list To Do" and "Edit the description...". The main content area features a checklist titled "Prepare article" with a progress bar at 0%. The checklist items are: "SEO Title", "SEO Description", "Proofreading", "Images", and "Formatting", each with an unchecked checkbox. Below the checklist is an "Add an item..." input field. Underneath is a comment section with the heading "Add Comment", a profile picture of a woman, a text input field with the placeholder "Write a comment...", and icons for attachments, mentions, emojis, and replies. A "Save" button is located below the comment field. At the bottom of the card, there is an "Activity" section with a "Hide Details" link. The activity log shows two entries: "Sonya added Prepare article to this card 8 minutes ago" and "Sonya added this card to To Do 8 minutes ago". On the right side of the card, there is a sidebar with an "Add" section containing buttons for "Members", "Labels", "Checklist", "Due Date", and "Attachment". Below that is an "Actions" section with buttons for "Move", "Copy", "Subscribe", and "Archive". At the very bottom of the sidebar is a "Share and more..." link.

Trello (4)

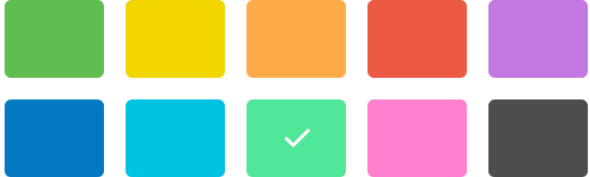
- Dodeljivanje boje zadatku (kartici)
- Dodavanje članova tima određenom zadatku


← Change Label ×

Name


Technical Writing

Select a color



 No color.
This won't show up on the front of cards.


Save **Delete**

 **Members**

Members ×

Search members

Board Members

 **Sonya (sonya_s)**

[Show other team members...](#)

Trello (5)

- Izgled Kanban table na jednom projektu

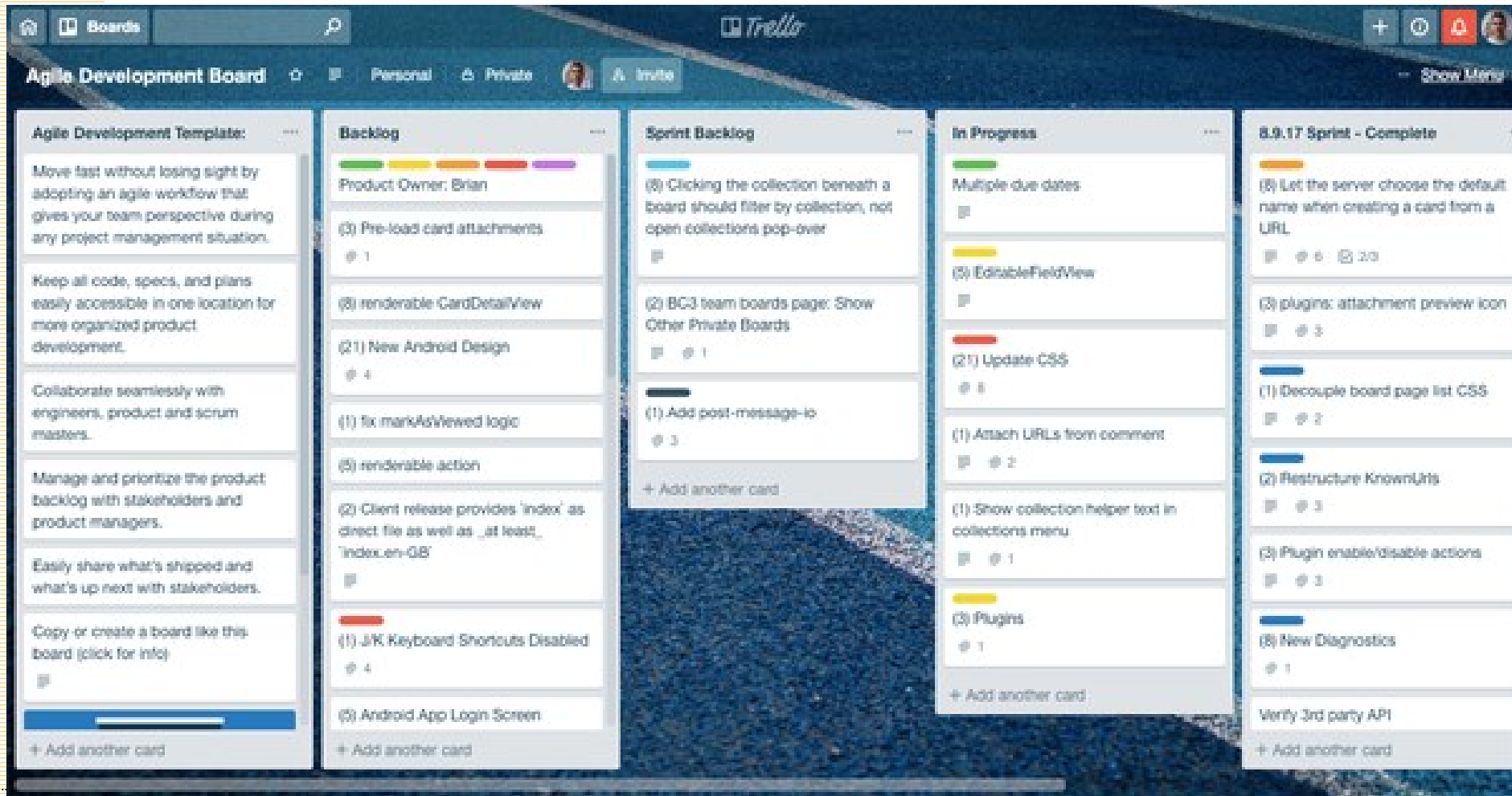
The screenshot shows a Trello board titled "Kitchen project" with a search bar and navigation icons at the top. The board is organized into five columns: "Ideas", "To Do", "In Process", "Done!", and "Waiting For".

- Ideas:**
 - Installation: Install pot rack over the island (3 items, 2/3 completed)
 - Plumbing: Install new sink (1 item, 3/5 completed)
- To Do:**
 - Plumbing: Adjust water pressure of the sink (19 Jan 2019, 1/4 completed)
 - Remove old refrigerator and stove
 - Buy paint for cabinets
 - Install new flooring (1 item, 0/5 completed)
 - Get a new window valance to match the cabinet colors
 - Pricing for double oven vs. single
- In Process:**
 - Call contractor (1/2 completed)
 - Design new kitchen space (26 Dec, 2 items)
 - Research potential appliances (0/4 completed)
- Done!:**
 - Replace drawer knobs with antique ones (2 items)
 - Order glass stovetop
 - Tile delivery for backsplash (1 item)
- Waiting For:**
 - Pantry and spice rack (1 item)
 - Plumbing: Pick countertop colors (9 Feb 2018)
 - Bar stool measurement
 - Plumbing: Pick faucet to match new sink (1 item)

The board includes various visual elements such as photos of kitchen fixtures, a kitchen floor image, and a kitchen floor plan diagram.

Trello (6)

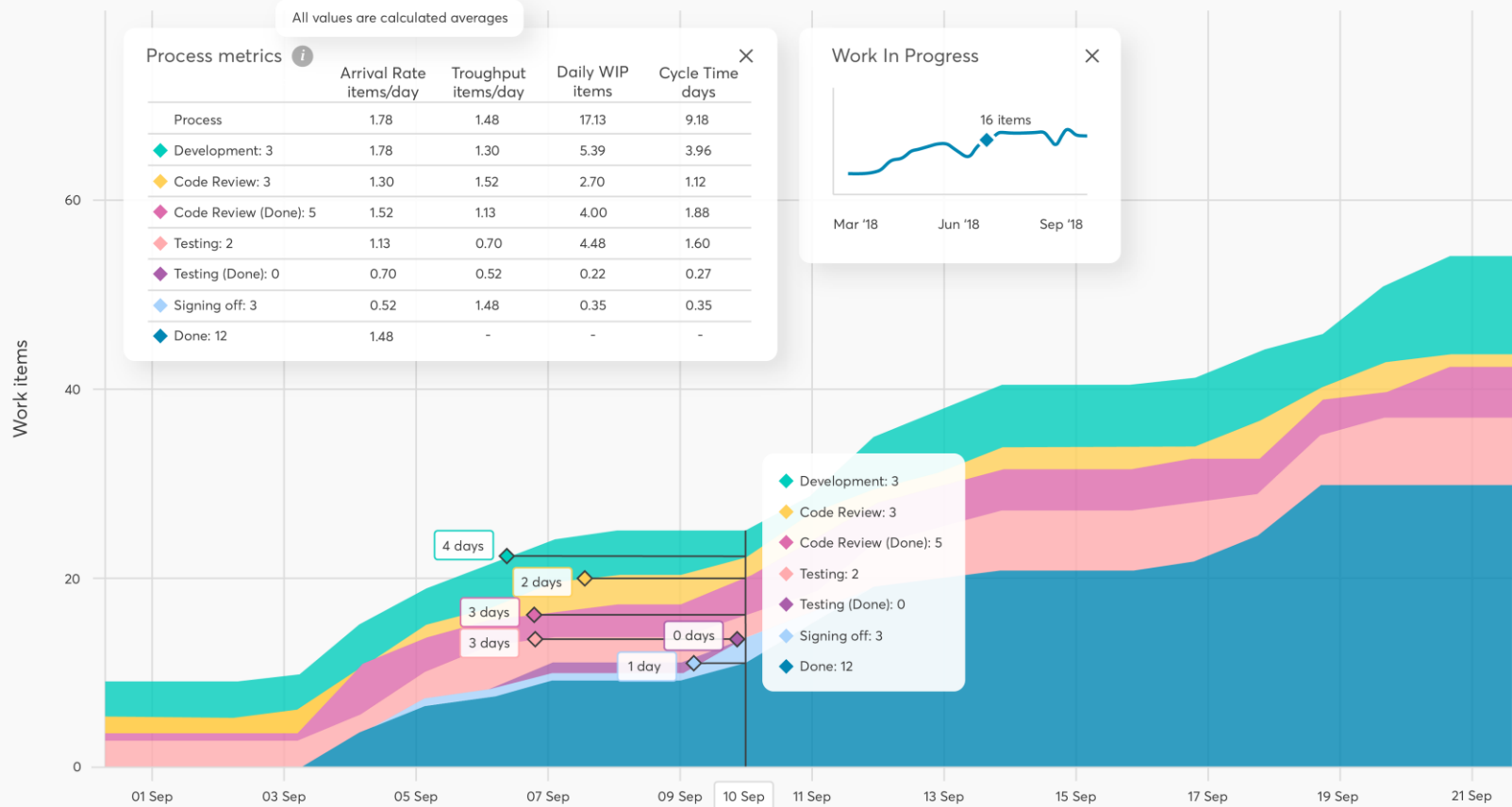
- Može i kao Scrumban, sa sprint-ovima



Trello - Kumulativni dijagram toka (1)

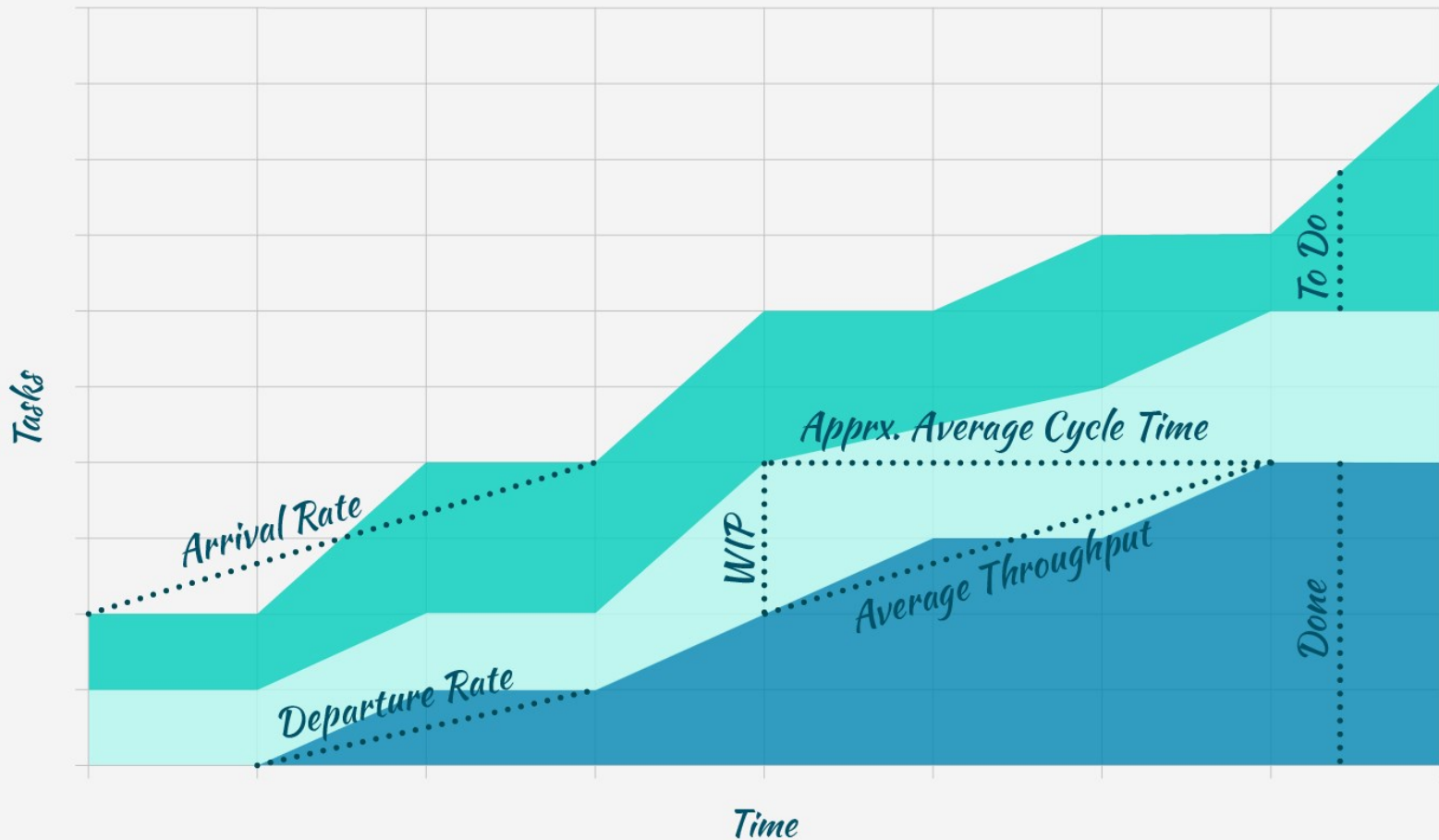
- Za praćenje ukupne količine zadataka u našem procesu, kao i količine zadataka u svakom stanju procesa

Cumulative Flow Diagram



Trello - Kumulativni dijagram toka (2)

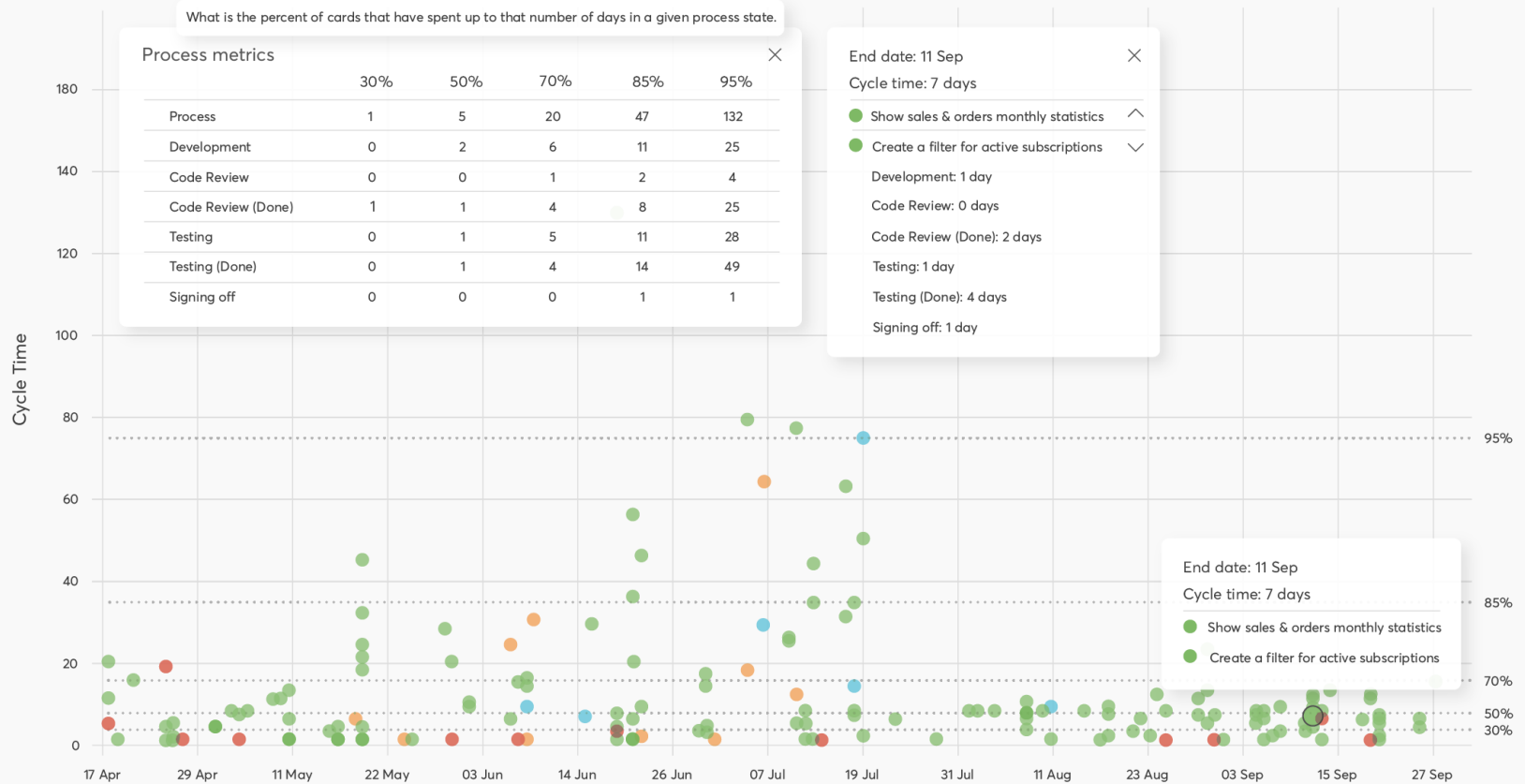
- Može se proračunati prosečni protok između 2 tačke (*Average Throughput*)



Trello - Vreme ciklusa (1)

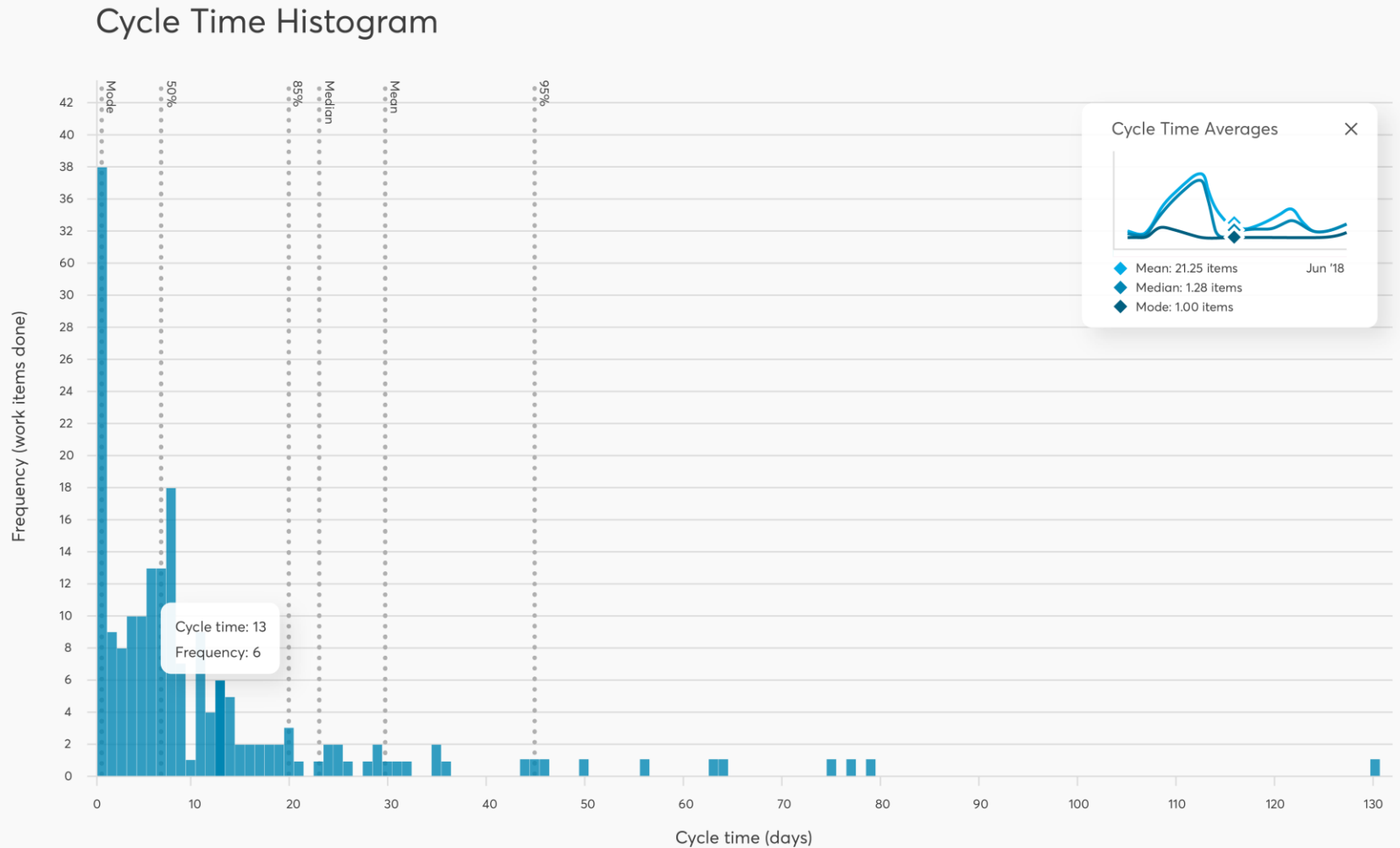
- Grafikon za merenje vremena od početka do kraja svakog zadatka (1 tačka = 1 zadatak)

Cycle Time Scatterplot



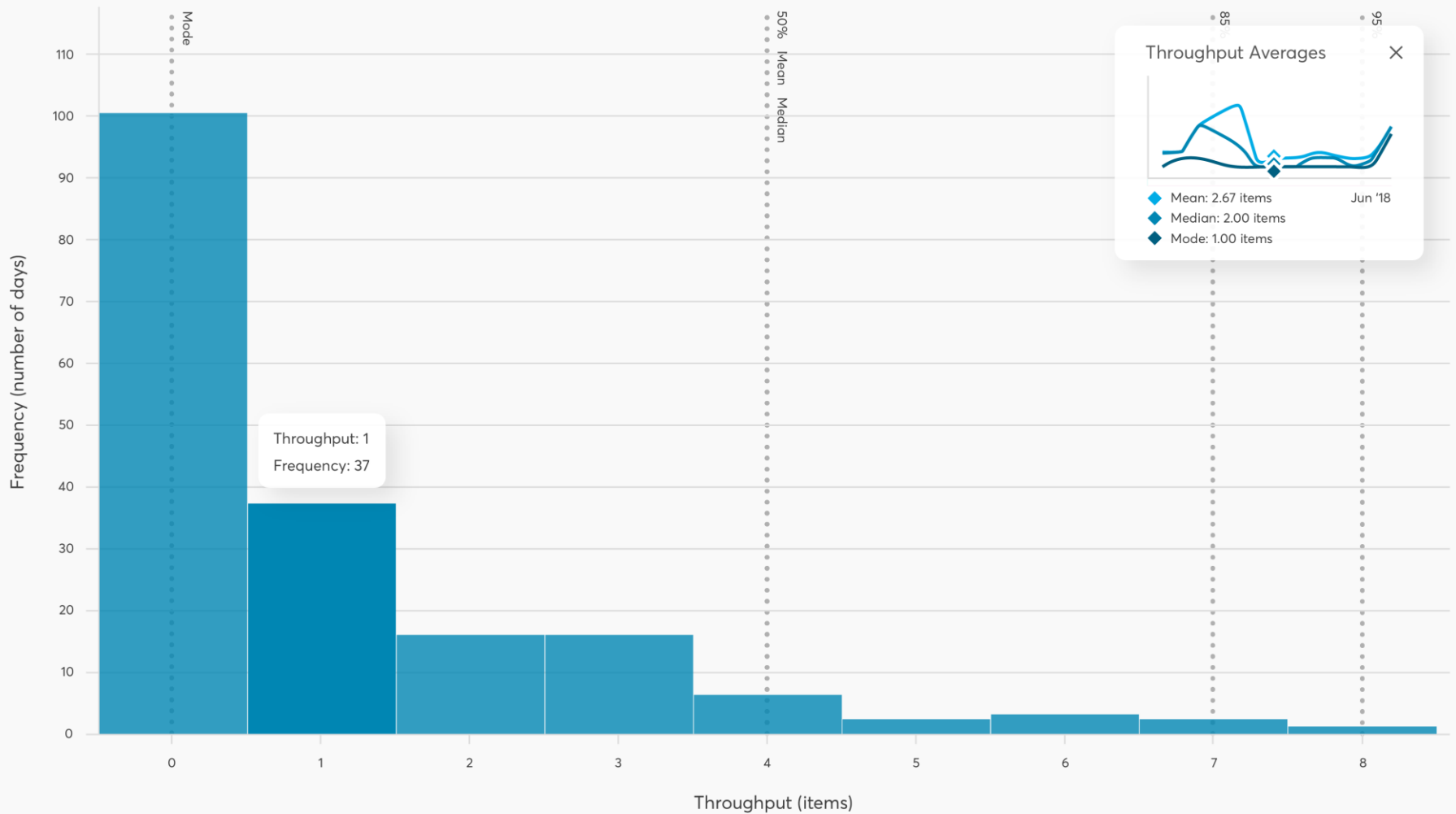
Trello - Vreme ciklusa (2)

- Histogram za lako merenje prosečnog, max i min vremena ciklusa



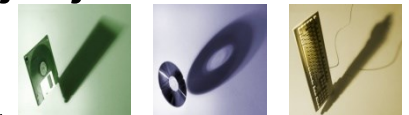
Trello - Dijagram propusnosti

Throughput Histogram



Alati za rad u softverskoj industriji (7)

- Atlassian JIRA je alat koji omogućava:
 - Agilno upravljanje projektima
 - Sistem za evidentiranje grešaka
- Gojira - japanski naziv za Godzillu (čudovište), zbog toga što su Atlassian programeri koristili alat Bugzilla, i to im je bio nadimak tima
- Komercijalni alat, ali može besplatno da se koristi (trial)
- Sastoji se i 4 paketa:
 - *Jira Core* – glavni deo za upravljanje projektima
 - *Jira Software* – uključuje sve ono što treba da imaju menadžeri koji upravljaju agilnim projektom
 - *Jira Service Desk* – alat koji služi IT i poslovnim serviserima
 - *Jira Ops* – alat koji treba da se koristi za upravljanje incidentima



Jira Software (1)

- Ima podršku i za Scrum tablu i za Kanban tablu:
 - Scrum tabla: Backlog, Aktivni sprint-ovi, Izveštaji
 - Kanban tabla: Tabla sa karticama, Backlog (opc.), Izveštaji
- Timovi koji planiraju rad sa Sprint-ovima treba da se odluče za Scrum izgled projekta.
- Timovi koji će upravljati projektom i ograničavati WiP treba da se odluče za Kanban izgled projekta.
- Može se koristiti neka postojeća tabla (koju je neko kreirao) ili možete kreirati svoju sopstvenu.
- Projekat može imati i više tabli, a mogu se i kombinovati Scrum i Kanban, ako to želite.
- Jira Software ima veliki broj izveštaja koji možete koristiti za prikaz informacija o vašem projektu, verzijama, sprintovima, problemima, itd.

Jira Software (2) - Stavke

- Radni panel za pregled posla koji treba da se radi
- Odabirom stavke dobijaju se detalji o njoj
 - Status, Priroitet, Komponente, Labele, Verzija, Ljudi (koji izveštava i kome je dodeljena stavka), Datumi, Opis.

The screenshot displays the Jira Backlog for the 'Teams in Space' project. The main area shows a list of 7 issues in the backlog, with the top issue 'TIS-1 Spaceship tracker app updates' selected. The issue details panel on the right shows the following information:

- Status: TO DO (View Workflow)
- Priority: Medium
- Component/s: Spaceship
- Labels: rocket, software
- Affects Version/s: None
- Fix Version/s: 8.0
- Epic Link: Mars Landing
- Reporter: Captain Joe
- Assignee: Master Engineer
- Created: 13 minutes ago
- Updated: 3 minutes ago

Jira Software (3) – Backlog stranica

- (1) Brzo filtriranje – po ključu, tipu, dodeljenoj osobi,...
- (2) Brzi linkovi za nalaženje stavki koje su dodeljene samo meni ili stavki koje su skoro ažurirane
- (3) Meni za table – kreiranje, konfiguracija, kopiranje table

The screenshot shows the Jira Backlog interface for the 'Teams in Space' project. The left sidebar (callout 1) contains navigation options like 'Scrum in Space', 'Backlog', 'Active sprints', 'Releases', 'Reports', 'Issues', and 'Components'. The main area (callout 2) displays a backlog of 7 issues, including 'TIS-1 Spaceship tracker app updates' (8.0, Mars Landing), 'TIS-2 Verify the landing site', 'TIS-3 Enter the landing trajectory into the landing module', 'TIS-4 Send the pre-landing report to Earth', 'TIS-5 Cleanup the landing site', 'TIS-6 Hire the team', and 'TIS-10 Complete ignition tests'. The right sidebar (callout 3) shows details for the selected issue, including status (TO DO), priority (Medium), component (Spaceship), labels (rocket, software), and assignee (Master Engineer).

Jira Software (4) - Kreiranje table

Create an Agile board

Scrum

Scrum focuses on planning, committing and delivering time-boxed chunks of work called Sprints.

Create a Scrum board

Kanban

Kanban focuses on visualising your workflow and limiting work-in-progress to facilitate incremental improvements to your existing process.

Create a Kanban board

Cancel

Create an Agile board

- Board from an existing project
Boards can contain one or more projects.
- Board from an existing Saved Filter
An advanced option using a JQL filter.

Back Next Cancel

Name this board

Board name* Scrum-tabla

Project(s)* SI-sajt (SIS) x

Select one or more projects to be included in this board.

Back Create board Cancel

Jira Software (5) - Nova kolona

- Umesto inicijalne 3 kolone na table, možemo dodati još neku kolonu (npr. IN TEST)

Projects / RTI sajt katedre

RTI board

GROUP BY None

TO DO 5

- Izrada dijagrama sekvence
RTI-8
- Migriranje starih podataka
RTI-9
- Izbor tehnologije
RTI-10
- Implementacija javno vidljivih stranica
RTI-11
- Implementacija stranice za predmete i administratorskog unosa
RTI-12

IN PROGRESS 2 MAX: 4

- Izrada novog logotipa i vizuelni identitet
RTI-5
- Izrada dijagrama slučajeva korišćenja
RTI-7


IN TEST 1 MAX: 2

- Modelovanje baze podataka
RTI-6

DONE 1 ✓

- Analiza korisničkih zahteva
RTI-1

+ Create issue



Jira Software (6) - Nova stavka

- Tipovi: Story, Task, Bug, Epic

The screenshot displays the Jira Software interface. At the top, there is a navigation bar with the Jira logo, 'Jira Software', and several menu items: 'Dashboards', 'Projects', 'Issues', and 'Boards'. A prominent blue 'Create' button is located on the right side of this bar. Below the navigation bar, the main content area shows 'RTEST board' and 'Kanban board'. Underneath, there are 'QUICK FILTERS' for 'Only My Issues' and 'Recently Updated'. A red arrow points upwards from the 'Create' button to a tooltip that reads 'Create a new issue / bug / feature request / etc'. In the foreground, a 'Create Issue' dialog box is open. It features a 'Configure Fields' button in the top right corner. The form contains the following fields: 'Project' (RTI test (RTEST)), 'Issue Type' (Story), 'Summary' (Implementacija javnog dela veb sajta), and 'Reporter' (Dražen Drašković). Below these is a 'Component/s' field set to 'None'. At the bottom of the dialog is a 'Description' field with a rich text editor toolbar. A blue arrow points from a priority dropdown menu (showing 'Medium', 'Highest', 'High', 'Low', 'Lowest') to the 'Issue Type' field in the dialog. At the bottom right of the dialog, there are three buttons: 'Create another', 'Create', and 'Cancel'.

Jira Software (7) - Nova verzija

- Kada hoćemo neki deo proizvoda da objavimo (*Release*), onda pravimo novu verziju od završenih (*Done*) stavki

The screenshot displays the Jira Software interface for a Kanban board. The top navigation bar includes 'Jira Software', 'Dashboards', 'Projects', 'Issues', 'Boards', and a 'Create' button. A search bar is also present. The board is titled 'RTEST board' and 'Kanban board'. Below the title, there are quick filters: 'Only My Issues' and 'Recently Updated'. The board is divided into four columns: 'BACKLOG 3', 'SELECTED FOR DEVELOPMENT 1', 'IN PROGRESS 0', and 'DONE 2'. A red arrow points to the 'Release...' button in the 'DONE' column. The 'DONE' column contains two issues: 'RTEST-2 Zadatak #1' and 'RTEST-3 Zadatak #2'. The 'BACKLOG' column contains three issues: 'RTEST-1 Prica br. 1', 'RTEST-4 Podtask-1', and 'RTEST-5 Podtask-2'. The 'SELECTED FOR DEVELOPMENT' column contains one issue: 'RTEST-6 Zadatak #3'.


Jira Software (8) - Nova verzija

- Svaka verzija ima svoj naziv, datum verzije, kao i kratak opis (opciono). Navedeno je koliko stavki imamo u toj verziji.

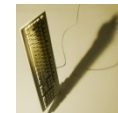
Release

2 issues will be released from **RTI test**

Version name

Release Date 

Description



Jira Software (9) - Lista verzija

- Verzije mogu da se filtriraju po brzim filterima (*Released, Unreleased, Archived*)
- Ulaskom u verziju vidimo i šta su urađene stavke

RTI test

RTEST board

Kanban board

Releases

Reports

Issues

Components

PROJECT SHORTCUTS

Add a link to useful information for your whole team to see.

+ Add link

Project settings

15 = 9 = 12 =

These are the versions in your project. Hover over the progress bar for a summary or click a version to see the complete status, including the issues, development data, and potential problems.

Learn more

Manage Versions

Releases

QUICK FILTERS: Released Unreleased Archived

Version name Start date (optional) Release date (optional) Description (optional) Add

Version	Status	Progress	Start date	Release date	Description	Actions
Verzija 1.0	RELEASED	<div style="width: 100%;"></div>		05/May/20	Prva verzija RTI sajta	<ul style="list-style-type: none">UnreleaseArchiveDeleteEdit

Atlassian Jira Project Management Software (v8.5.4#805004-sha1:0444eab) · About Jira · Release

Powered by a free Atlassian Jira community license for University of Belgrade - School of Electrical Engineering. Try Jira for your team.

ATLASSIAN

Jira Software (10) - Ažuriranje

- Da li možemo dodati nove zadatke (koji još nisu počeli) u trenutnu neku verziju projekta?

Version Verzija 1.0 **RELEASED**

📅 Start date not set Released: 05/May/20 [Release Notes](#)

Prva verzija RTI sajta

3 Issues in version




2 Issues done

0 Issues in progress

1 Issues to do

1-3 of 3

[View in Issue Navigator](#)

P	T	Key	Summary	Assignee	Status
^	☑	RTEST-3	Zadatak #2	 Dražen Drašković	DONE
^	☑	RTEST-7	Analiza starog sajta	 Dražen Drašković	BACKLOG
=	☑	RTEST-2	Zadatak #1	 Dražen Drašković	DONE

1-3 of 3

Jira Software (11) - Log work

- Izveštavanje o odrađenom poslu (More > Log work)

Log Work: RTEST-2

Time Spent* (eg. 3w 4d 12h) ?

An estimate of how much time you have spent working.

Date Started* 






Remaining Estimate

- Adjust automatically
the estimate will be reduced by the amount of work done, but never below 0.
- Leave estimate unset
- Set to (eg. 3w 4d 12h)
- Reduce by (eg. 3w 4d 12h)

Work Description

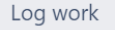
Style ▾ **B** *I* U A ▾ ^o ▾  ▾    ▾ + ▾ 


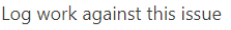
RTI test / RTEST-2
Zadatak #1

 Edit  Comment  Assign **More ▾**  Backlog  Selected for Development

Details

Type: Task

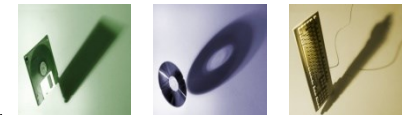
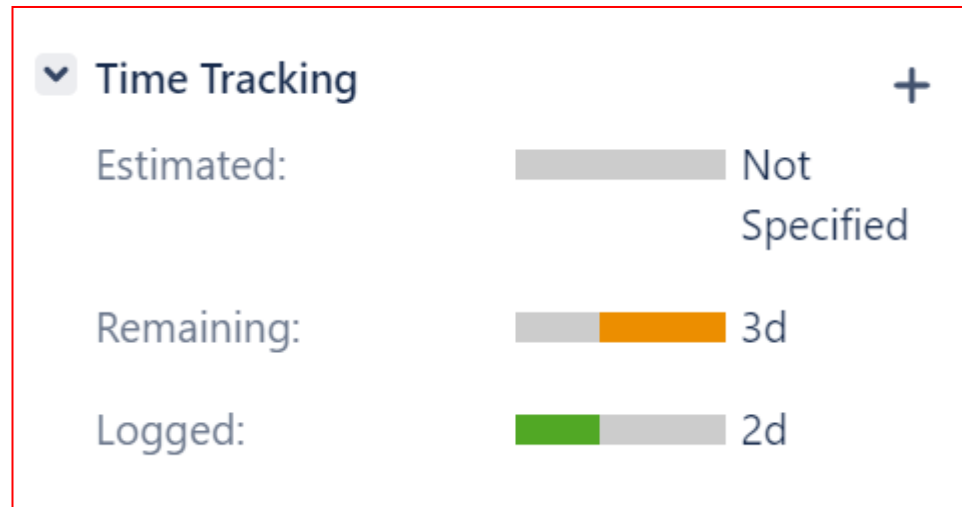
 Log work

 Agile Board  Log work against this issue

Log Cancel

Jira Software (12) - Praćenje vremena

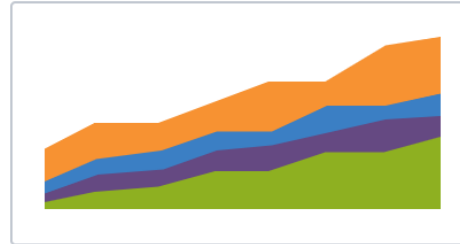
- Potrošeno vreme:
 - Procenjeno (*Estimated*)
 - Izvršeno (*Logged*)
 - Preostalo (*Remaining*)



Jira Software (13) - Izveštaji

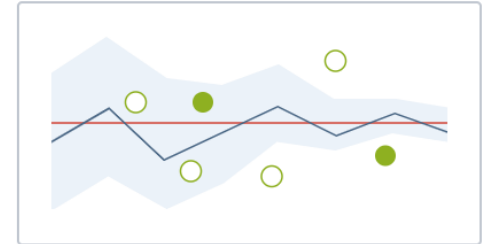
- Agilni izveštaji i izveštaji o ostvarenom vremenu

Agile



Cumulative Flow Diagram

Shows the statuses of issues over time. This helps you identify potential bottlenecks that need to be investigated.



Control Chart

Shows the cycle time for your product, version or sprint. This helps you identify whether data from the current process can be used to determine future performance.

Forecast & management



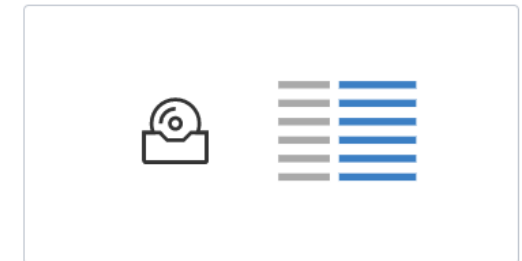
Time Tracking Report

Shows the original and current time estimates for issues in the current project. This can help you determine whether work is on track for those issues.



User Workload Report

Shows the time estimates for all unresolved issues assigned to a user across projects. This helps you understand the user's workload better.



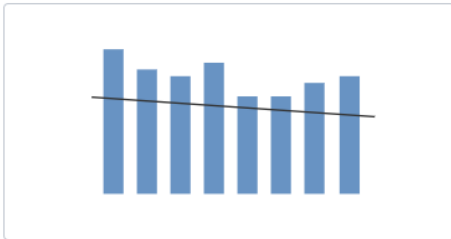
Version Workload Report

Shows the time estimates for all unresolved issues assigned to a version, broken down by user and issues. This helps you understand the remaining work for the version.

Jira Software (14) - Izveštaji

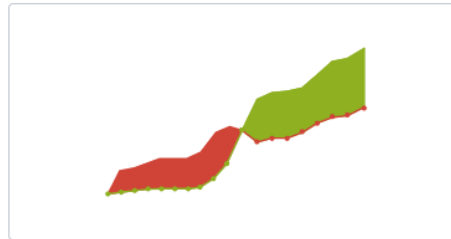
- Analiza prema stavkama

Issue analysis



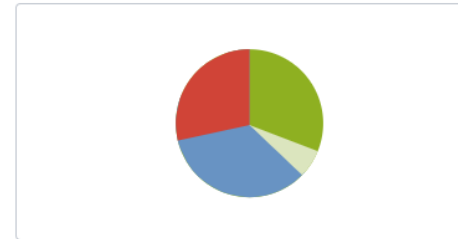
Average Age Report

Shows the average age of unresolved issues for a project or filter. This helps you see whether your backlog is being kept up to date.



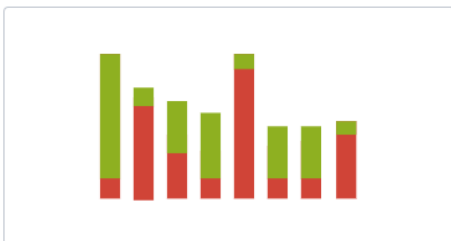
Created vs. Resolved Issues Report

Maps created issues versus resolved issues over a period of time. This can help you understand whether your overall backlog is growing or shrinking.



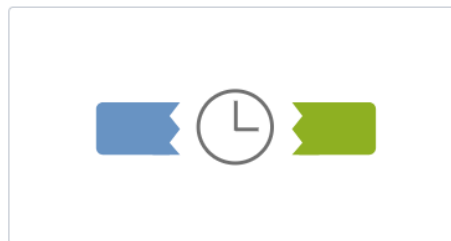
Pie Chart Report

Shows a pie chart of issues for a project/filter grouped by a specified field. This helps you see the breakdown of a set of issues, at a glance.



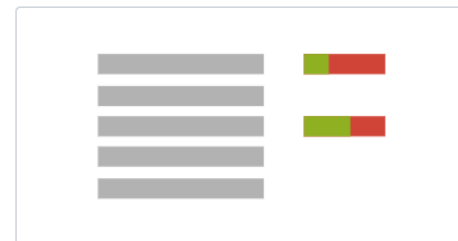
Recently Created Issues Report

Shows the number of issues created over a period of time for a project/filter, and how many were resolved. This helps you understand if your team is keeping up with incoming work.



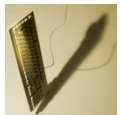
Resolution Time Report

Shows the length of time taken to resolve a set of issues for a project/filter. This helps you identify trends and incidents that you can investigate further.



Single Level Group By Report

Shows issues grouped by a particular field for a filter. This helps you group search results by a field and see the overall status of each group.



Zaključak

- Koriste se i Scrum i Kanban aktivno u softverskoj industriji, kada govorimo o agilnim projektima
- Za oba pristupa moraju da se poštuju pravila!
- Alati neophodni za upravljanje i dobro praćenje projekta i svih zadataka na projektu



HVALA NA PAŽNJI!

E-mail: drazen.draskovic@etf.bg.ac.rs

