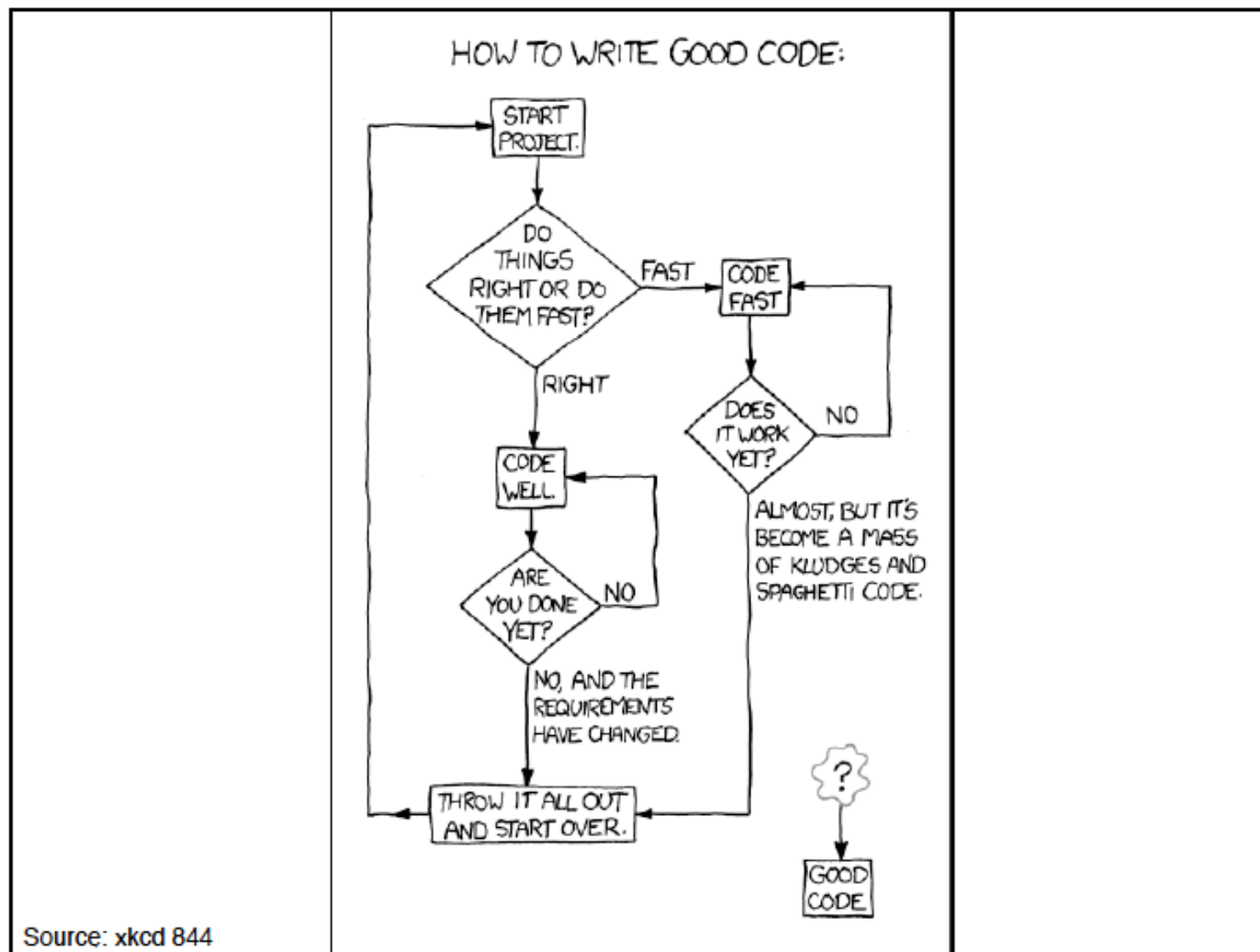


# Uvod u metodologije razvoja softvera

Metodologije razvoja softvera

# Kako napisati (dobar) softver?



# Šta podrazumeva da je softverski projekat uspešan?

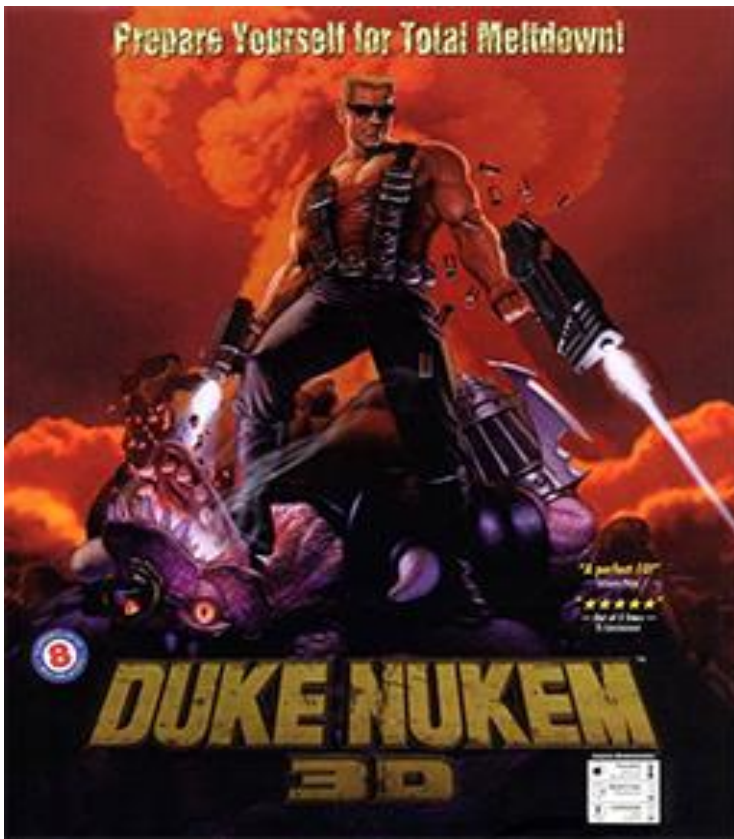
- *“ Od svih IT projekata koji su započeti od 5% do 15% će se odustati čak i pre prvog release-a. Mnogi drugi će imati prvi release koji kasniti i/ili će probiti budžet i/ili će zahtevati ozbiljne izmene jer funkcionalnosti/performance nisu zadovoljavajuće. Samo nekoliko će zaista i biti uspešni. ” --- Robert Charette, Why Software Fails*

# Šta podrazumeva da je softverski projekat uspešan?

- Softver je isporučen na vreme
- Troškovi razvoja su u okviru budžeta
- Softver je zadovoljio zahteve korisnika po pitanju
  - Funkcionalnosti
  - Kvaliteta

# Duke Nukem Forever

- Prva verzija 1996
- Sledeća verzija planirana za 1997, a pojavila se tek 2011



# Tehnike za izbegavanje neuspeha

- Od 1960 pojavljuju se tehnike kako prevazići neuspehe u softverskim projektima
- Ne postoji univerzalno rešenje za sve projekte
- Mnoge od tehnika se nisu pokazale kao uspešne
- Problemi
  - Kreativnost – programiranje je u velikoj meri kreativan proces
  - Sloboda – postoji veliki stepen fleksibilnosti za skoro svaki programerski zadatak
    - Funkcionalni aspekti
    - Nefunkcionalni aspekti (skalabilnost, brzina, bezbednost, ...)

# Tehnike za izbegavanje neuspeha

- Jedan od elemenata koji može pomoći u izbegavanju/smanjenju neuspeha u softverskim projektima je primena odgovarajuće metodologije tokom razvoja softvera
- Ipak sama primena metodologije ne znači, a priori, da će neuspeh sigurno biti izbegnut

# Definicija

- Metodologije razvoja softvera (Software Development Methodology - SDM)
- Okruženje za primenu prakse u softverskog inženjerstvu sa posebnim ciljem da se obezbede neophodna sredstva za razvoj softverskih sistema
- Sastoji se od dve osnovne celine
  - Skupa pravila za modelovanje, tj. jezik za modelovanje (sintaksa i semantika)
  - Proces, koji
    - Daje smernice za redosled aktivnosti
    - Definiše koji elementi treba da se razviju koristeći jezike za modelovanje
    - Usmerava zadatke pojedinih inženjera i tima kao celine
    - Obezbeđuje kriterijume za praćenje i merenje proizvoda (softvera) i aktivnosti na projektu



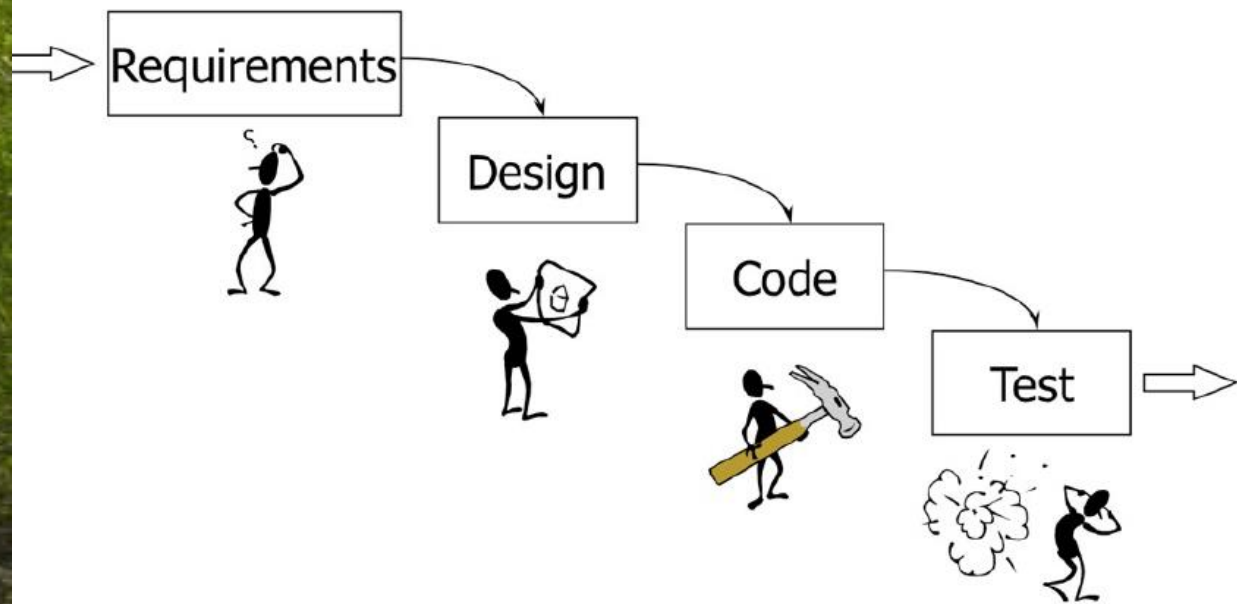
# Razvoj metodologija

- Metodologije se mogu svrstati u skladu sa okolnostima koje su dovele do njihovog razvoja uključujući primenjeni pristup i metode
  - *Revolucionarne* – nove ideje i pristupi
  - *Evolutivne* – bazirane na postojećim metodologijama
  - *Integracione* – preuzima ideje/pristupe iz dve ili više metodologija
    - *Spajanje* - Obično se odvija po “design-by-committee” principu (određeno telo kreira novu metodologiju na osnovu postojećih)
    - *Ad hoc* – karakteristike preuzete iz istaknutih metodologija kako bi se zadovoljili zahtevi korisnika koji je kreirao
    - *Isplanirane (Engineered)* – nastale na osnovu analize problema u odgovarajućem domenu i na osnovu identifikovanih zahteva

# Waterfall metodologija

- Jedna od prvih metodologija koja se pojavila za potrebe razvoja softvera

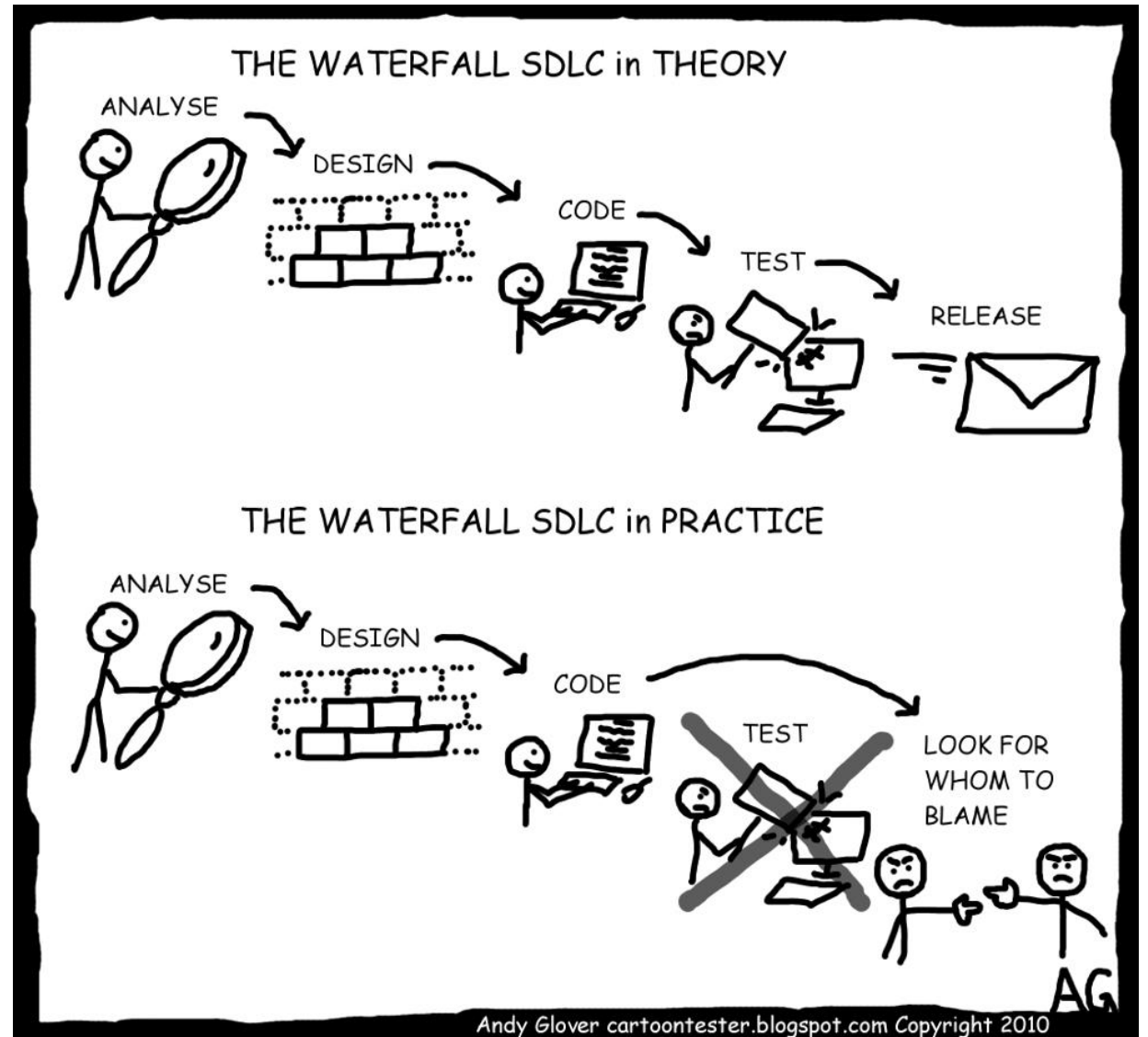
# Waterfall metodologija



# Waterfall metodologija

- Pomogla da se smanji stopa neuspeha u softverskim projektima
- **Ali** čak do 70% softverski projekata koji koriste ovu metodologiju ne ispune bar deo svojih ciljeva (funktionalnosti, vreme, resurse, finansije)

# Waterfall metodologija – posledica



# Waterfall metodologija – Pogrešne pretpostavke

- Proces razvoja softvera može da se definiše kao predvidljiv i ponavljajući proces
- Što više planiramo na početku, imaćemo manje promena na samom projektu
- Moguće je "prikupiti" sve zahteve unapred ili čak i "razumeti" sve zahteve unapred
- Sve prikupljene informacije moguće je efikasno napisati u dokumentu bez neke velike kolaboracije između ljudi

# Waterfall metodologija – Realnost

- Uvek postoji promena
  - Neizvesnost/neodređenost se najbolje smanjuje ako se iterativno uči i o proizvodu i o procesu u okviru koga se proizvod razvija
- Zahtevi evoluiraju kako naše znanje raste – na osnovu iskustva i povratnih informacija (feedback-a)
- Razvoj novih proizvoda je evoluirajući i adaptivni proces
- Bitno znanje se gubi prilikom svakog vida “primopredaje” i ljudska interakcija je neophodna da bi se to prevazišlo

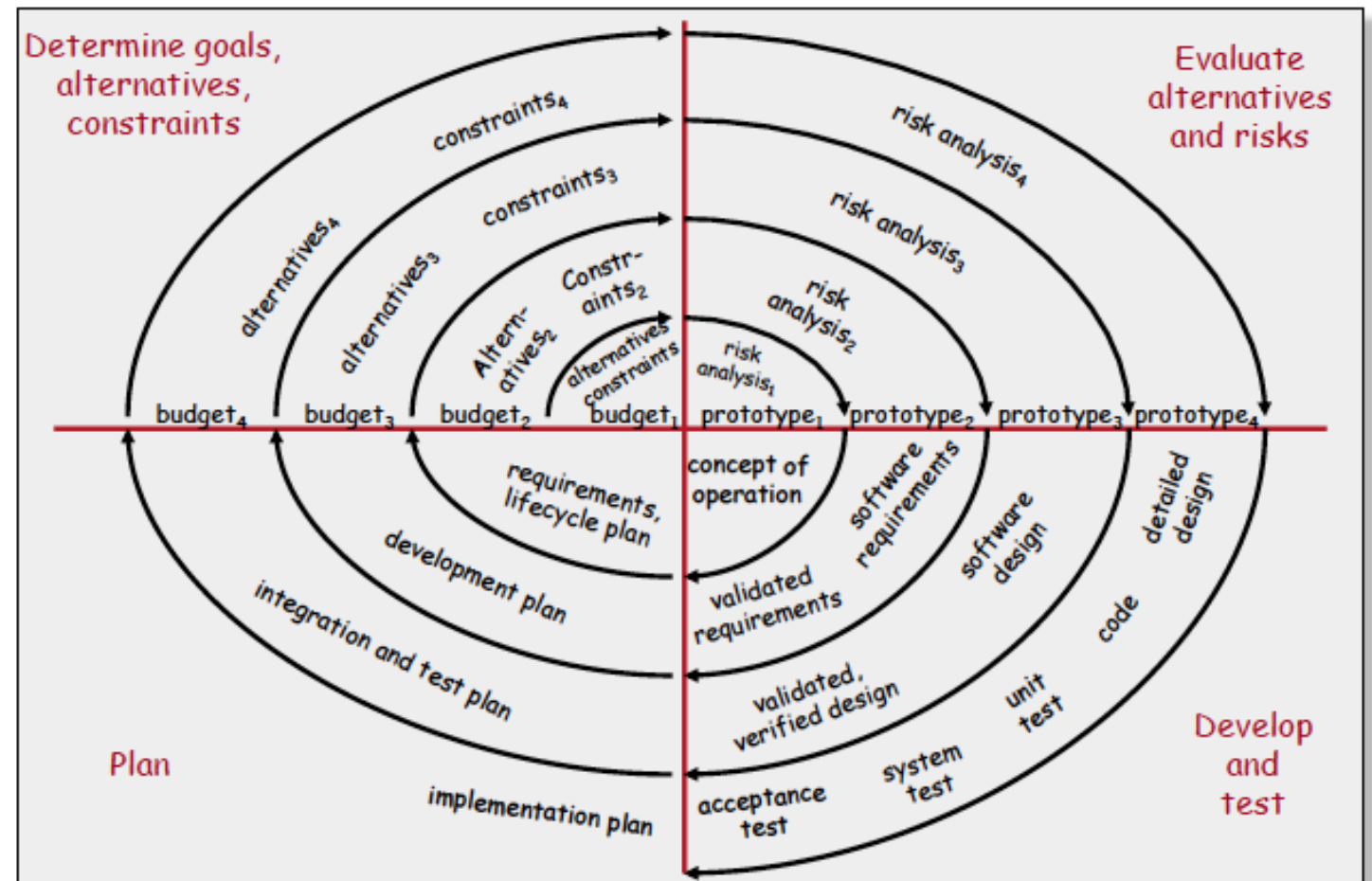
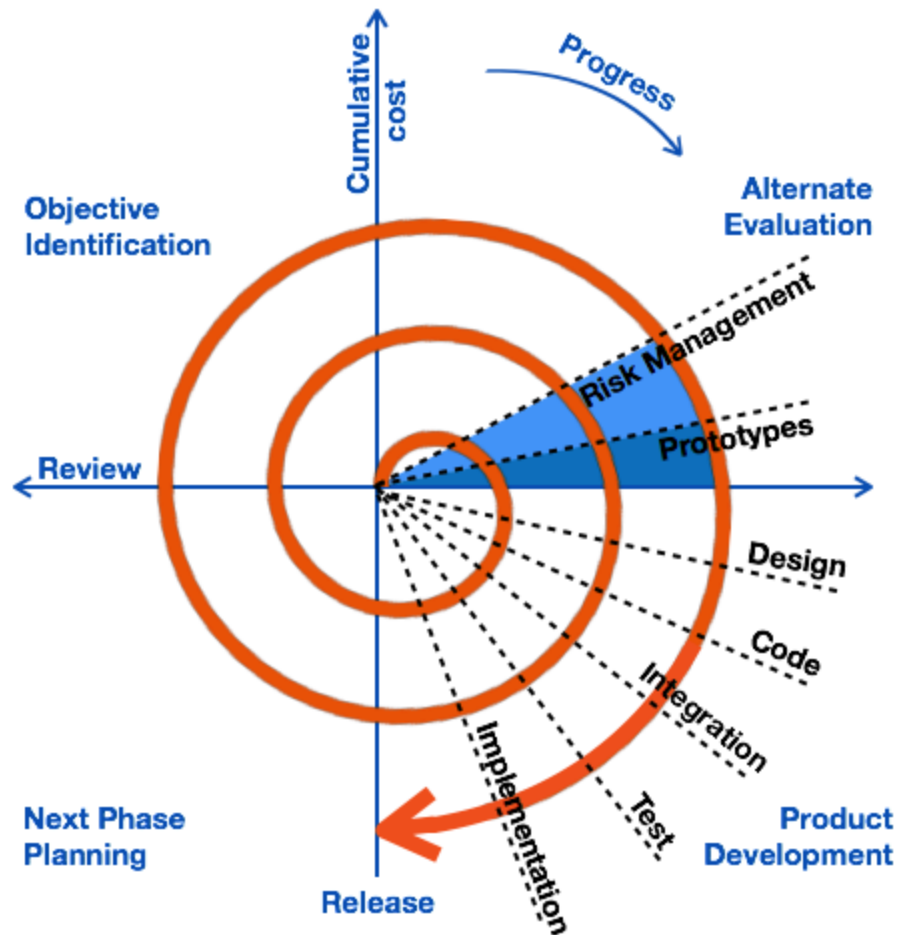
# Metodologije posle waterfall-a

- Paralelni razvoj
  - Umesto da se dizajn i implementacija rade sekvencijalno, formira se neki generalni dizajn čitavog projekta i onda se projekta deli u manje delove
- V – modeli
  - Bazirani na različitim vrstama testiranja
- Spiralni modeli
  - Kombinacija waterfall modela sa iterativnim razvojem
- RAD – Rapid Application Development metodologije
  - Fazni razvoj
    - Sistem se sekvencijalno razvija kroz različite verzije
  - Razvoj prototipa
    - Prvo se razvija prototip sistema
  - Razvoj baziran na prototipu koji se odbacuje
    - Razvijaju se prototipu u različitim fazama koji se potom odbacuju



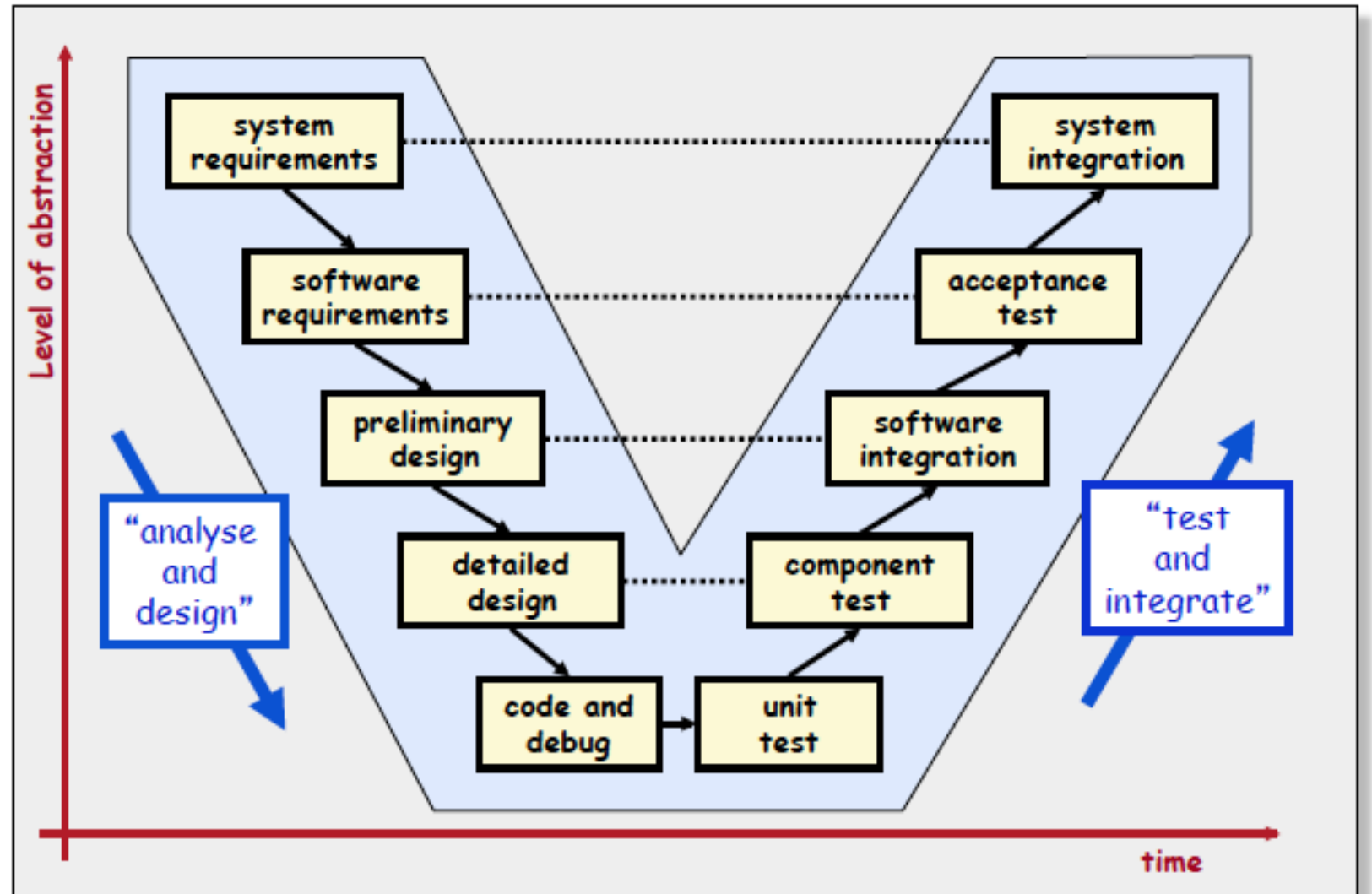
# Metodologije posle waterfall-a

- Sprialni model



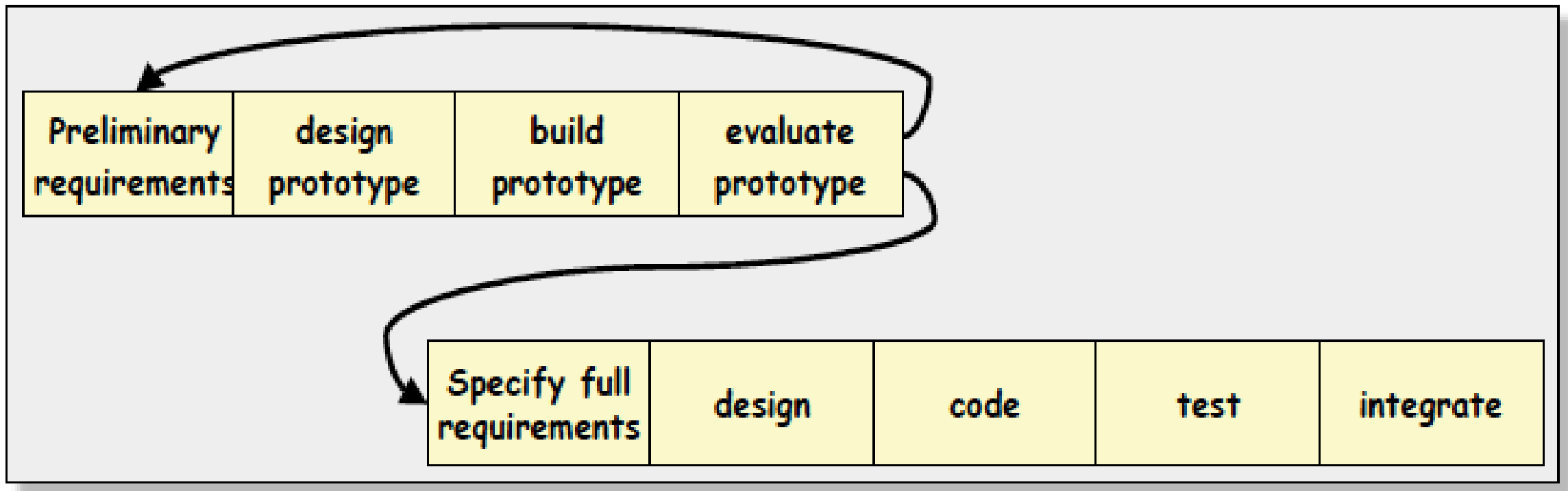
# Metodologije posle waterfall-a

- V-model



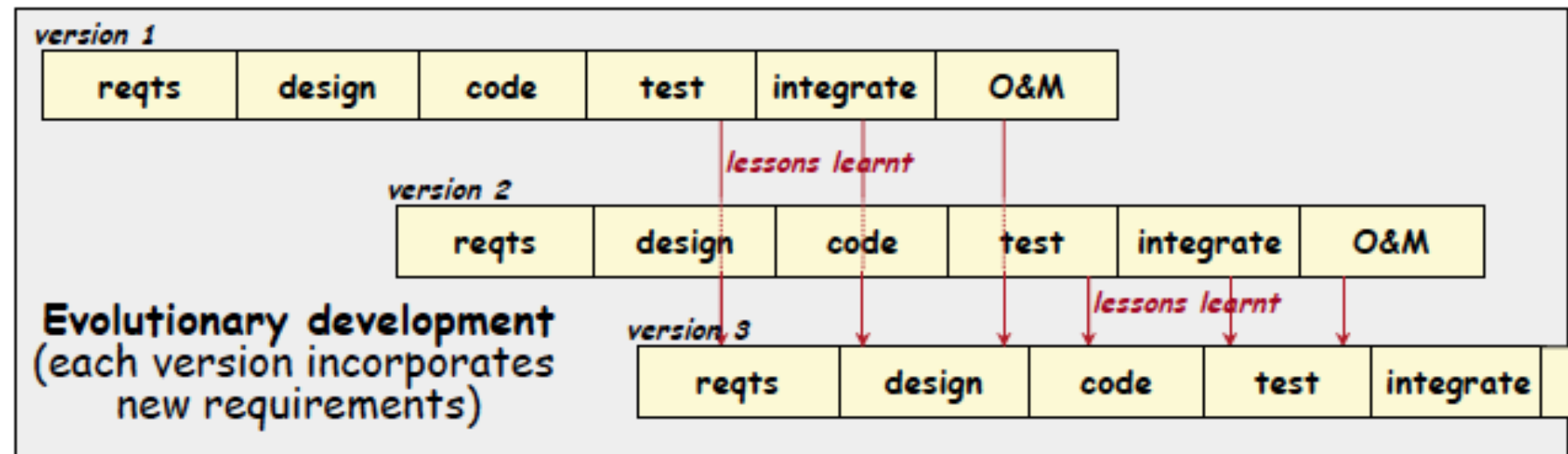
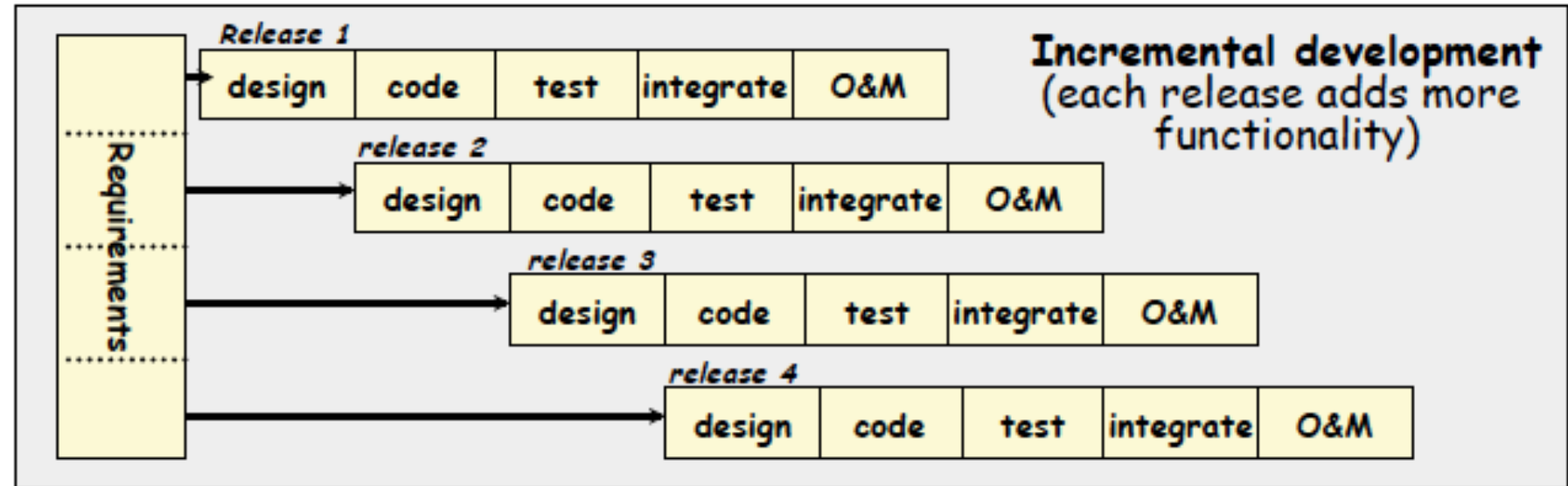
# Metodologije posle waterfall-a

- Razvoj prototipa



# Metodologije posle waterfall-a

- Fazni razvoj



# Metodologije posle waterfall-a

- Agilne metodologije
  - Fokusirane na programiranju
  - Jako malo fiksnih pravila – ideja je u prilagođavanju
  - Otklanjaju neke nedostatke uočene kod ranijih metodologija uključujući i waterfall
  - I dalje nisu **silver bullet**, ali se u praksi pokazalo da vrlo često daju bolje rezultate nego li neke druge metodologije

