Softversko inženjerstvo

Agilne metodologije za razvoj softvera

dr Miloš Stojanović*

Visoka tehnička škola strukovnih studija Niš

2017.

*milos.stojanovic@vtsnis.edu.rs

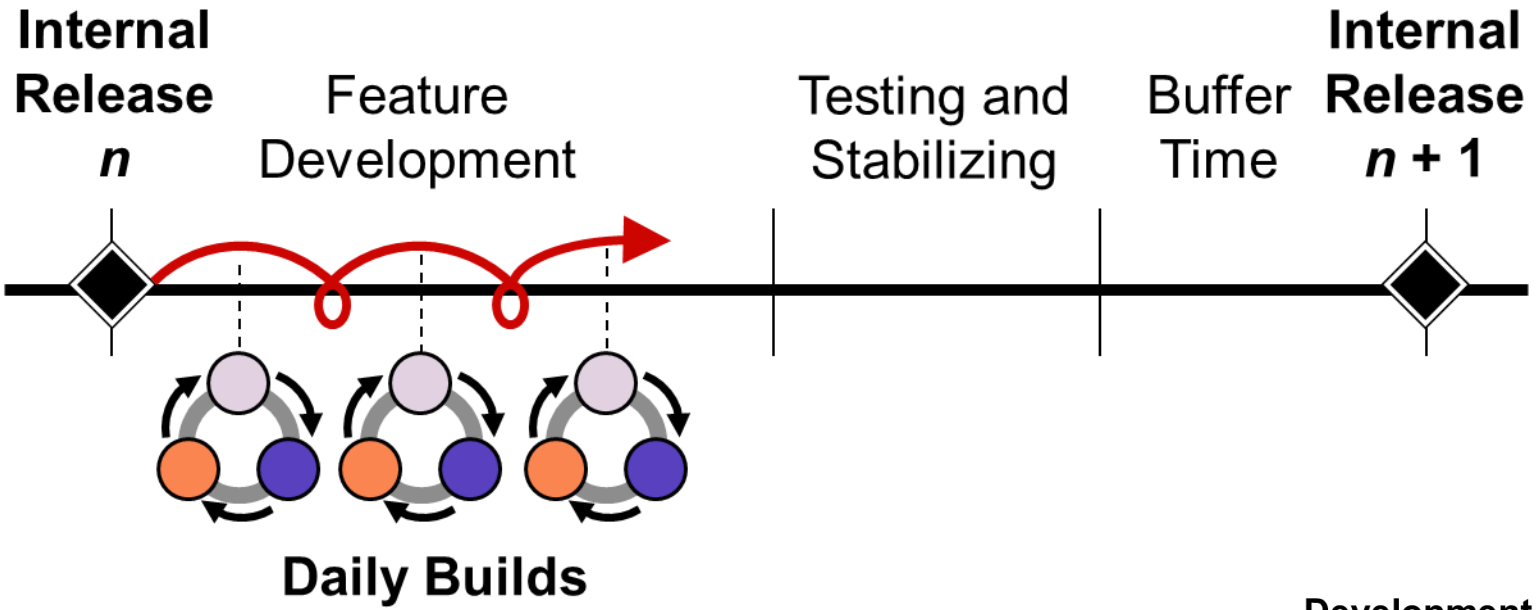
Agilni razvoj softvera

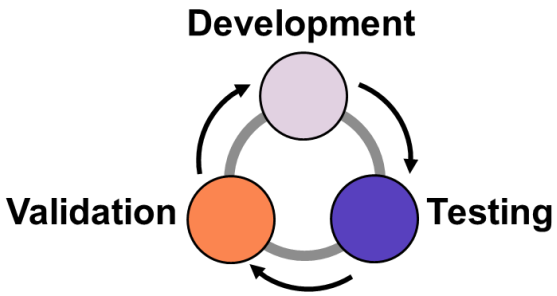
- Agilno = aktivnost, hitnost, spremnost za pokret
- I iterativni i inkrementalni razvoj
- Inkrementalne metode 3 - 6 meseci za razvoj
- Agilne metode 1 - 4 nedelje za razvoj

Agilne metodologije

- Uključuju **iteracije** sa kratkim vremenskim intervalima.
- Razdvajaju zadatke na male **inkremente** sa minimalnim planiranjem.
- Mali timovi rade zajedno da definišu brze prototipove, dokaze koncepata i druga vizuelna sredstva za opisivanje problema koji se rešava.
- **Tim** definiše zahteve za iteraciju, razvija kod, definiše i izvršava integrisane testove a **korisnici** verifikuju rezultate.
- Verifikacija se izvršava mnogo ranije u razvojnom procesu nego kod Waterfall modela.

Agilni razvoj

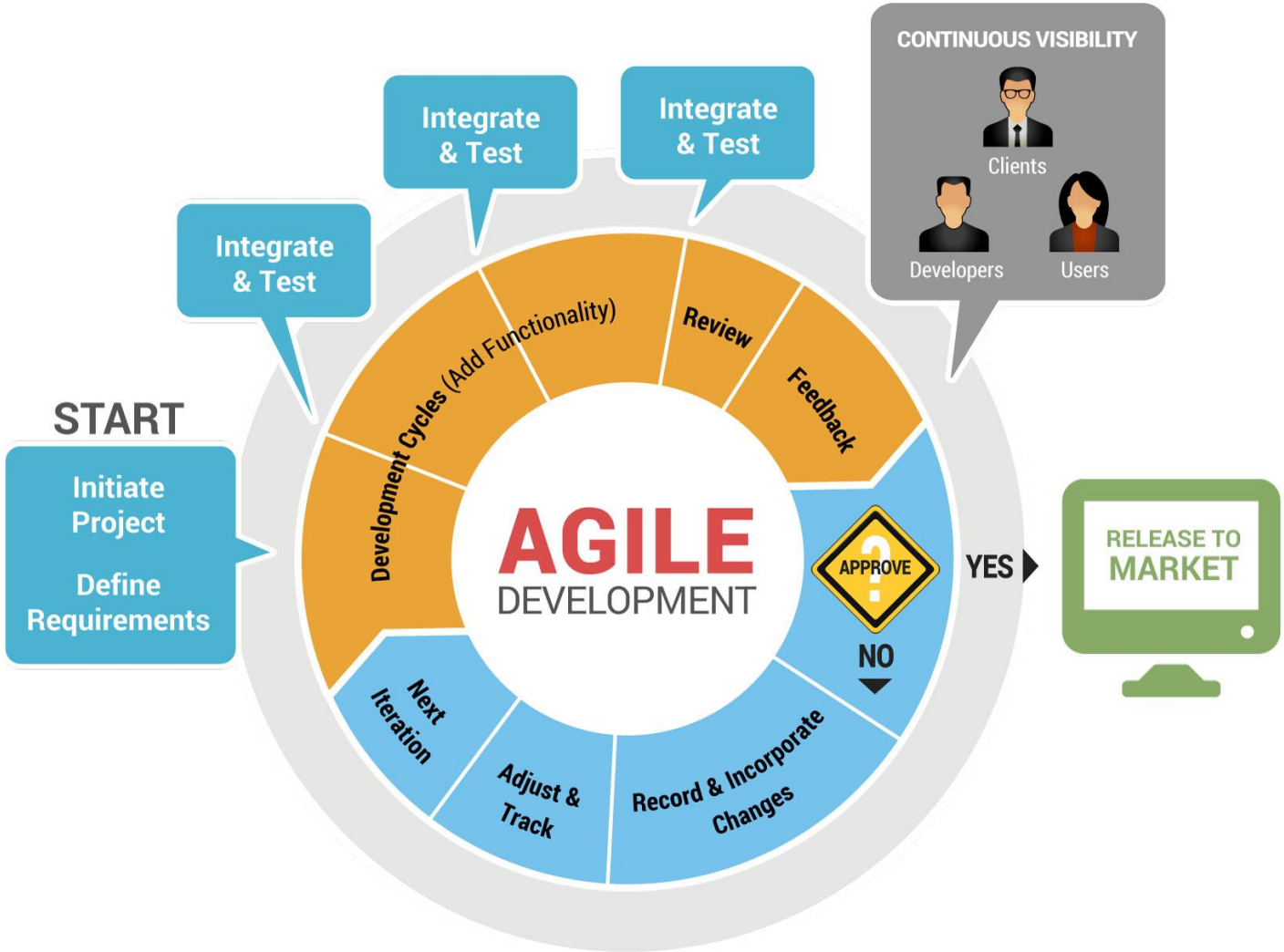




“Agile Manifesto”
Definicija agilnog programiranja

- **Pojedinci i interakcije** između njih su važnije od procesa i alata.
- **Softver koji radi** je važniji od dokumentacije koja raste.
- **Saradnja sa korisnicima** je važnija od procesa ugovaranja.
- **Odziv na promene** je važniji od striktnog pridržavanja i praćenja plana.

Agilni razvoj



Principi

- Zadovoljstvo korisnika brzom isporukom korisnog softvera
- Mogućnost promene zahteva, čak i u poodmakloj fazi razvoja
- Česta isporuka softvera, u razmaku od par nedelja
- Ispravan softver je osnovna mera napretka
- Razvoj koji je u stanju da održi konstantan tempo
- Bliska saradnja između projektanata i poslovnih saradnika

Principi

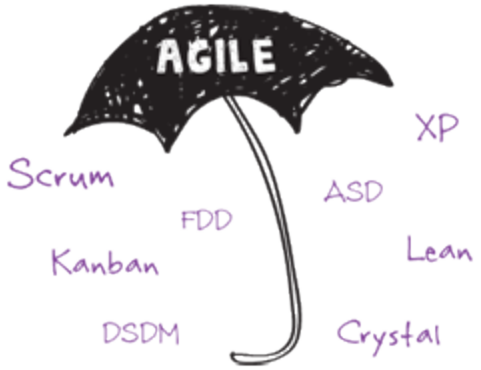
- Najbolji tip komunikacije je komunikacija “licem-ulice”
- Projekti se izvode u okruženju u kojem su motivisani pojedinci, u koje se može imati poverenja
- Kontinualno usmeravanje pažnje ka tehničkoj veštini i dobrom dizajnu
- Jednostavnost
- Samoorganizovani timovi
- Prilagođavanje promenljivim okolnostima

Praksa nije uvek idilična

- Kupac ne može uvek biti umešan u razvoj
- Osobine članova tima
- Različiti prioriteti
- Jednostavnost zahteva dodatni trud
- Stare navike kompanija

Agilne metode

- Ekstremno programiranje (XP)
- Scrum
- Crystal
- Adaptivni razvoj softvera (ASD)
- Feature driven development (FDD)
- Metode dinamičkog razvoja sistema (DSDM)
- Lean Development (LD)
- Agilno modelovanje (AM)



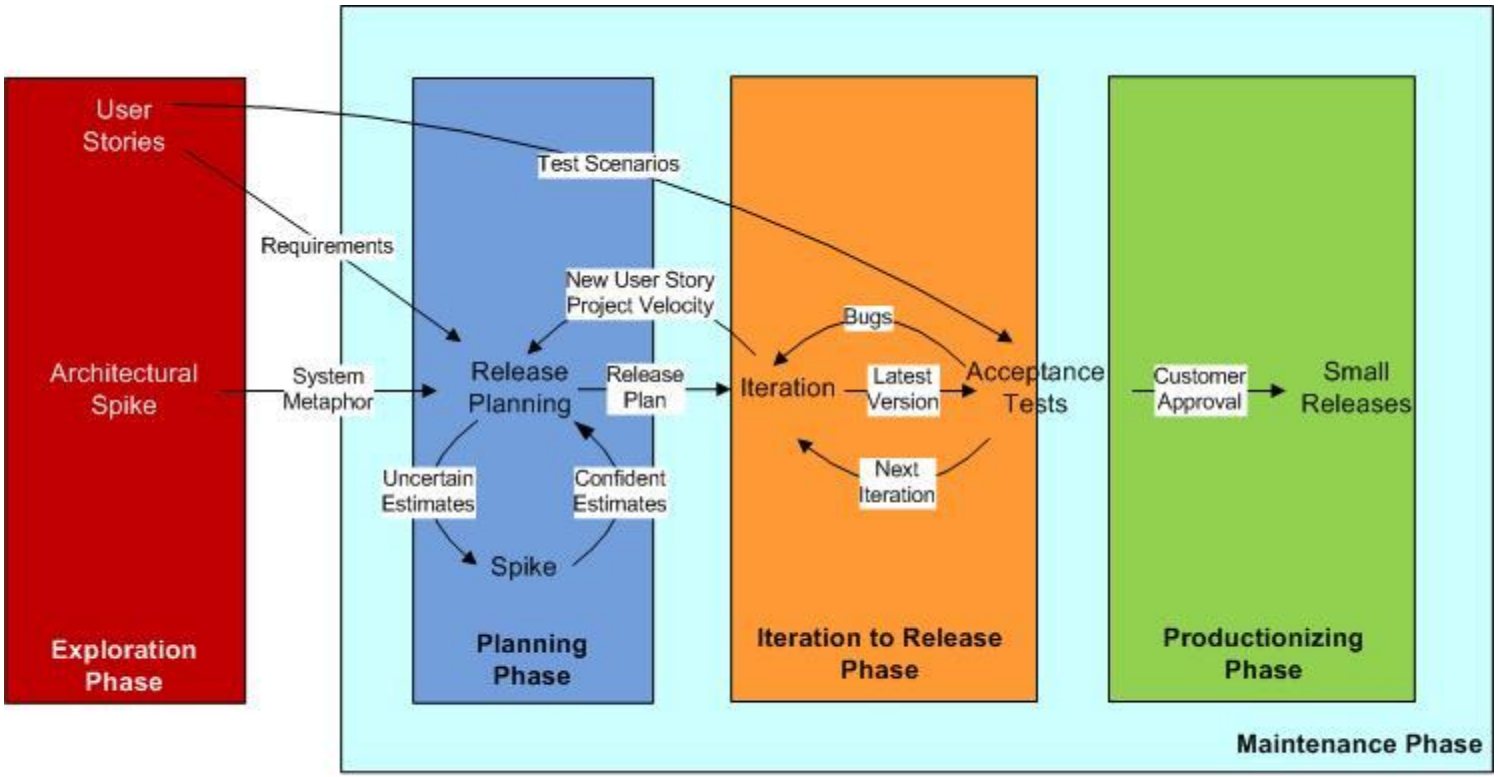
Extreme Programming (XP)

- **Najpopularnija** agilna metoda, nastala 1999. god
- **Mali timovi**
- Proizvodnja dugotrajnog softvera
- **Sposobnost reagovanja na promene zahteva**

Životni ciklus XP-a

1. Istraživanje (pisanje priča - zahteva, upoznavanje tehnologija)
2. Planiranje (utvrđivanje potrebnog vremena, prioriteti, stvaranje)
3. Iteracije (pisanje, procena i dodeljivanje prioriteta novim zadacima, testiranje na kraju svake iteracije)
4. Proizvodnja (testiranje pre isporuke, evidentiranje novih zahteva)
5. Održavanje (dodavanje novih funkcionalnosti, ispravke grešaka)
6. Kraj (zahtevi ispunjeni, završetak projekta)

Životni ciklus XP-a



Extreme Programming (XP) lifecycle

Vrednosti XP-a

- Metodologija se temelji na pet osnovnih vrednosti:

➢ Komunikacija

➢ Jednostavnost

➢ Povratna sprega

➢ Hrabrost

➢ Poštovanje

Aktivnosti u XP-u

- Da bi se dostigle vrednosti ekstremnog programiranja sprovode se četiri vrste aktivnosti pomoću kojih se te vrednosti implementiraju u ponašanje i rad tima:

➢ Kodiranje

➢ Slušanje

➢ Testiranje

➢ Projektovanje

Uloge u XP-u

- Menadžer - formira tim i upravlja njime
- Trener - uči članove tima o XP metodologiji
- Tragač - prikuplja korisničke zahteve i prati napredak testova
- Programer - piše testove, projektuje i kodira
- Tester – pomaže kupcu da piše i razvija testove
- Kupac - piše zahteve i testove, te bira zahteve koji će se raditi u određenoj iteraciji.

Uslovi u XP-u

- Otvorena radna sredina
- Oglasna tabla
- Normalno radno vreme
- Dnevni sastanci
- Nedeljni sastanci
- Tromesečni sastanci

SCRUM

- Polovina 90-ih
- Agilno **upravljanje**, ne **projektovanje** softvera
- Propisuje načine **upravljanja** zahtevima, formiranja iteracija, kontrole implementacije i isporuke klijentu
- Projektni okvir
- Termin preuzet iz ragbija

Uloge u SCRUM-u

- 5 do 10 članova tima:
- Vlasnik projekta
- “Scrum Master”
- Scrum tim
- Zainteresovane strane (eng. stakeholders)

Faze u SCRUM-u

- Scrum proces se sastoji od 3 faze:

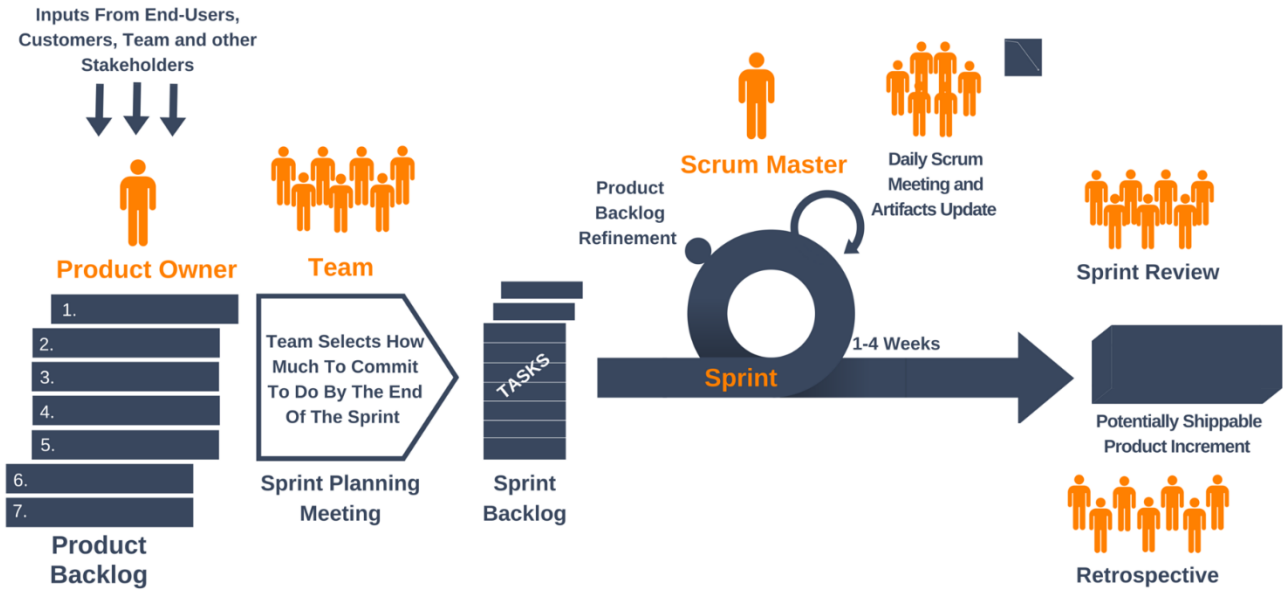
➢ Pre igre: planiranje, projektovanje arhitekture, visok nivo apstrakcije

➢ Igra: razvoj, sprintovi – iterativni ciklusi, poboljšanja, nove verzije

➢ Posle igre: nema novih zahteva, sistem spreman za korišćenje

SCRUM proces

- **Iteracija** - **sprint** traje dve do četiri nedelje
- **Celokupni projekat** (jedna igra) **traje tri do osam sprinteva.**



SCRUM proces

Dnevni scrum

- Svaki član tima odgovara na tri pitanja:
  - Šta je uradio od prethodnog scrum-a
  - Šta mu je u planu da uradi do narednog scrum-a
  - Koji su potencijalni problemi koji mu stoje na putu
- Ciljevi ovog sastanka su:
  - Identifikovanje problema koji usporavaju proces
  - Donošenje zajedničkih odluka, na nivou tima
  - Praćenje napretka tokom sprinta

Analiza sprinta

- Demonstracija funkcionalnosti dodatih tokom sprinta
- Prisutni su svi učesnici razvoja i klijenti
- Proverava se da li su dostignuti planirani ciljevi sprinta
- Diskutuju se uspesi i neuspesi tokom sprinta

SCRUM dokumentacija

- Tri glavna dokumenta koje generiše Scrum ekipa:

➢ zaliha softvera,

➢ zaliha trenutne iteracije i

➢ dijagram preostalog posla u sprintu

Crystal metodologije

- Predložene 1991. god
- Metodologije orijentisane na **ljude** su bolje od metodologija koje su orijentisane na **procese**.
- Sedam ključnih principa Crystal metodologije:

➢ Učestala isporuka

➢ Kontinuirane povratne informacije

➢ Stalna komunikacija

➢ Sigurnost

➢ Fokus

➢ Raspoloživost korisnika

➢ Automatski testovi i integracija

Elementi Crystal Clear metodologije

- **Dokumenti:** plan puštanja u rad, slučajevi korišćenja, skice dizajna, testovi i korisnički priručnik…
- **Uloge** (Tim: 3 - 8 osoba):

➢ sponzor projekta (kupac),

➢ vodeći programer projektant,

➢ programer projektant,

➢ korisnik

- **Proces**

Crystal Clear proces

- Inkrementalna dostava projekta
- Projekat se pušta u rad u manje od dva do tri meseca
- Testovi su automatizovani
- Korisnik je direktno uključen
- Dve korisničke recenzije pre puštanja u rad
- Prati se napredak na projektu

Adaptivni razvoj softvera (ASD)

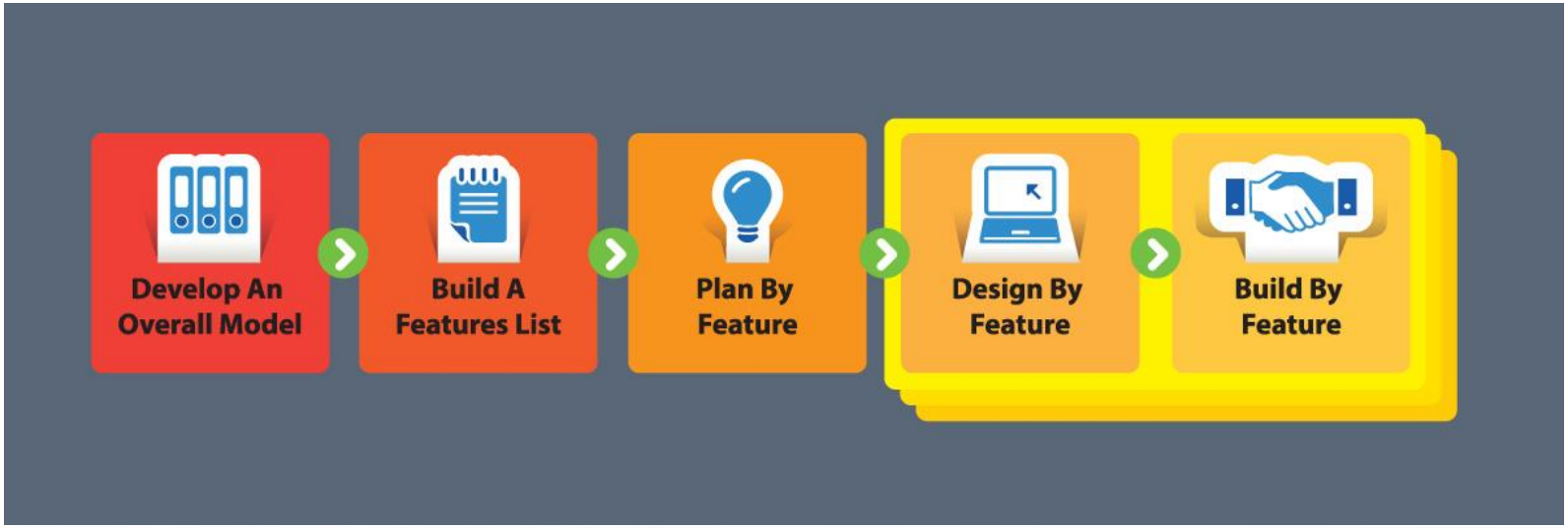
- Namenjena je za razvoj kompleksnih rešenja, uz oslonac na inkrementalni i iterativni razvoj i evolutivne prototipove.
- Fokusira se na misiju
- Zasniva se na funkcionalnostima
- Iterativan
- Vođen je rizikom
- Tolerantan na promene

Feature Driven Development (FDD)

- Razvoj vođen karakteristikama (feature-ima).
- Ne bavi se celokupnim životnim ciklusom, već se **koncentriše na faze projektovanja i konstrukcije.**
- Realizacija malih, zaokruženih funkcija koje se ne kreiraju duže od dve nedelje.

Faze u FDD-u

- Razvoj celokupnog modela
- Realizacija liste potrebnih funkcija
- Planiranje realizacije funkcija
- Detaljno projektovanje
- Konstrukcija potrebnih funkcija



Uloge u FDD-u

- Vođa projekta
- Glavni arhitekta
- Vođa razvoja
- Glavni programer
- Vlasnik klase
- Stručnjak u određenom domenu
- Tim koji razvija određenu funkcionalnost

Metode dinamičkog razvoja sistema (DSDM)

- DSDM je baziran na sledećim principima:

➢ Fokusirati se na potrebe poslovanja

➢ Isporuka na vreme

➢ Kolaboracija

➢ Nikada ne praviti kompromis po pitanju kvaliteta

➢ Inkrementalni i iterativni razvoj

➢ Kontinualna i jasna komunikacija

➢ Demonstracija kontrole

Životni ciklus DSDM projekta

- Studija o izvodljivosti (Feasibility Study)
- Studija poslovnog domena (Business Study)
- Iteracija razvoja funkcionalnog modela(Functional Model Iteration)
- Iteracija projektovanja i izgradnje (Design and Build Iteration)
- Implementacija u realnom okruženju (instaliranje kod korisnika)

Lean Development (LD)

- Metodologija inspirisana sistemom proizvodnje u japanskim fabrikama automobila
- Vreme razvoja u Tojoti je duplo kraće, a produktivnost radnika je 4 puta veća, uz znatno manji procenat grešaka nego u fabrikama u SAD-u
- **LD je skoncentrisan na vrh preduzeća i strategiju upravljanja, ne na detalje razvojnog procesa na niskom nivou.**
- Pošto se LD bavi više filozofijom upravljanja nego konkretnim razvojnim procesom, detalji razvoja nisu precizno definisani, kao ni veličina tima i način njegovog organizovanja

Principi LD-a

- Najveći prioritet je zadovoljiti očekivanja korisnika
- Potrebno je obezbediti najbolje moguće rešenje za novac korisnika
- Uspeh zavisi od aktivnog učešća korisnika
- Svaki LD projekat je rezultat timskog rada
- Sve se može menjati
- Rešenje primenjivo na širi problemski domen, a ne samo za konkretni problem
- Realizacija “unikatnog” rešenja je preskupa

Principi LD-a

- Bolje je prilagoditi gotovo rešenje, nego raditi svaki put iz početka
- Bolje je imati 80% rešenja danas, nego 100% rešenja u budućnosti
- Minimalizam je esencijalan za uspeh projekta
- Potreba određuje tehnologiju, ne obrnuto

Agilno modelovanje (AM)

- Nije metod koji podržava celokupni životni ciklus softvera, već se bavi isključivo tehnikama za realizaciju **modela i dokumentacije** u "agilnom" maniru
- **Može se primenjivati u okviru bilo koje druge metodologije, agilne ili ne**
- Vrednosti se poklapaju sa vrednostima XP-a:

➢ Komunikacija
➢ Jednostavnost
➢ Povratna sprega od strane korisnika
➢ Hrabrost
➢ Jedna dodatna - skromnost

Principi AM-a

- Softver koji zadovoljava potrebe korisnika je primarni cilj
- Priprema za period koji sledi je drugi po važnosti cilj projekta
- Minimalno opterećivati projekat dodatnim artefaktima
- Pretpostaviti da je najjednostavnije rešenje najbolje rešenje
- Prihvatiti da su promene neminovne
- Razvijati sistem inkrementalno
- Modelovati sa svrhom, tj. ne kreirati modele radi njih samih
- Raditi kvalitetno
- Naručiocu pružiti najbolje za njegov novac

Dodatni materjal

- https://www.versionone.com/agile-101/agile-methodologies/
- https://www.youtube.com/watch?v=WxiuE-1ujCM
- https://www.scrumalliance.org/
- http://www.cs.ccsu.edu/~stan/classes/CS410/CS410-FA16.html