Tradicionalne i agilne metode razvoja softvera

Ivan Boban, dr.sc.

Sadržaj

- Životni ciklus razvoja softvera
- Tradicionalne metode
- Agilne metode

Životni ciklus razvoja softvera

- Životni ciklus razvoja softvera (SDLC – System Development Life Cycle) opisuje „životni vijek“ od nastanka potrebe za softverom, preko implementacije, održavanja, pa sve dok se ne prestane koristiti.

- Model procesa razvoja softvera definira redoslijed razvojnih faza.

- Postoji više varijanti SDLC-a, a to su:
  - ✓ Tradicionalni linearni (sekvencijalni) “vodopadni” (eng. “waterfall”)
  - ✓ Paralelni model,
  - ✓ Iterativni model ili evolucijski model,
  - ✓ Spiralni model,
  - ✓ Agilne metode itd.

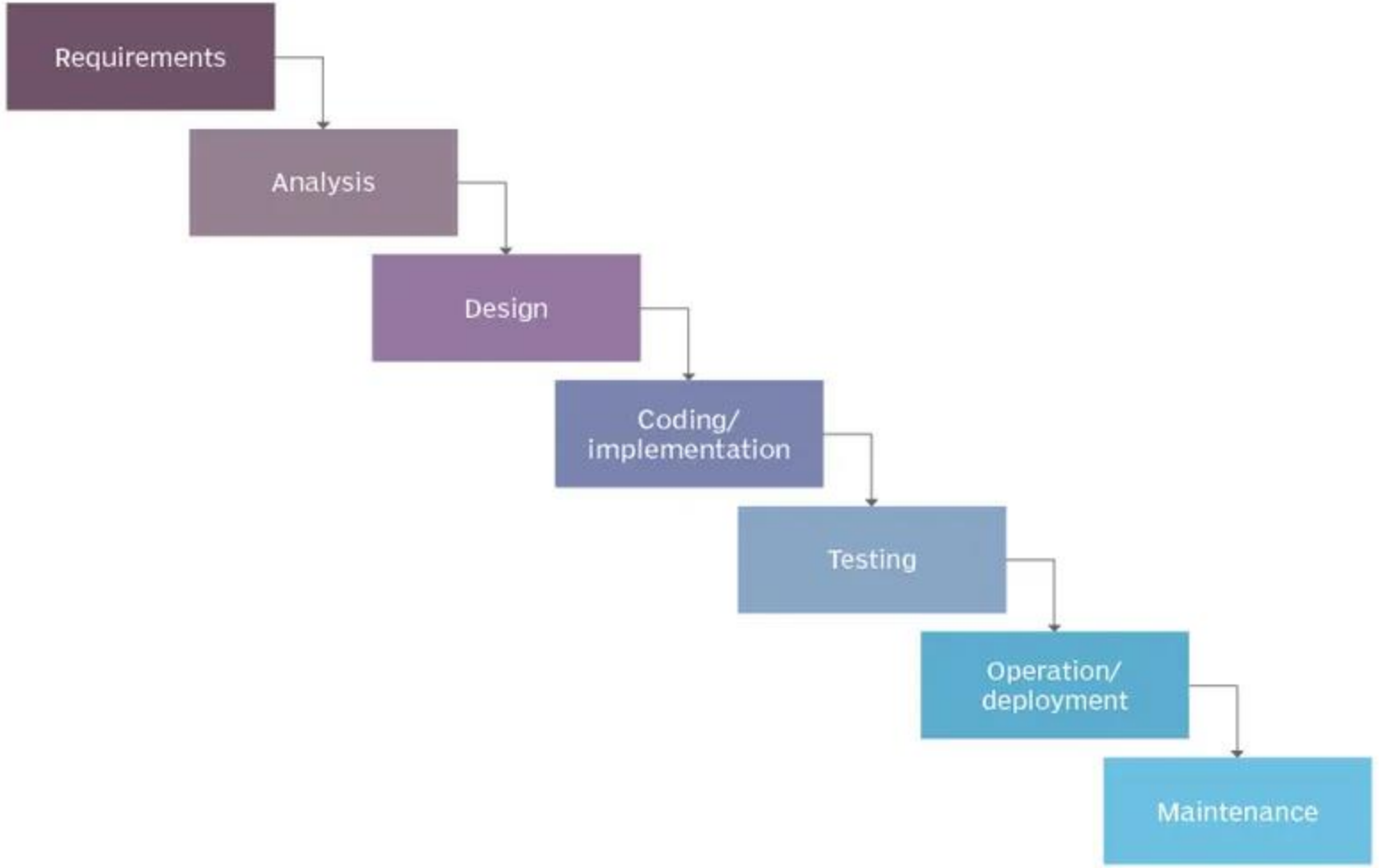
Vodopadni model

- Model vodopada poznat je i kao model životnog ciklusa razvoja softvera.
- Bio je to prvi procesni model uveden kao linearno-sekvencijalni model životnog ciklusa.
- Model vodopada jednostavno govori fenomenu da se jedna faza u potpunosti završi prije započinjanja nove faze razvoja, tako da ne postoji preklapanje faza.

Faktori vodopadnog modela

- ✓ U prijavi nema dvosmislenih zahtjeva.
- ✓ Stabilnost definicije proizvoda
- ✓ Nije dinamičan
- ✓ Na raspolaganju su veliki resursi s odgovarajućom stručnošću kako bi podržali proizvod
- ✓ Projekt kratke duljine
- ✓ Dobar dokument, jasni i fiksni zahtjevi.

Waterfall model



Faktori vodopadnog modela

- Pri korištenju sekvencijalnih metodologija razvoja analitičar sistema i članovi razvojnog tima napreduju sekvencijalno od jedne do druge faze projekta. Svaka faza se u cijelosti završi i potom se prelazi na sljedeću fazu. Nije uobičajeno vraćati se na prethodnu fazu.

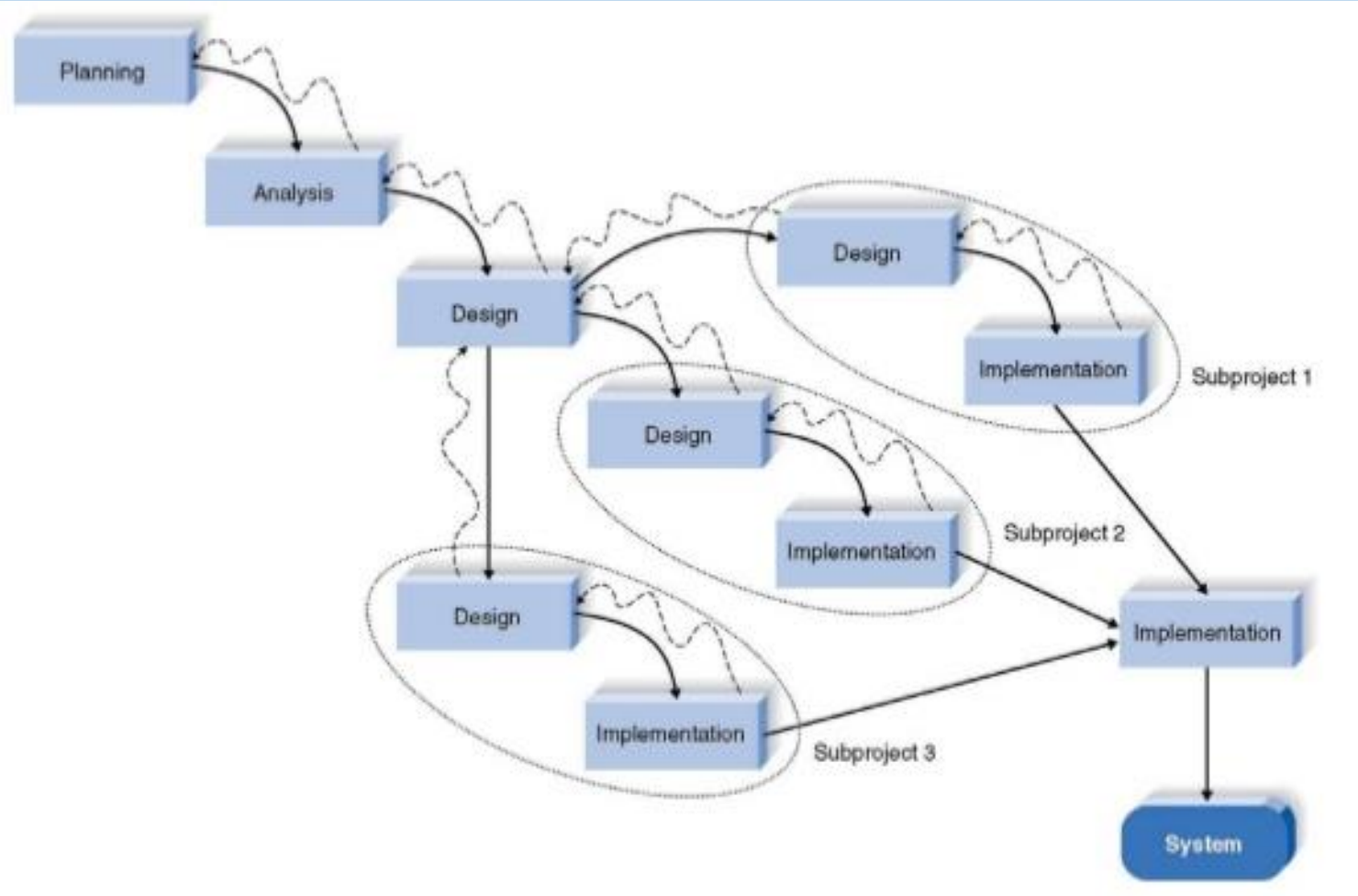
- Prednosti:
  - Unaprijed identifikovani zahtjevi (prije početka razvoja)
  - Promjene u zahtjevima su minimalne

- Nedostaci:
  - Dizajn mora biti u potpunosti završen prije početka implementacije
  - Vrijeme provedeno na svakoj od faza

Paralelni pristup

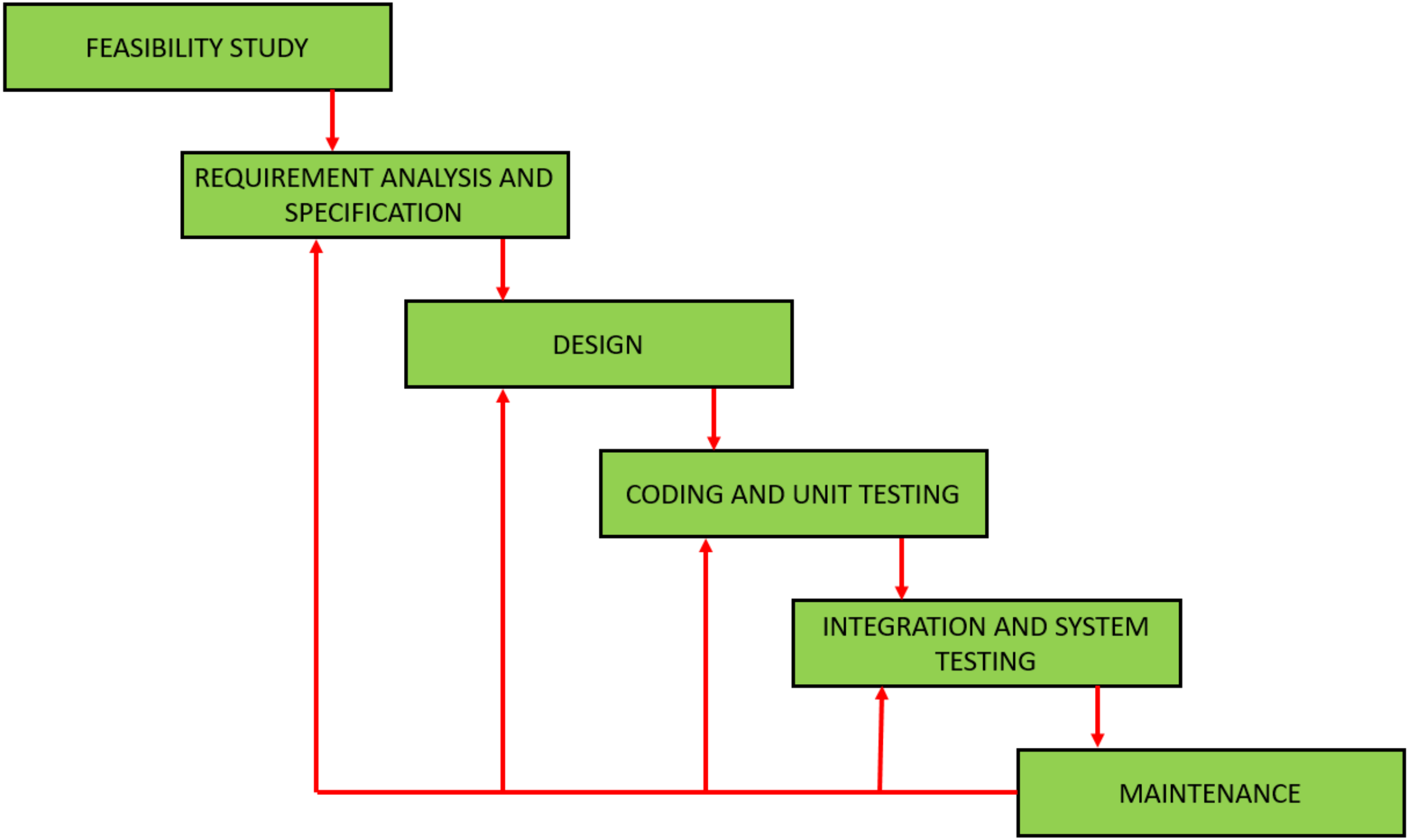
- Metodologija paralelnog razvoja nastoji smanjiti vrijeme između faze analize i isporuke softverskog proizvoda. Prvo se napravi dizajn cjelokupnog proizvoda, a zatim se projekt dijeli na više podprojekata. Svi podprojekti, ukoliko ne postoji međusobna ovisnost među njima, se realiziraju paralelno. Rezultati svakog od podprojekata potom se integriraju u jednu cjelinu, projekt.

- Prednosti:
  - Znatno se smanjuje vrijeme razvoja
  - Manje su šanse za prepravke
- Nedostaci:
  - Potrebno više resursa
  - Ponekad je teško integrirati podprojekte


Planning
Analysis
Design
Design
Implementation
Subproject 1
Design
Implementation
Subproject 2
Design
Implementation
Subproject 3
Implementation
System

Iterativni pristup

- Iterativni ili evolucijski model je model u kojem korištenje softverskog proizvoda mijenja perspektivu korisnika u odnosu na softverski proizvod. Svaka promjena perspektive može rezultirati generiranjem novih korisničkih zahtjeva. Na taj način softverski proizvod raste zajedno sa organizacijom u kojoj se koristi. Ovakav pristup razvoja podrazumijeva brzo prototipiranje, odnosno, gradnju dijelova softverskog proizvoda u inkrementima.
- Prednosti:
  - Softver se proizvodi u ranim fazama životnog ciklusa
  - Korisnici u ranim fazama probaju rezultate
  - Pojednostavljeno testiranje
- Nedostaci:
  - Izmjene zahtjeva
  - Duže vrijeme razvoja


FEASIBILITY STUDY
REQUIREMENT ANALYSIS AND SPECIFICATION
DESIGN
CODING AND UNIT TESTING
INTEGRATION AND SYSTEM TESTING
MAINTENANCE

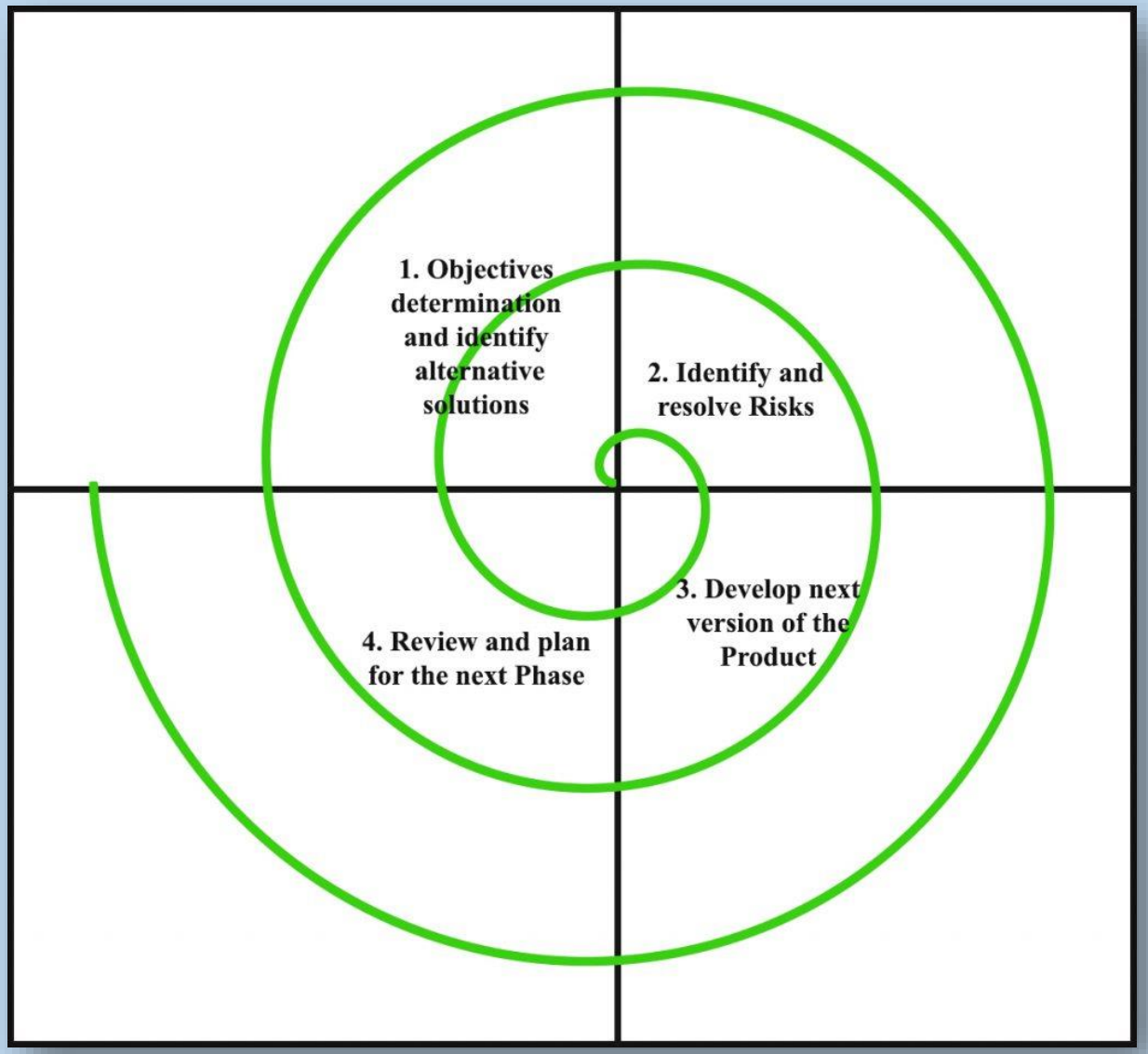
Spiralni model

- Na početku svake faze radi se analiza rizika i nastoje se utvrditi mogući rizici. Cilj analize rizika je eliminirati rizike ili barem ih svesti na najmanju moguću mjeru. Ako je rizik nastavka projekta prevelik, tada se projekat prekida.

- Faze spiralnog modela su:
  - Analiza rizika i procjena alternativa
  - Razvoj i verifikacija sljedećeg "proizvoda„
  - Planiranje sljedeće faze
  - Pregled - Određivanje ciljeva, alternativa i ograničenja.
- Prednosti
  - Visok stepen analize rizika
  - Softver se proizvodi u ranim fazama životnog ciklusa

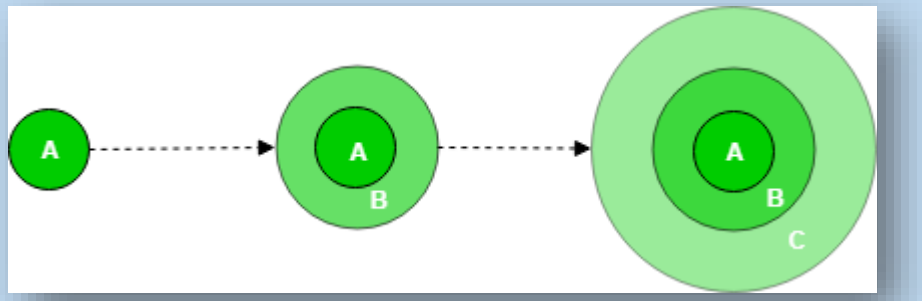
Spiralni model

- Nedostaci
  - Može biti veoma skup model
  - Zahtjeva visok stepen ekspertize u analizi rizika
  - Nije dobar za male projekte


1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

Inkrementalni model

- Model inkrementalnog procesa također je poznat kao model sukcesivne verzije.
- Najprije se izgradi jednostavan radni sustav koji implementira samo nekoliko osnovnih značajki, a zatim se isporučuje kupcu. Zatim se nakon toga mnoge uzastopne iteracije/verzije implementiraju i isporučuju kupcu dok se željeni sustav ne pusti u promet.

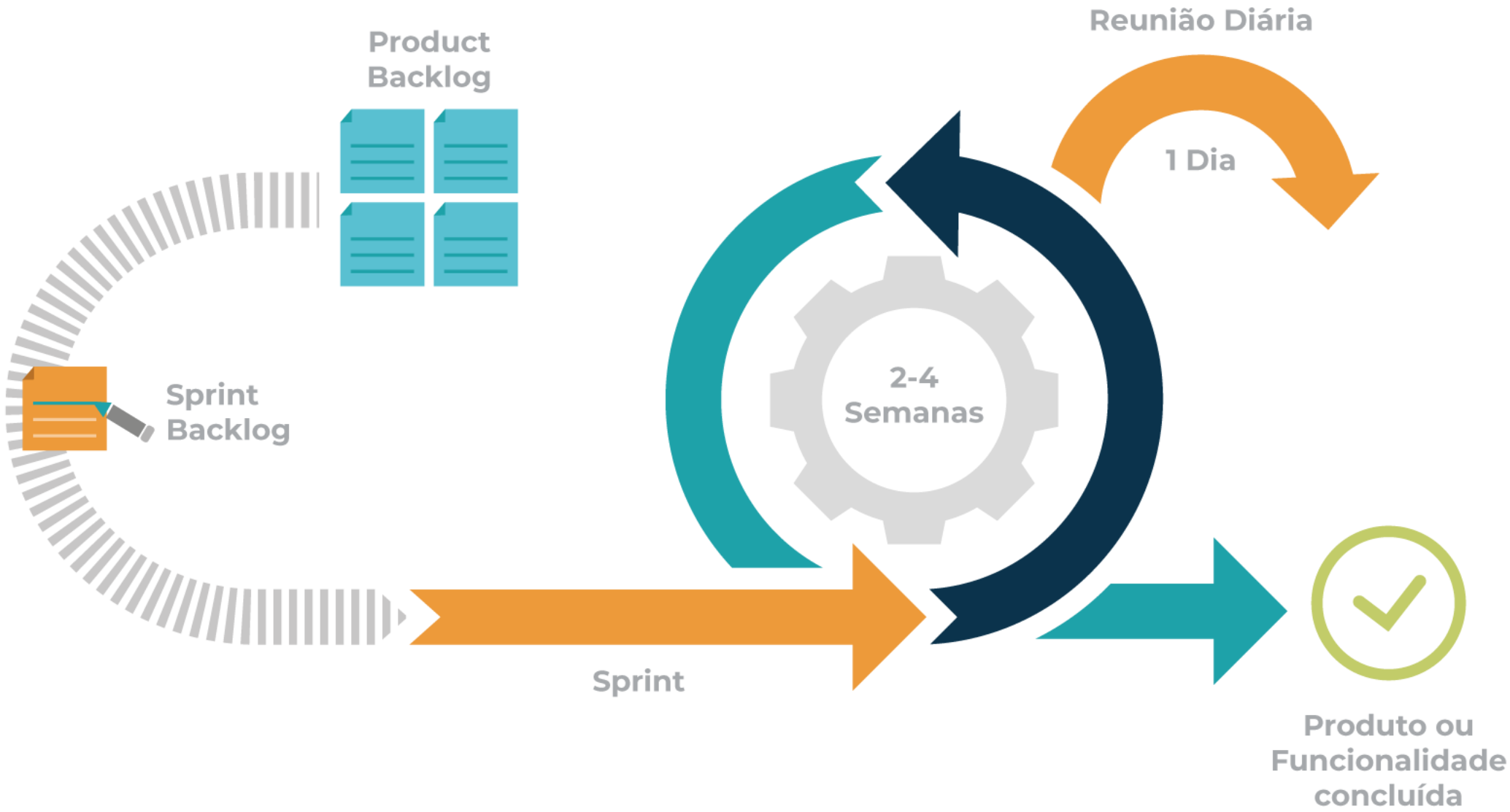


AGILNE METODE

- Scrum
- Kanban
- Extreme Programming (“XP”)
- Lean Development
- Crystal

Scrum

- Scrum je najčešće korišteni od mnogih okvira koji podupiru Agile metodologiju.
- Scrum karakteriziraju ciklusi ili faze razvoja, poznati kao sprintovi i maksimiziranje vremena razvoja za softverski proizvod.
- Obično se koristi u upravljanju razvojem softverskih proizvoda, ali se može uspješno koristiti u poslovnom kontekstu.
- Svaki dan počinje malim 15-minutnim sastankom, dnevnim Scrumom, koji preuzima ulogu sinkronizacije aktivnosti i pronalaženja najboljeg načina za planiranje radnog dana.


Reunião Diária
Product Backlog
1 Dia
Sprint Backlog
2-4 Semanas
Sprint
Produto ou Funcionalidade concluída

Scrum

PREDNOSTI

- Motivacija tima je dobra jer programeri žele ispuniti rok za svaki sprint;
- Transparentnost omogućuje da projekt prate svi članovi u timu ili čak u cijeloj organizaciji;
- Za provjeru valjanosti zahtjeva koristi se jednostavna "definicija gotovog";
- Usredotočenost na kvalitetu je konstanta kod scrum metode, što rezultira s manje pogrešaka;
- Dinamika ove metode omogućuje programerima da reorganiziraju prioritete, osiguravajući da sprintovi koji još nisu dovršeni dobiju više pažnje;
- Dobro planiranje sprinta ima prioritet;

Scrum

NEDOSTACI

- Segmentacija projekta i potraga za agilnošću razvoja ponekad može dovesti do toga da tim izgubi pojam o projektu kao cjelini, fokusirajući se na jedan dio;
- Svaka uloga razvojnog programera možda nije dobro definirana, što može dovesti do zabune među članovima tima.

Kanban

U praksi, Kanban metoda organizirana je na ploči ili tablici (Kanban ploča), podijeljenoj u stupce, prikazujući svaki tijek unutar projekta proizvodnje softvera. Kako se razvoj razvija, informacije sadržane u tablici se mijenjaju, a svaki put kada novi zadatak dođe u igru, stvara se nova "kartica".

Ova metodologija također je korisna u pojedinačnim poslovnim odjelima, kao što su HR, marketing itd., donoseći željenu vidljivost nad svim zadacima tima.

Kanban metoda zahtijeva komunikaciju i transparentnost tako da svi članovi bilo kojeg tima znaju točno u kojoj je fazi razvoja i mogu vidjeti status projekta u bilo kojem trenutku. Prvenstveno je usmjeren na kapacitet tima i najbolji je za procese koji prolaze kroz male promjene.


TO DO
DOING
DONE

Kanban

PREDNOSTI

- Mogućnost pregleda svih zadataka u okviru jednog projekta;
- Možete ograničiti broj pokrenutih zadataka
- Fokusira se na trajanje ciklusa;
- Omogućuje kontinuirane isporuke;
- Vjerojatno jedna od najjednostavnijih metodologija za implementaciju izvan “IT svijeta”.

Kanban

NEDOSTACI

- Moguće je da članovi tima pogrešno protumače informacije prikazane na Kanban ploči, posebno kada su prikazane kao zastarjele;
- Budući da u Kanbanu nema vremenskih okvira, možete se suočiti s problemima povezanim s vremenom, poput kašnjenja, u svakoj fazi.

Ekstremno programiranje (XP)

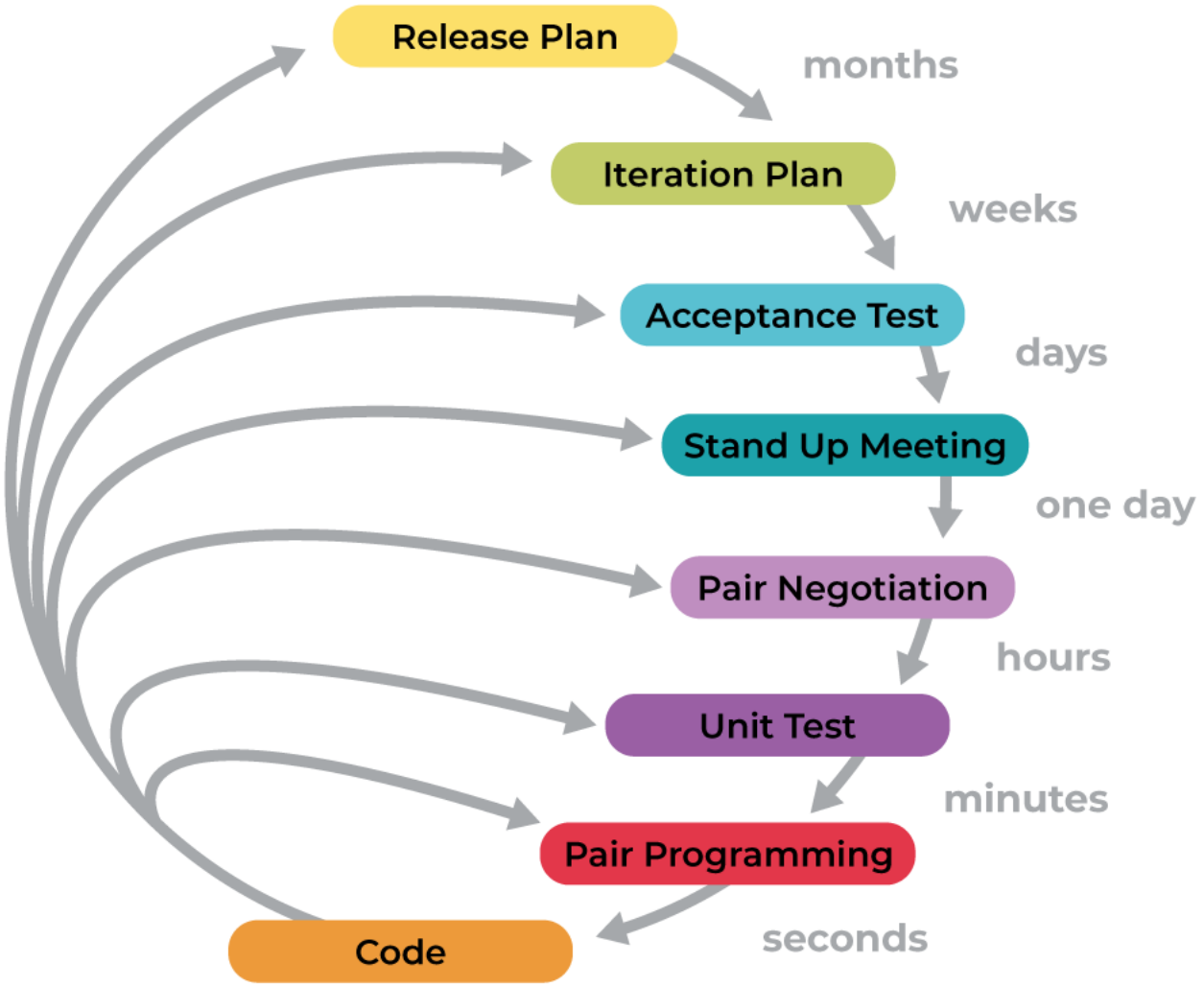
- Metodologija ekstremnog programiranja (“XP”) temelji se na ideji otkrivanja “najjednostavnije stvari koja će raditi” bez stavljanja prevelike težine na dugoročni pogled na proizvod.

- To je metodologija koja naglašava vrijednosti kao što su komunikacija, jednostavnost, povratne informacije, hrabrost i poštovanje te daje prednost zadovoljstvu korisnika nad svim ostalim. Ova metodologija potiče povjerenje motivirajući programere da prihvate promjene u zahtjevima korisnika, čak i ako one stignu tijekom kasnijih faza razvojnog ciklusa.

Ekstremno programiranje (XP)

- Timski rad iznimno je važan u XP-u, budući da, kada postoji problem, rješava ga cijeli tim menadžera, programera ili kupaca, okupljajući ih kako bi promovirali razgovor i angažman i razbili prepreke u komunikaciji. Svi oni postaju ključni dijelovi iste slagalice, stvarajući plodno okruženje za visoku produktivnost i učinkovitost unutar timova.
- Softver se testira od prvog dana, prikupljajući povratne informacije za poboljšanje razvoja. XP promiče aktivnosti kao što je **programiranje u paru**, a uz jaku komponentu testiranja izvrsna je inženjerska metodologija.

Extreme Programming
Planning / Feedback Loops



Ekstremno programiranje (XP)

PREDNOSTI

- Jednostavnost pisanog koda je prednost jer omogućuje poboljšanje u bilo kojem trenutku;
- Cijeli proces i razvojni ciklus XP-a je vidljiv, stvarajući ciljeve za programere uz relativno brze rezultate;
- Razvoj softvera je agilniji nego kod korištenja drugih metodologija, zbog stalnog testiranja;
- Promiče visoki tempo rada;
- XP također doprinosi podizanju i održavanju timskog talenta.

Ekstremno programiranje (XP)

NEDOSTACI

- Ekstremna usredotočenost na kod može dovesti do toga da se manje važnosti pridaje dizajnu, što znači da mu se kasnije mora posvetiti dodatna pozornost;
- Ovaj okvir možda neće raditi najbolje ako se svi članovi tima ne nalaze na istom geografskom području;
- U XP projektima, registar mogućih grešaka se ne održava uvijek, a ovaj nedostatak praćenja može dovesti do sličnih grešaka u budućnosti.

Lean metoda

Lean razvoj tjera tim da ukloni sve aktivnosti koje ne donose konačnu vrijednost proizvodu.

Postoji sedam osnovnih načela:

- ✓ brisanje stvari koje nisu bitne (sve što ne donosi učinkovitu vrijednost projektu kupca);
- ✓ razvoj kvalitete (stvaranje kvalitete u razvoju zahtijeva disciplinu i kontrolu broja stvorenih ostataka);
- ✓ stvaranje znanja (tim je motiviran dokumentirati cijelu infrastrukturu kako bi kasnije zadržao tu vrijednost);
- ✓ različite obveze (ova točka potiče tim da se ne fokusira previše na planiranje i predviđanje ideja bez potpunog prethodnog razumijevanja zahtjeva poslovanja);

Lean metoda

- ✓ brza isporuka (dobiti vrijednost kupcu što je prije moguće);
- ✓ poštivanje tima (komunikacija i upravljanje sukobima dvije su bitne točke);
- ✓ optimizirati cjelinu (slijed razvoja mora biti dovoljno usavršen da se mogu brisati greške u kodu,



Crystal

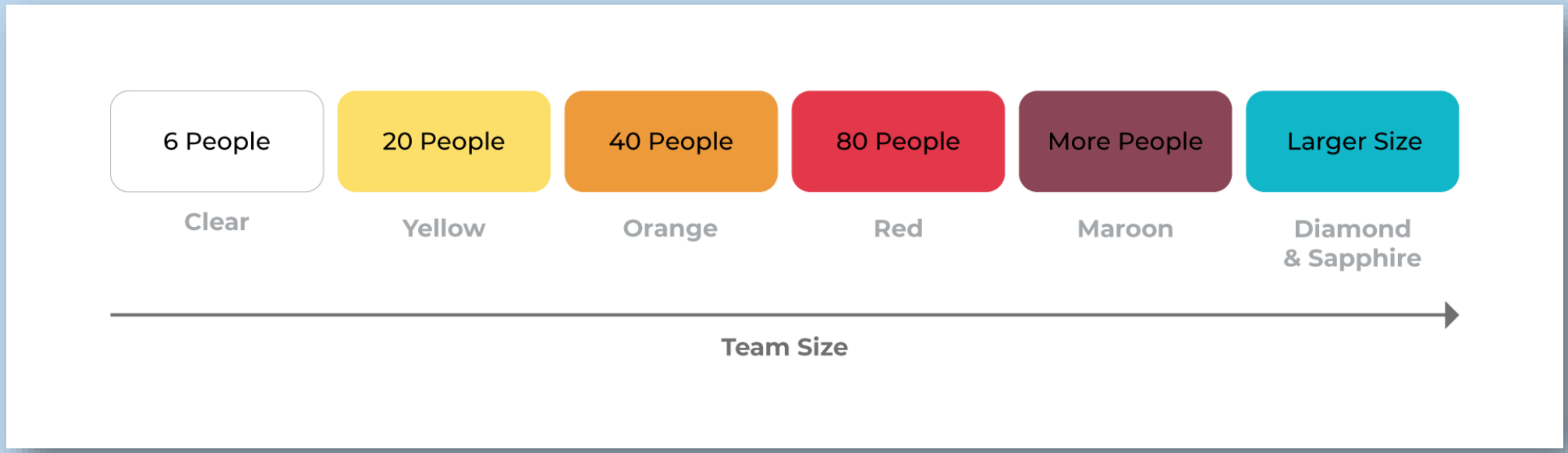
- Crystal je jedan od najfleksibilnijih okvira koji daje ogromnu slobodu timu da razvija vlastite procese. Više se usredotočuje na pojedince i način na koji oni međusobno djeluju, a ne na proces ili alate – stoga je komunikacija ključni aspekt.

- Crystal ima varijante kao što su Crystal Clear (do 8 članova tima), Crystal Yellow (do 10 do 20 članova tima), Crystal Orange (do 20 do 50 članova tima) i Crystal Red (za velike timovi od 50 do 1000 ljudi).

- Crystal se fokusira na principe kao što su ljudi, interakcije, zajednica, vještine, talent i komunikacija, s ciljem pružanja najboljeg mogućeg procesa razvoja softvera. Srž ovog razvojnog procesa je interakcija i simbioza, koji moraju postojati između ljudi koji su dodijeljeni projektima i procesima kako bi projekt bio učinkovit.

Crystal

- Svaki projekt je jedinstven i prolazi kroz česte promjene, tako da tim mora pronaći vlastite načine da ga dovede do kraja koristeći najbolje odluke.
- To je lagana metodologija u smislu dokumentacije, gdje timovi mogu pronaći vlastite načine umjesto preferiranih modaliteta rada.



Hvala na pažnji!