

# **Agilne metodologije**

# Agilne metodologije

[Agilni manifest 2001]

- **Za njih pojedinci i interakcije imaju veću vrednost nego procesi i alati.** Ova filozofija podrazumeva da je projektnom timu potrebno obezbediti neophodne resurse i imati poverenje u to da će oni svoje poslove uraditi dobro. Timovi se samoorganizuju i komuniciraju direktno licem u lice, umesto posredstvom dokumentacije.
- **Oni više vole da ulože vreme u izradu softvera koji radi, nego u izradu sveobuhvatne dokumentacije.** To jest, primarno merilo uspeha je stepen do koga softver radi ispravno.
- **Oni se usredsređuju na zajednički rad sa naručiocern, umesto na proces ugovaranja i pregovaranja,** čime se naručilac uključuje u ključne aspekte razvojnog procesa.
- **Oni se usredsreduju da odgovore na promene, umesto na planiranje i praćenje plana,** jer veruju da je nemoguće da se svi zahtevi predvide na početku razvojnog procesa.

# Agilne metodologije

- ▶ Predstavnici agilnih metodologija razvoja:
  - Ekstremno programiranje (XP)
  - Scrum

# Ekstremno programiranje (XP)

## Osnovni koncepti:

- ▶ **Igra planiranja:** Naručilac definiše svaki zahtev prema vrednosti koju njegova implementacija donosi sistemu. Korisnici pišu scenarije o tome kako bi sistem trebalo da radi, a projektni tim procenjuje neophodne resurse za realizaciju tih scenarija. Nakon pisanja scenarija, potencijalni korisnici dodeljuju prioritete zahtevima, razdvajaju ih i spajaju, sve dok se ne postigne konsenzus o tome šta je stvarno potrebno, šta može da se testira. kao i šta može da se uradi sa raspoloživim resursima. Planeri onda generišu mapu svake verzije, dokumentujući šta ta verzija sadrži i kada će biti isporučena.
- ▶ **Male verzije:** Sistem je projektovao tako da funkcionalnost može da se isporuči što je pre moguće. Funkcije su dekomponovane na male delove, sa ciljem da se neka funkcionalnost isporuči u ranoj fazi, a zatim poboljšava ili proširuje u kasnijim verzijama. Male verzije zahtevaju fazni pristup razvoju, sa inkrementalnim ili iterativnim ciklusima.
- ▶ **Metafora:** Projektni tim se usaglašava oko zajedničke vizije načina na koji će sistem raditi. Kao podršku zajedničkoj viziji tim bira zajedničku nomenklaturu i sporazumeva se oko zajedničkog načina tretiranja ključnih pitanja.

# Ekstremno programiranje

- ▶ **Jednostavan dizajn:** Jednostavnost dizajna se održava tako što se tretiraju jedino aktuelne potrebe. Ovaj pristup se zasniva na gledištu da predviđanje potencijalnih budućih potreba dovodi do nepotrebnih funkcija.
- ▶ **Pisanje testova pre kodiranja:** Da bi se osiguralo da potrebe naručioca budu glavni pokretač razvoja, prvo se pišu scenariji testova, kao način kojim se naručilac primorava da specifikuje zahteve koji se, po izradi softvera, mogu testirati i verifikovati. U XP-u se koriste dve vrste testova. Prva vrsta su funkcionalni testovi koje specinkujenaručilac, a koje sprovode projektni tim i korisnici. Druga vrsta su testovi delova koje piše i izvršava razvojni tim. U XP-u su funkcionalni testovi automatizovani, a u idealnom slučaju oni se izvršavaju svakog dana. Funkcionalni testovi se smatraju delom specifikacije sistema. Testovi delova se pišu i pre i posle kodiranja, radi verifikovanja da svaki modul implementacije radi prema očekivanjima.
- ▶ **Refaktorisanje (prerađivanje koda):** Tokom izgradnje sistema, postoji verovatnoća da dode do promene zahteva. Kako je glavna karakteristika XP filozofije dizajn jedino aktuelnih zahteva, pojava novih zahteva često primorava projektni tim da preispitaju postojeća projektna rešenja. Refaktorisanje (prerađivanje koda) se odnosi na preispitivanje zahteva i dizajna, i njihovo preformulisanje u skladu sa novim i postojećim potrebama. Nekada se refaktorisanje bavi prestrukturisanjem koda, bez izmena spoljašnjeg ponašanja sistema. Refaktorisanje se vrši u malim koracima, uz testiranje delova i programiranje u paru, sa jednostavnošću kao idejom vodiljom.

# Ekstremno programiranje

- ▶ **Programiranje u paru:** Koristeći jednu tastaturu, dva programera koji rade u paru razvijaju sistem na osnovu specifikacija i dizajna. Jedna osoba je odgovorna za kodiranje, no rad u paru poseduje fleksibilnost, jer jedan programer može da ima više partnera u istom danu.
- ▶ **Kolektivna svojina:** U XP-u svaki učesnik u razvoju može da izmeni bilo koji deo sistema, dok je on u fazi razvoja.
- ▶ **Neprekidna integracija:** Brza isporuka funkcionalnosti znači da isporuka sistema koji funkcionišu može da se obeća naručiocu u roku od jednog dana ili čak jednog sata. Naglasak je na malim inkrementirna ili poboljšanjima, umesto na velikim skokovima od jedne revizije do druge.

# Ekstremno programiranje

- ▶ **Održiv korak:** Naglasak na ljudima u XP-u uzima u obzir i činjenicu da umor može proizvesti greške. Tako pobornici XP-a sugerišu da je cilj 40 radnih časova nedeljno. Ako projektni tim treba da utroši ogromno vreme da bi sustigao rokove, to je siguran znak da su rokovi nerealni ili da postoji nedostatak resursa neophodnih za njihovo poštovanje.
- ▶ **Naručilac raspoloživ na terenu:** U idealnom slučaju, naručilac bi trebalo da bude raspoloživ na terenu, tj. trebalo bi da radi sa razvojnim timom na definisanju zahteva i obezbeđivanju povratnih informacija o tome kako treba da se vrši testiranje.
- ▶ **Standardi kodiranja:** XP zastupa jasno definisanje standarda kodiranja, sa ciljem da se članovi tima osposobe da razumeju neophodne izmene u produktima rada drugih članova tima. Rezultat treba da bude kod koji izgleda kao da ga je pisala jedna osoba, konsistentan sa aspekta pristupa i izražavanja.

**Scrum**



# Scrum

- ▶ **Scrum** je okvir za razvoj i održavanje složenih (softverskih) proizvoda.
- ▶ Scrum se bazira na ideji empirijske kontrole procesa ili empirizma. Empirizam tvrdi da znanje dolazi iz iskustva i donošenja odluka na osnovu onoga što je poznato
- ▶ Scrum koristi iterativni, inkrementalni pristup sa ciljem povećanja izvesnosti i upravljanja rizicima.
- ▶ Scrum okvir čine Scrum timovi i odgovarajuće uloge, događaji, artefakti i pravila.

# Opis Scruma na jednoj strani

## Roles



Product Owner:  
Set priorities



ScrumMaster:  
Manage process,  
remove blocks

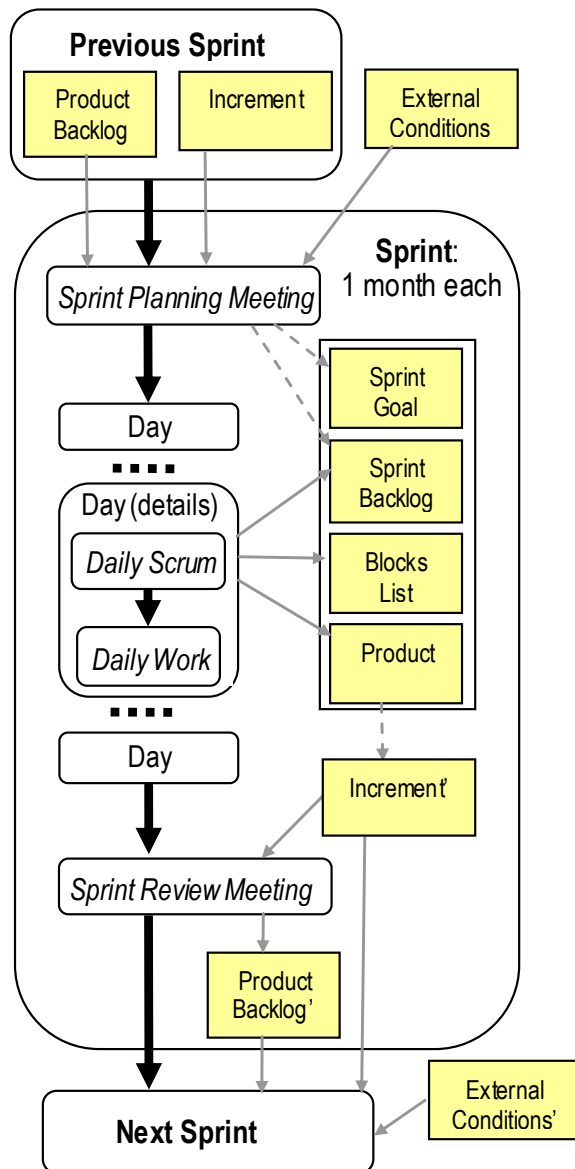


Team Members:  
Develop product



Stakeholders:  
observe & advise

## Sprint Process



## Key Meetings

### **Sprint Review Meeting**

- ScrumMaster hosts; all attend. ½ day
- Informal, informational
- Agenda:
  1. Demo and discuss Increment
  2. Hold retrospective
  3. Announce next Sprint Review Meeting

### **Sprint Planning Meeting**

- ScrumMaster hosts; all attend. ½ day
- In: Product Backlog, latest Increment, business and technology conditions
- Creates: Sprint Goal, Sprint Backlog
- Agenda:
  1. Product Owner presents candidate items
  2. Team estimates items and budget
  3. Team selects highest priority items from Product Backlog
  4. Product Owner declares Sprint Goal
  5. Team turns items into new Sprint Backlog

### **Daily Scrum**

- Hosted by ScrumMaster; all attend; Stakeholders observe
- 15 minutes; same time every day
- Each speaker answers:
  1. What did you do yesterday?
  2. What will you do today?
  3. What's in your way?
- Team updates Sprint Backlog; ScrumMaster updates Blocks List

### **Sprint Cancellation (rare)**

ScrumMaster calls, if they feel team cannot meet Sprint Goal.  
Ex.: drastic change in environment; inability to address impediments; severe mis-estimation.

## Key Artifacts

### **Product Backlog**

- List of requirements & issues
- Owned by Product Owner
- Anybody can add to it
- Only Product Owner prioritizes

### **Sprint Goal**

- One-sentence summary
- Declared by Product Owner
- Mutually acceptable to team and Product Owner
- Creative solutions are welcome

### **Sprint Backlog**

- List of tasks
- Owned by team
- Only team modifies it

### **Blocks List**

- Impediments, blocks, and pending decisions
- Owned by ScrumMaster
- Updated daily

### **Increment**

- Version of the product
- Shippable functionality (tested, documented, etc.)

### **Visual Feedback**

- "Information radiators"
- Ex.: Burndown chart, architecture diagram

# Glavne uloge u scrum timu

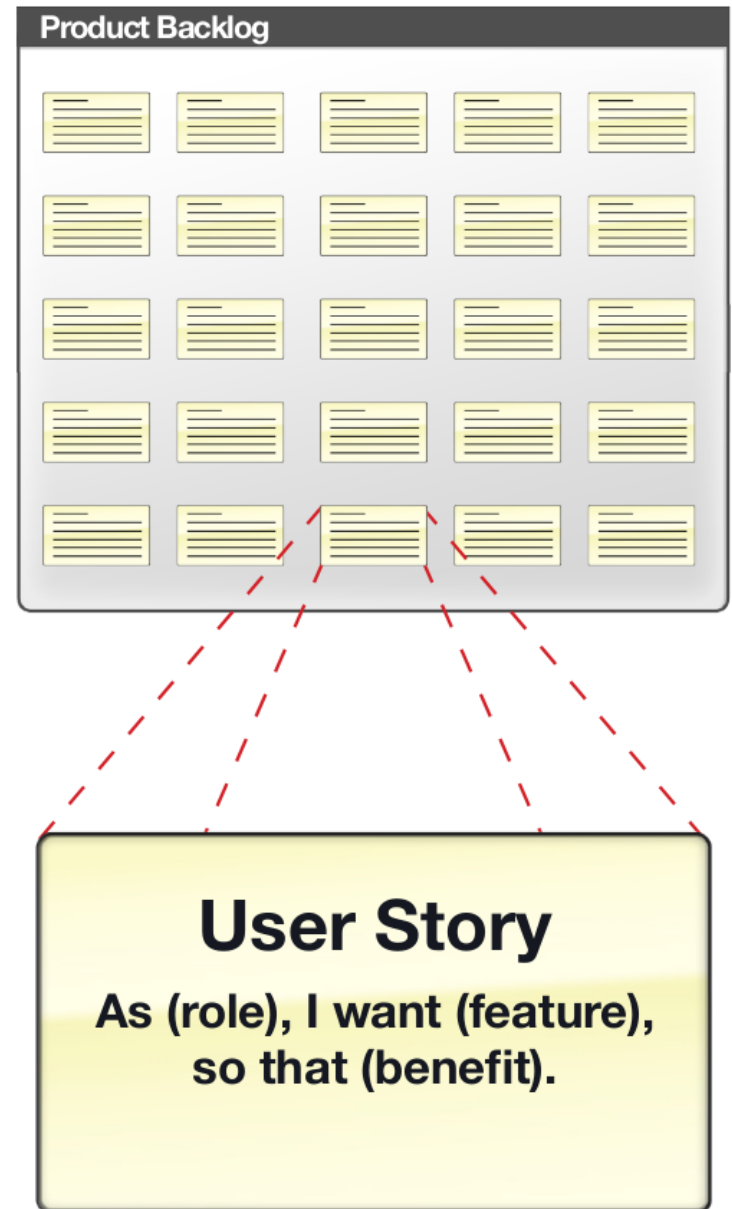
- 1) Scrum Master**, moderator timskog rada; obezbeđuje sve što je potrebno timu, priprema sastanke i nadgleda proces.
- 2) Product Owner**, (vlasnik proizvoda) reprezentuje klijenta i odgovara za sadržaj i prioritete u product backlog-u.
- 3) Razvojni Tim**, grupa ljudi sa različitim specijalnostima (developeri, tester) koji rade konkretne aktivnosti analize, dizajna, implementacije, testiranje itd.

# Glavni artefakti Scruma

- ▶ Product backlog (preostali poslovi proizvoda)
- ▶ Sprint backlog (preostali poslovi u sprintu)
- ▶ Iteration
- ▶ Burndown (grafikon preostalog posla)

# Product backlog (PB)

- ▶ PB je lista svih zahteva, funkcionalnosti, poboljšanja i popravki koje zajedno čine promene koje treba da se implementiraju u budućim inkrementima proizvoda.
- ▶ PB stavke imaju opis, redosled, procenu i (poslovnu) vrednost.
- ▶ Stavke PB-a često su u obliku korisničkih priča (user stories)
- ▶ Za PB odgovoran je Product Owner.



# Sprint backlog (SB)

- ▶ SB je lista poslova koje razvojni tim mora uraditi tokom sledećeg sprinta. Lista je izvedena izborom onoliko stavki sa vrha Product Backloga koliko razvojni tim smatra da ima dovoljno posla da popuni sprint.
- ▶ SB stavke razvojni tim deli na zadatke. Zadaci se nikada ne dodeljuju, ih preuzimaju pojedini članovi tima u skladu sa postavljenim prioritetima i veštinama članova razvojnog tima. Ovo promovise samoorganizovanje razvojnog tima.

# Sprint backlog

- Često se za predstavljanje taskova i njihovog statusa (to do, in progress, done,...) koristi task board.

Story	To Do	In Process	To Verify	Done
<div data-bbox="244 658 430 772">As a user, I... 8 points</div>	<div data-bbox="478 658 665 772">Code the... 9</div> <div data-bbox="679 658 866 772">Test the... 8</div> <div data-bbox="478 782 665 896">Code the... 2</div> <div data-bbox="679 782 866 896">Code the... 8</div> <div data-bbox="478 906 665 1021">Test the... 8</div> <div data-bbox="679 906 866 1021">Test the... 4</div>	<div data-bbox="913 658 1100 772">Code the... DC 4</div> <div data-bbox="913 782 1100 896">Test the... SC 8</div>	<div data-bbox="1141 658 1328 772">Test the... SC 6</div>	<div data-bbox="1369 658 1556 772">Code the... D</div> <div data-bbox="1369 725 1556 839">Test the... SC 8</div> <div data-bbox="1369 796 1556 911">Test the... SC</div> <div data-bbox="1369 868 1556 982">Test the... SC</div> <div data-bbox="1369 911 1556 1025">Test the... SC 6</div>
<div data-bbox="244 1053 430 1168">As a user, I... 5 points</div>	<div data-bbox="478 1053 665 1168">Code the... 8</div> <div data-bbox="679 1053 866 1168">Test the... 8</div> <div data-bbox="478 1178 665 1292">Code the... 4</div> <div data-bbox="679 1178 866 1292">Code the... 6</div>	<div data-bbox="913 1053 1100 1168">Code the... DC 8</div>		<div data-bbox="1369 1053 1556 1168">Test the... SC</div> <div data-bbox="1369 1110 1556 1225">Test the... SC</div> <div data-bbox="1369 1168 1556 1282">Test the... SC 6</div>

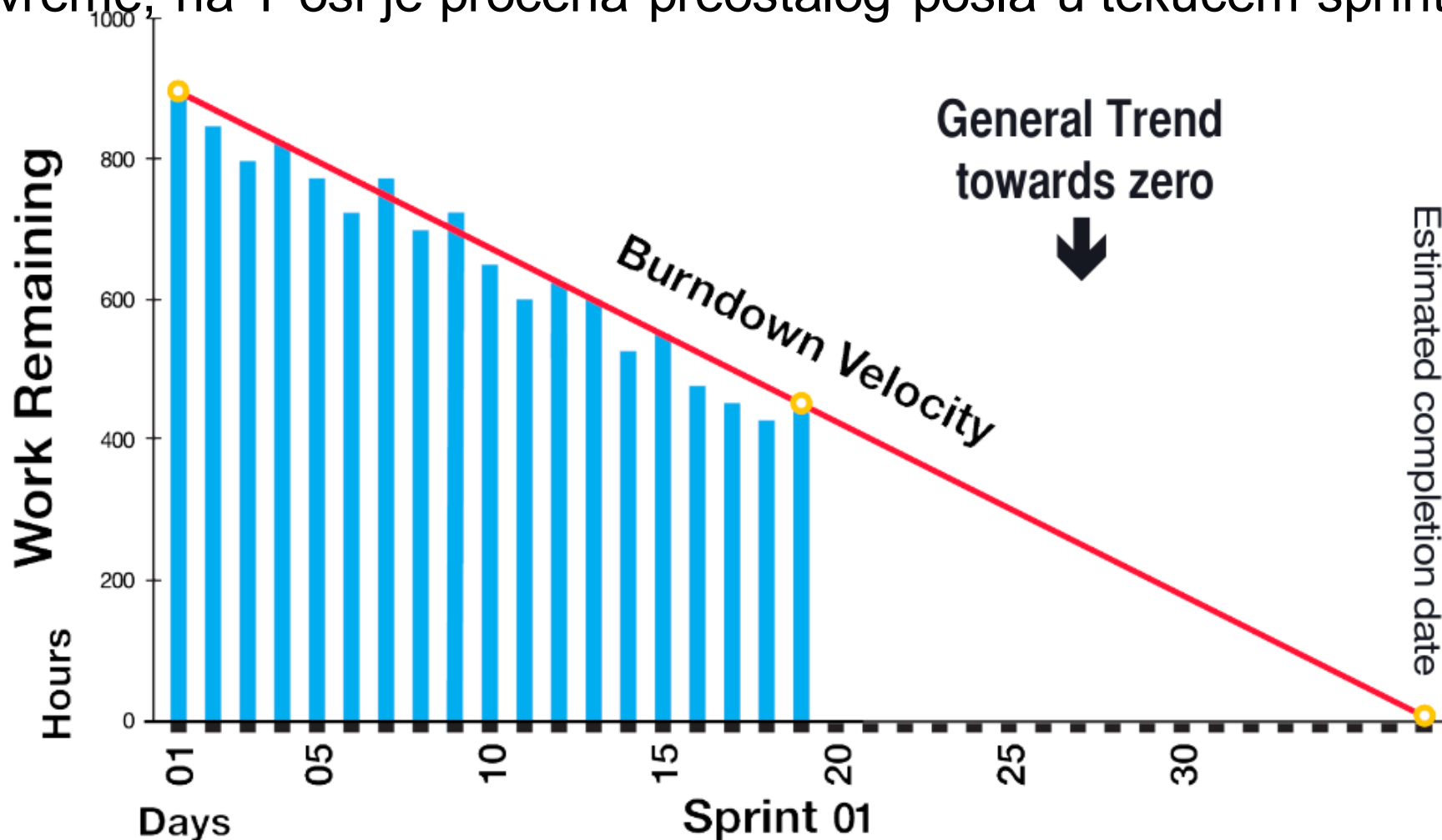
# Inkrement

- ▶ Inkrement je skup svih Product Backlog stavki koje su završene tokom tekućeg i svih prethodnih sprintova.
- ▶ Na kraju sprinta, novi inkrement mora biti završen što znači da mora biti u upotrebljivom stanju i da ispunjava definiciju završenoga (engl. Definition of Done) tog Scrum tima.
- ▶ Inkrement mora biti upotrebljiv bez obzira na to da li će ga Product Owner pustiti u upotrebu ili ne.



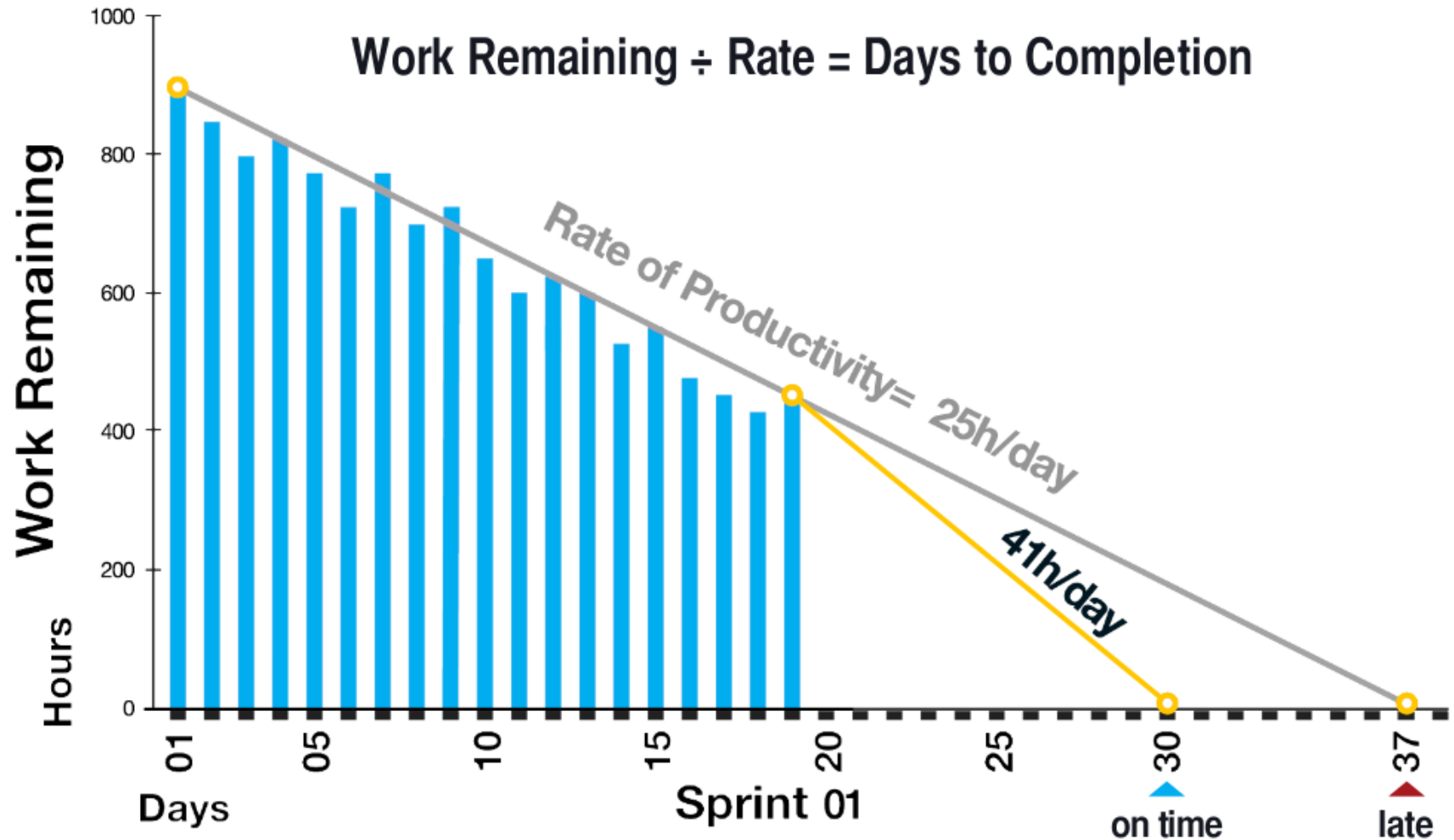
# Grafikon preostalog posla (burndown)

- ▶ Napredak tima se prati korišćenjem burndown grafikona
- ▶ Ovaj grafikon predstavlja preostali posao u vremenu. Na x osi je vreme, na Y osi je procena preostalog posla u tekućem sprintu.



# Velocity (Brzina)

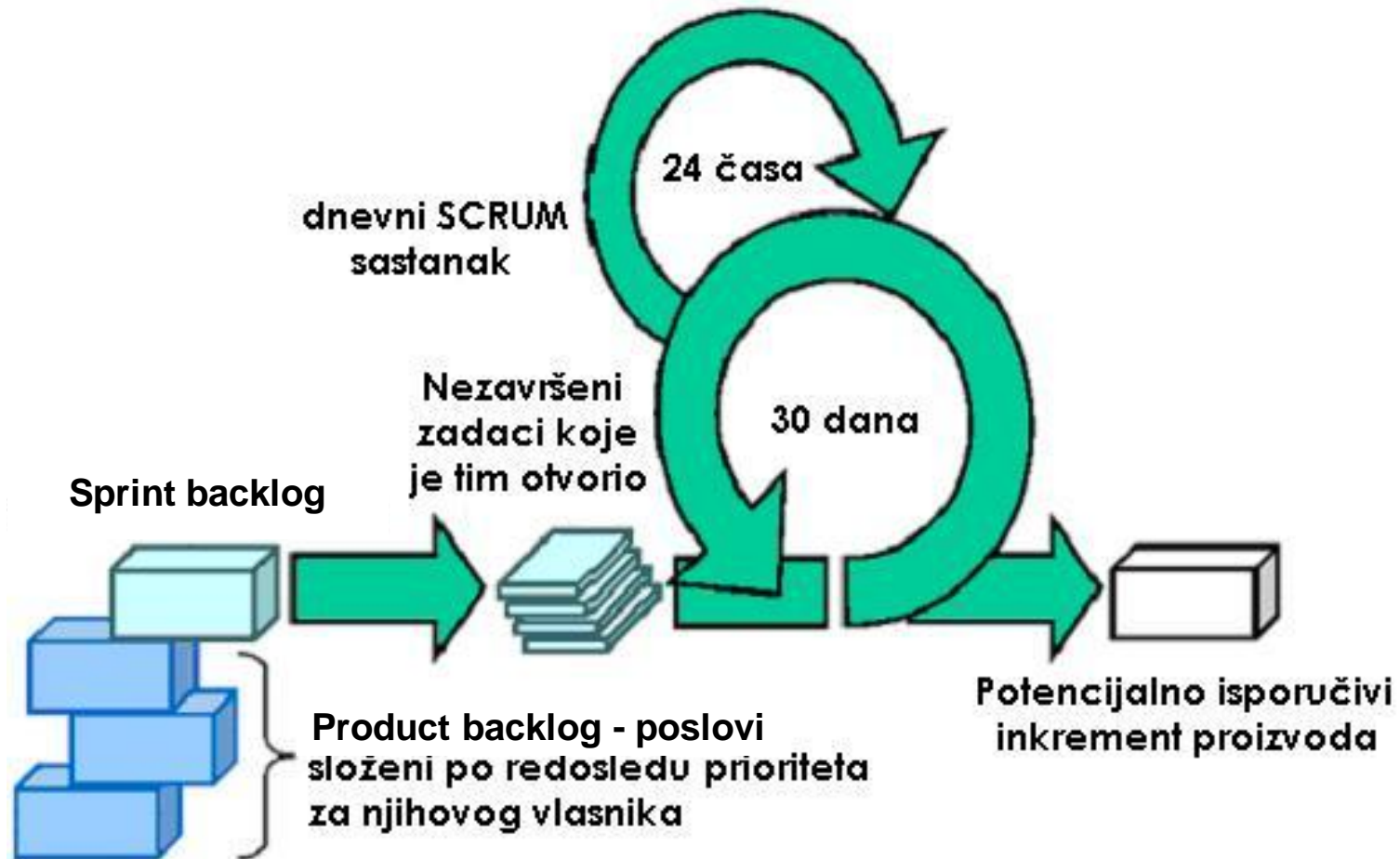
- ▶ Nagib Burndown grafa, naziva se brzina i predstavlja prosečnu produktivnost po danu
- ▶ Informacija da li projekat ide po planu rano u toku sprinta, može pomoći da se naprave doterivanja koja će omogućiti uklapanje u raspored.



# Sprint

- ▶ Centralni koncept Scrum-a je Sprint, vremenski ograničen na mesec dana ili kraći vremenski period, tokom kojeg se stvara upotrebljiv, potencijalno isporučiv inkrement proizvoda.
- ▶ Svaki Sprint uobičajno ima **konstantno trajanje** tokom procesa razvoja (timeboxed). Novi Sprint počinje odmah po zaključivanju prethodnog.
- ▶ Sprint se sastoji od sledećih događaja i aktivnosti:
  - Sprint Planning (Sastanak planiranja Sprinta),
  - Daily Scrum (Dnevni Scrum ili brifing),
  - razvoj proizvoda,
  - Sprint Review (Pregled Sprinta) i
  - Sprint Retrospective (Osvrt na Sprint).

# Scrum proces



[www.methodsandtools.com/archive/archive.php?id=18](http://www.methodsandtools.com/archive/archive.php?id=18)

# Dnevni Scrum (15 minutni sastanak)

- ▶ Šta ste uradili od juče?
- ▶ Šta planirate za danas da uradite?
- ▶ Ima li problema koji vas sprečavaju da ostvarite plan?

# Sastanak planiranja sprinta

- ▶ Planira se sledećih 15-30 dana
- ▶ Biraju se poslovi koji će da se rade (tj. šta se može isporučiti sa inkrementom koji je rezultat ovog sprinta)
- ▶ Priprema se Sprint Backlog koji elaborira vreme koje će trebati da se urade ti poslovi, učestvuje ceo tim
- ▶ Limit sastanka je 8 sati

# Pregledni sastanak sprinta (Sprint Review Meeting)

- Pregled šta je od posla završeno a šta nije (korigovati Product backlog po potrebi)
- Prezentacija završenog posla zainteresovanim stranama ("demo")
- Limitirano trajanje na 4 sata

# Retrospektiva sprinta (Osvrt na Sprint)

- Osvrt na sprint se održava nakon svakog Sprint Review sastanka, a pre narednog planiranja Sprinta
- Svi članovi tima daju osvrt na protekli sprint.
- Svrha je poboljšanje procesa u smislu odnosa ljudi, procesa, alata.
- Dva glavna pitanja tokom osvrta: Šta se dobro odvijalo tokom sprinta? Šta može da se poboljša u sledećem sprintu?
- Limit od 3 sata