

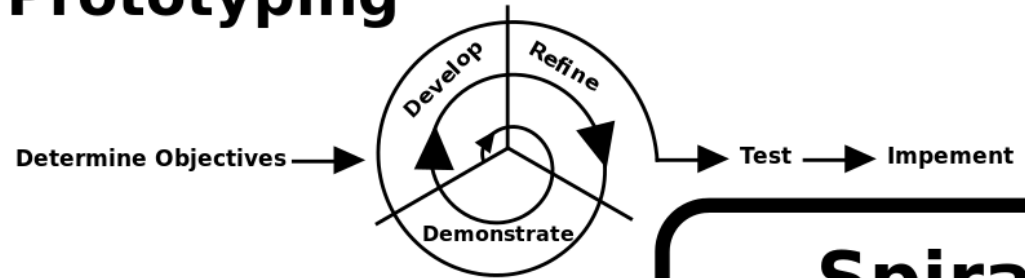


Inxhinieria Softuerike

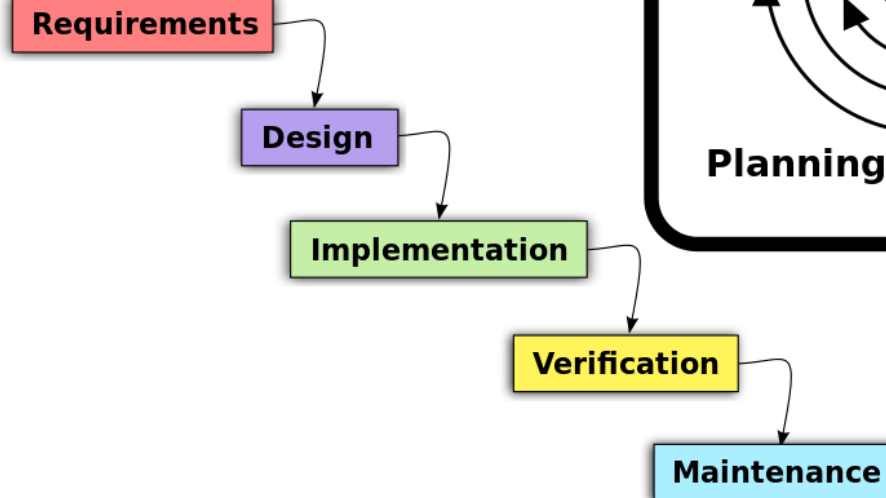
Proceset dhe modelet e proceseve e softuerit (Pjesa 1)

FAKULTETI: SHKENCA KOMPJUTERIKE DHE INXHINIERI

Prototyping



Waterfall



Spiral



Objektivat

- ❑ Ky kapitull shtjellon aktivitetet koherente te prodhimit te softuerit, të cilat krijojnë procesin softurikë
- ❑ Në fund të këtij kapitulli duhet të jeni në gjendje të:
 - Kuptoni konceptet e procesit softurikë dhe modelet e këtoj procesi
 - Te keni njohuri për tre modele të proceseve softurike gjenerike dhe kur mund të përdoren ato
 - Kuptoni organizimin dhe kopjimin me ndryshime të proceseve softuerike



Procesi softuerikë /SDLC-definim (1)

- ❑ **Procesi i softuerit** (gjithashtu njehet si metodologjina e softuerit) është një **grup aktivitesh** që ndërlidhjet me prodhimin e softuerit.
- ❑ **Këto aktivitete** mund të përfshijnë **zhvillimin e softuerit** *nga zeroja*, ose *modifikimin* e një sistemi ekzistues.
- ❑ **Këto aktivitete** përfshijnë zhvillimin e produktit softuerikë nga *fillimi (baza)* e deri në *përfundim*.
- ❑ **Softuerët ekzistues** zakonisht zhvillohen duke i *zgjeruar* dhe *modifikuar* pjesët ekzistuese ose duke e *konfiguruar* dhe *integruar* softuerin në sistemin ekzistues.

4- aktivitetet kryesor të procesit Softuerik

PJESA PARË



Procesi softuerikë /SDLC-definim (2)

□ Çdo proces i softuerit **duhet** të përfshijë katër aktivitetet e mëposhtme:

1. Specifikimet e softuerit (ose kërkesat e inxhinierisë): Përcaktoni funksionalitetet kryesore të softuerit dhe kufizimet rreth tyre.
2. Dizajnimi dhe Implementimi i softuerit: Softueri duhet të projektohet dhe programohet.
3. Verifikimi dhe validimi i softuerit: Softueri duhet të përputhet me specifikimet e tij dhe t'i plotësojë nevojat e klientit.
4. Evuluimi i softuerit (mirëmbajtja e softuerit): Softueri modifikohet për të plotësuar ndryshimet e kërkesave të konsumatorëve dhe të tregut.

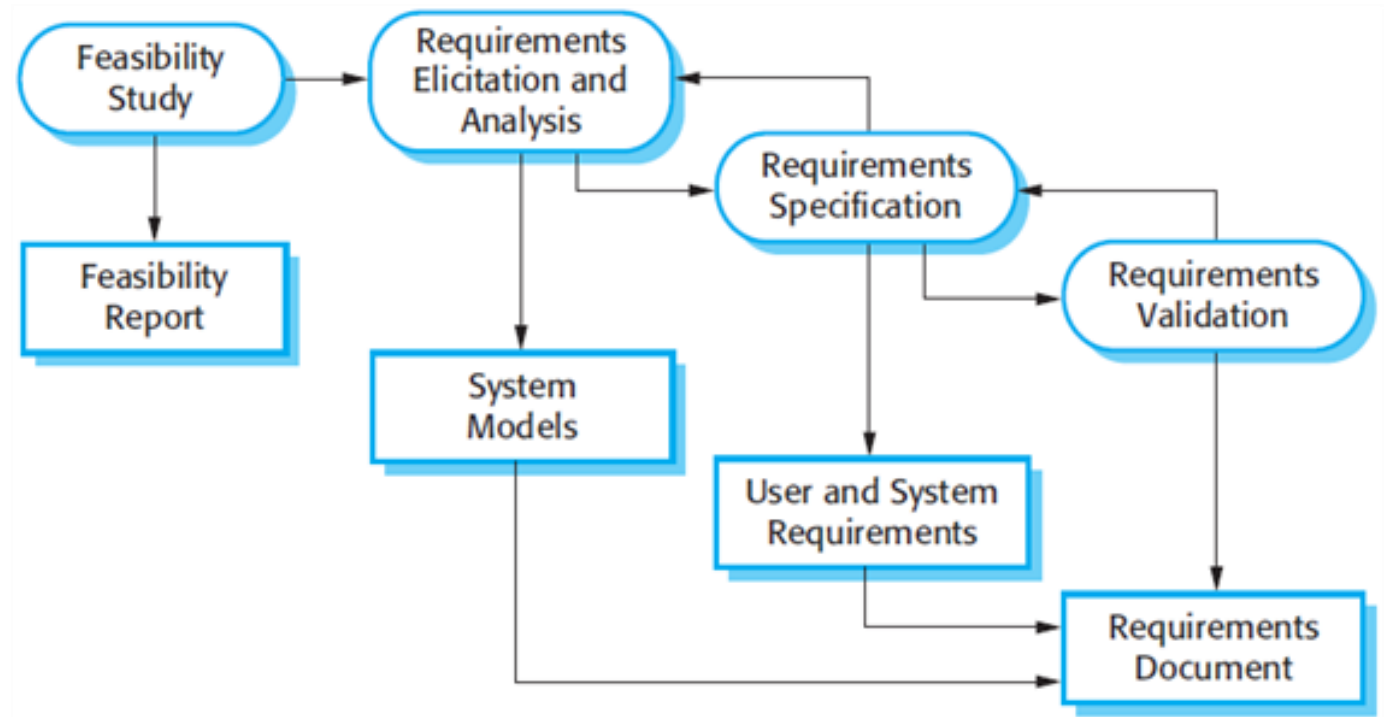
- Në praktikë, këto përfshijnë **nën-aktivitete** të tilla si; *validimi i kërkesave*, *dizajn arkitekturor*, *testimi i njësive*, etj.
- Ekzistojnë gjithashtu aktivitete mbështetëse si *konfigurimi* dhe *menaxhimi i ndryshimeve*, *sigurimi i cilësisë*, *menaxhimi i projektit*, *përvoja e përdoruesit*.

Procesi softuerikë /SDLC-definim (2)

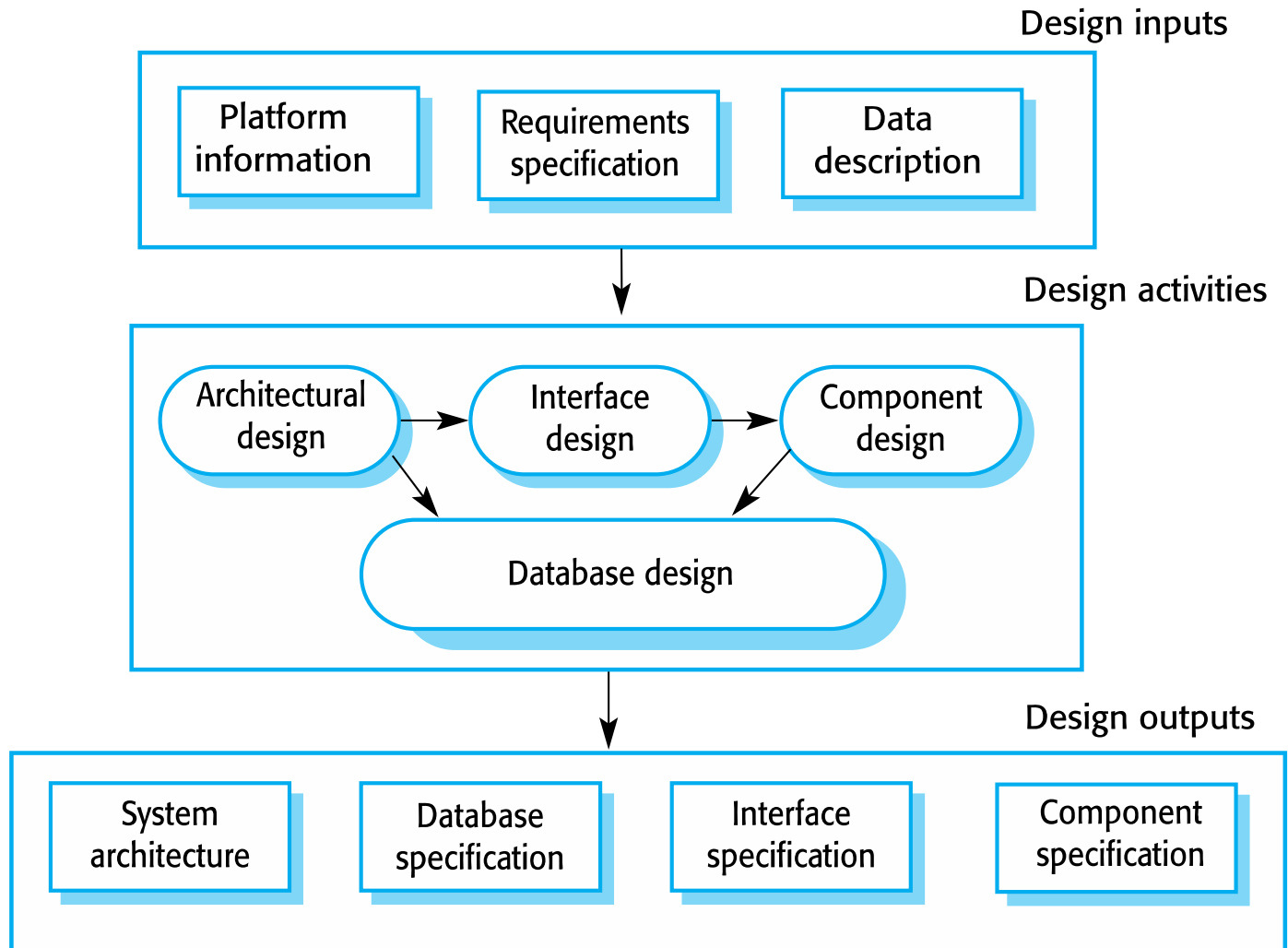
- Kur flasim për një **process softuerik**, zakonisht flasim për **aktivitetet** në të. Megjithatë, një **process softuerik** përfshin gjithashtu **përshkrimin e procesit** i cili përfshin:
- **Produktet**: **rezultatet** e një aktiviteti. p.sh, *rezultati i dizajnit arkitekturor* ndoshta një **model** për *arkitekturën e softuerit*.
 - **Rolet**: **përgjegjësitë** e njerëzve të përfshirë në proces. p.sh, *menaxheri i projektit*, *programuesi*, etj.
 - **Kushtet para** dhe **pas**: **kushtet** që duhet të jenë të **përfshira** para dhe **pas** një aktiviteti.
 - p.sh, kushtet paraprake të *dizajnit arkitekturor* janë *kërkesat e miratuara nga klienti*, ndërsa gjendja pas është *diagramet që përshkruajnë arkitekturën* që janë shqyrtuar.

□ Aktivitetet e proceseve: **Specifikimi i softuerit**

Aktivitetet e proceseve: Specifikimi i softuerit



□ Aktivitetet e proceseve: **Dizajni dhe implementimi**



Një model i përgjithshëm i procesit të dizajnit

Një model i përgjithshëm i
procesit të dizajnit

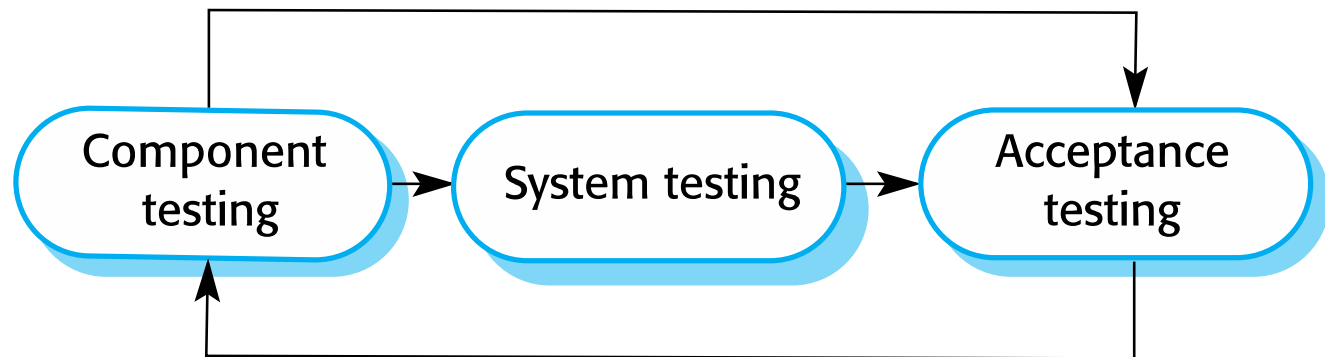


Aktivitetet e proceseve - **Miratimi i softuerit** Software validation

- ☐ **Verifikimi dhe Validimi (V & V)** ka për qëllim për të treguar se një sistem përputhet me specifikimet e tij dhe plotëson kërkesat e klientit në sistemi.
- ☐ Përfshin **kontrollimin** dhe **rishikimin** e proceseve dhe **testimin** e **sistemit**.
- ☐ Testimi i sistemit përfshin ekzekutimin e sistemit me **rastet testuese** që rrjedhin nga **specifikimi i sistemit** dhe për te **dhënave reale** që do të **procesohen** nga sistemi.
- ☐ Testimi është aktiviteti më i zakonshëm **V & V**.

□ Aktivitetet e proceseve - **Miratimi i softuerit**

□ Fazat e Testimit





Aktivitetet e proceseve - **Evuluimi i softuerit**

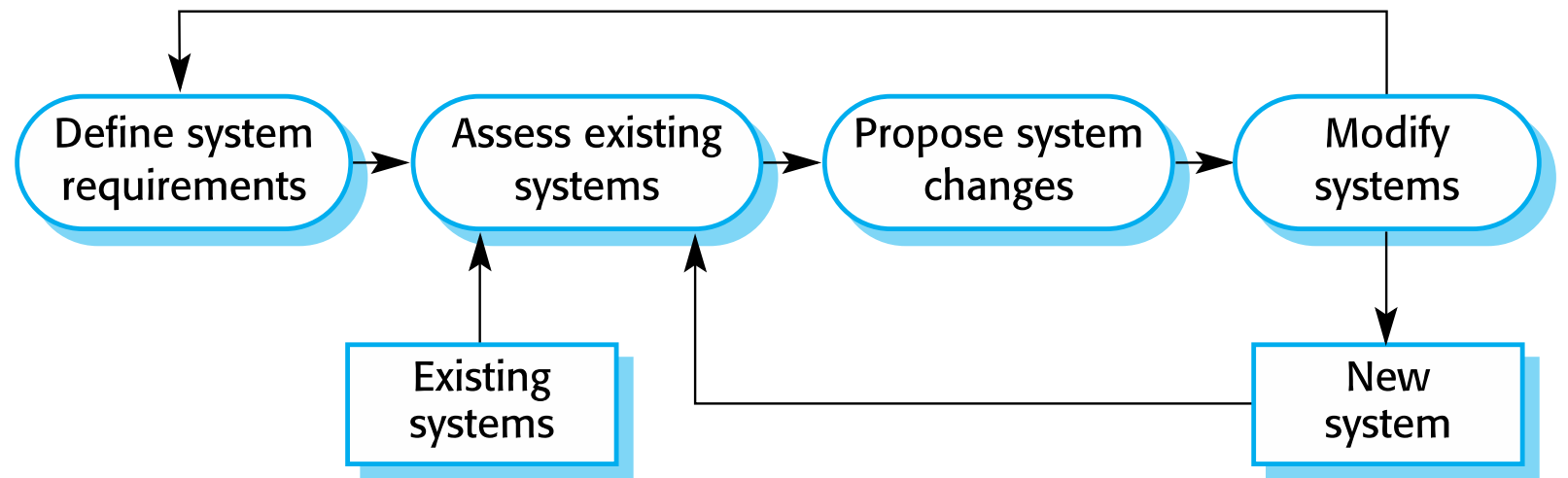
Software evolution

- ☐ Sftueri është e natyrshme **fleksibël** dhe **mund të ndryshojë**.
- ☐ Ndërkohë që **kërkesat ndryshojnë** nëpërmjet ndryshimit të rrethanave të biznesit, *softueri që mbështet biznesin duhet të evoluojë* dhe të **ndryshojë**.
- ☐ Edhe pse ka pasur një ndarje në mes të **zhvillimit** dhe **evolucionit (mirëmbajtjen)** kjo është gjithnjë e më e parëndësishme, si **gjithnjë e më pak** sisteme janë **krejtësisht të re**.



Aktivitetet e proceseve - Evuluimi i softuerit

Aktivitetet e proceseve - Evuluimi i softuerit



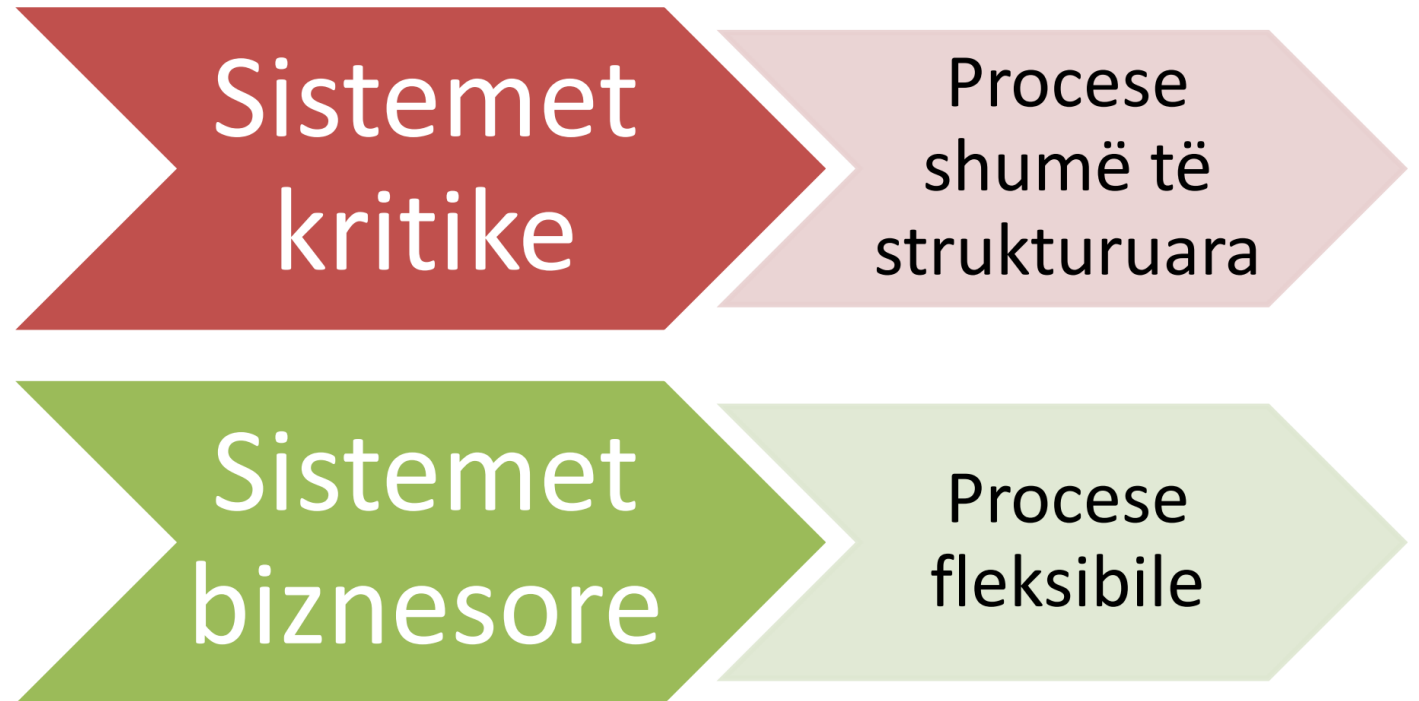
Modelet e Procesit Softuerikë

PJESA DYTË



Procesi softuerikë/SDLC : Llojet (1)

□ P.sh **Sistemet mund të klasifikohen:**



Procesi softuerikë/SDLC – Llojet (2)

Model-drejtuar në planifikim (Plan-Driven) vs. Agile

Waterfall Delivery: *Plan-Driven*



Agile Vs.



Procesi i drejtuar-nga-plani është një proces ku të **gjitha** aktivitetet **janë planifikuar** së **pari** dhe **progresi** matet **kundrejt planit**.

Ndërsa procesi Agile, *planifikimi* është në rritje dhe është më e **lehtë** për të **ndryshuar procesin** për të **reflektuar** në **ndryshimet e kërkesave**.

modelet e Procesit Softuerikë-Llojet

❑ modeli Ujëvara (The waterfall model)

- **Modeli-drejtuar në planifikim(Plan-driven).** Të gjitha aktivitetet të inxhinierisë softuerike i merrë dhe i ndan në faza fikëse

❑ modeli Zhvillimi Rritës/në Rritje (Incremental development)

- **Specifikimi, zhvillimi** dhe **testimi(validimi) ndërthuren**. Mund të jetë **model-drejtuar në planifikim** ose i **agile**
- Krijimi i **versioneve** të **ndryshme** duke i **zhvilluar vazhdimishtë**

❑ modeli i Orientuar në Ripërdorim (Integration and configuration/Reuse oriented SWE)

- Sistemi fokusuar në integrimin e komponentëve ekzistues të konfigurueshëm. Mund të jetë **model-drejtuar në planifikim** ose i **agile**
- Në praktikë, shumica e sistemeve të mëdha zhvillohen duke përdorur një proces që përfshin elementë nga të gjitha këto modele.

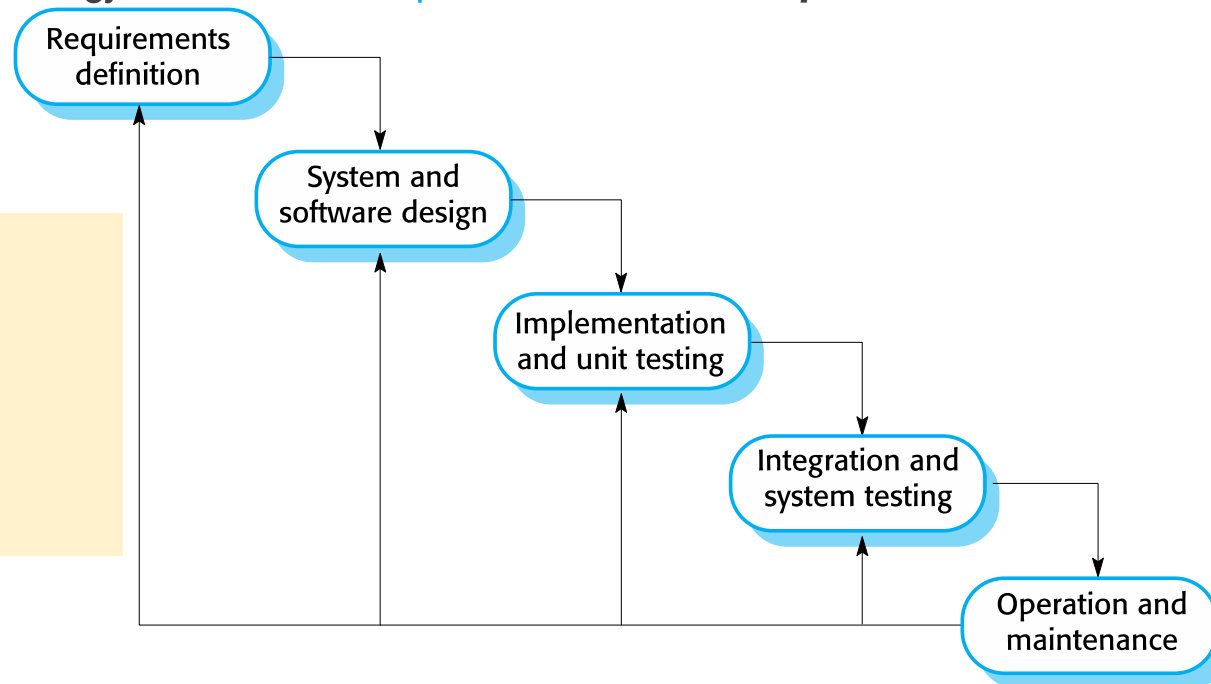
Modeli Ujvarës

□ **Modeli i ujëvarës** është një **qasje sekuenciale**, ku secila aktivitet i procesi të softuerit paraqitet si një **fazë e veçantë**, e organizuar në mënyrë **lineare**.

- Në modelin e **ujëvarë**, duhet të **planifikoni** të gjitha **aktivitetet para** se të **filloni të punoni** në to (**procesi i orientuar-në-planifikim**).

Modeli i parë – (Royce, 1970)
Është i **model-drejtuar në planifikim**

Fazat e modelit të ujëvarë janë: **Kërkesat, Dizajnimi, Implementimi, Testimi dhe Mirëmbajtja**



Natyra e fazave të Ujëvarës

□ Në parim **rezultati i secilës fazë** është një ose më shumë **dokumente** që duhet të oprovohen si rrjedhojë **faza tjetër nuk duhet** të **fillojë** derisa **faza e mëparshme** të jetë **përfunduar plotësisht**.

Në praktikë, megjithatë, këto faza ovelapin (mbivendosin) dhe japin informacionin për njëri-tjetrin.

- p.sh, **gjatë dizajnit mund** të identifikohen probleme me kërkesat, dhe gjatë kodimit mund të gjenden disa nga **problemet e dizajnit** etj.
- pra, **procesi i softuerit nuk është një linjë lineare**, por përfshin **informata** nga **një fazë** në **tjetrën**. Pra, dokumentet e prodhuara **në secilën fazë** pastaj **mund të modifikohen** për të **reflektuar** ndryshimet e bëra.

Kur duhet të përdoret një model ujëvarë?

- Në parim **modeli i ujëvarës** duhet të *zbatohet/implementohet* vetëm kur **kërkesat janë kuptuar mirë** dhe ***nuk ka gjasa të ndryshojnë rrënjësisht gjatë zhvillimit***, pasi ky model ka ***një strukturë relativisht të ngurtë*** e cila e bën ***relativisht të vështirë*** për të akomoduar ***ndryshimet*** kur procesi është ***duke u zhvilluar***.
- Modeli i ujëvarë përdoret kryesisht për projekte inxhinierike të sistemeve të ***mëdha/kritike*** ku një sistem zhvillohet në disa vende.
 - Në këto rrethana, natyra e **modelit të drejtuar në planifikim** të ujëvarë ndihmon në koordinimin e punëve.

Problemet e modelit të ujëvarës

- Ndarja e **jo-flexibile** e projektit në faza të veqanta e bën të vështirë t'i përgjigjet kërkesave të klientëve për ndryshime.
- Prandaj, ky model është i përshtatshëm vetëm kur kërkesat janë të **definuara** dhe **ndryshimet** do të jenë **shumë të kufizuara** gjatë procesit të dizajnit.
 - Pak sisteme biznesi kanë **kërkesa të qëndrueshme** (mire të definuara).

Ky model duhet të përdoret kur **kërkesat** janë **mirë të kuptuara** dhe **nuk ndryshojnë** shumë

Çdo fazë gjeneron dokument

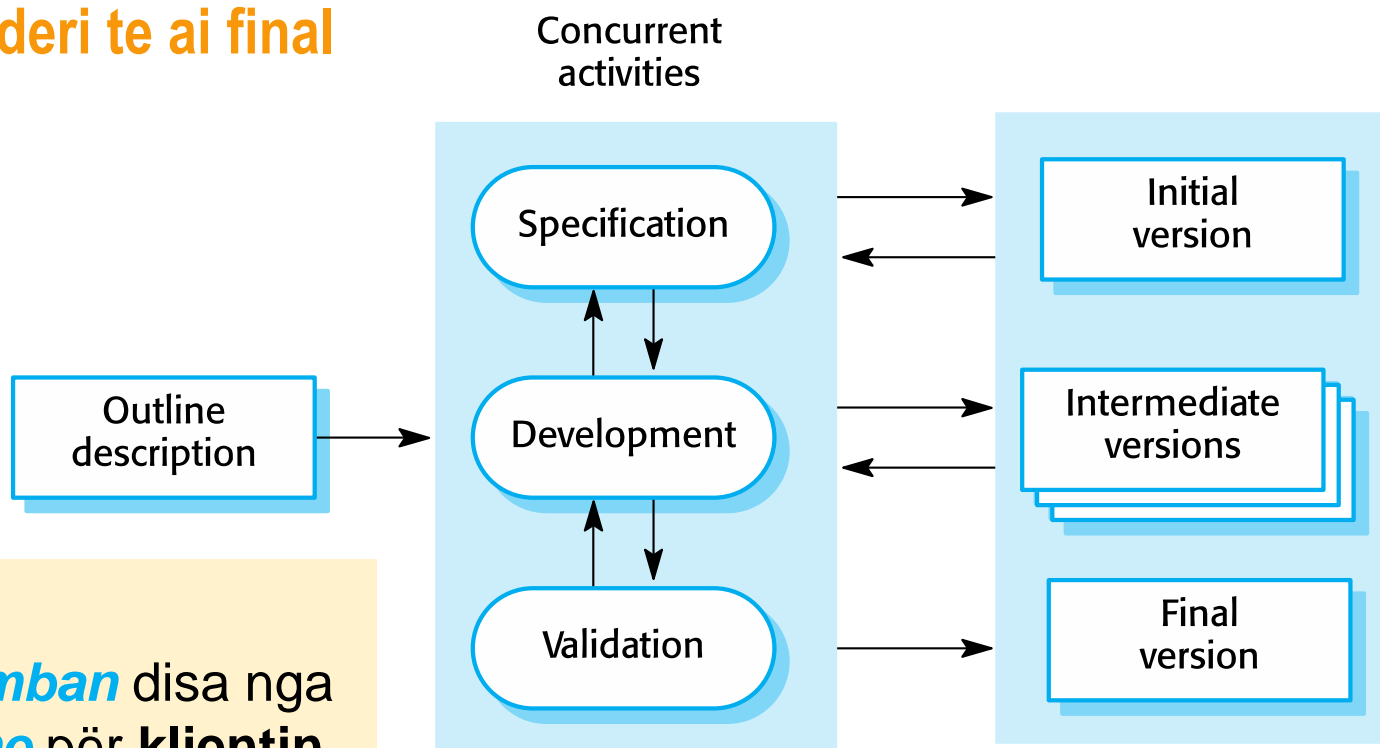
I mirë për menaxhim

Çdo kthim në iteracion kushton

Jo fleksibil

Modeli rritës - Incremental

□ Modeli bazohet në idenë e zhvillimit të *një versioni fillimisht* dhe më pas duke *përditësuar* këtë *versioni deri te ai final*



- Pjesë e qasjes **Agile**
- Çdo **version** rritës *përmban* disa nga *funskionet e nevojshme* për klientin

përfitime e implementimi të modelit rritës

- ❑ **Akomidimi i kostove/hargjimeve** në rast të *ndryshimit të kërkesave* të konsumatorëve është *zvogëluar*.
 - **Sasia e analizës dhe dokumentacionit** që duhet të *ripërpunohet* është *shumë më vogël*
- ❑ **Feedback** të vazhdueshm nga **klientët** mbi atë se *qka është zhvilluar* ose është *bërë*.
 - **Konsumatorët** mund të analizojn në atë se **qka ofron sistemi softuerik** në fazën përkatëse duke *demonstruar* dhe të *vlersaua* se sa është implementuar/zbatuar
- ❑ Ofrohet mundësia *më e shpejtë* ose *form rapide* të **implementimit/përdorimit** të softuerit
 - **Konsumatorët** janë në gjendje të *përdorin sistemin* dhe për të fituar vlerën nga softueri më herët se është e mundur me një **proces i ujëvarës**.



modeli rritës-Incremental: përparsit & mangësit

□ përparsitë ndaj modelit ujëvara janë:

- *Akomodimi i shpenzimeve* të ndryshimeve të kërkesave të klientit zvogëlohen
- Pranohet një *feedback* nga ana *klientit në fazat e mëhershme* para se të programohet
- Pjesët e softuerit mund të *dorëzohen* si rrjedhojë sistemi të fillohen punën me shpejt

□ mangësitë nga ana menaxheriale:

- Procesi nuk është i *dukshëm*, pra *nuk dokumentohen proceset* çdo here.
- Struktura e sistemit *dëmtohet* për çdo version.
- *Probleme* me sisteme të “*medha*”



PJESA DYTË

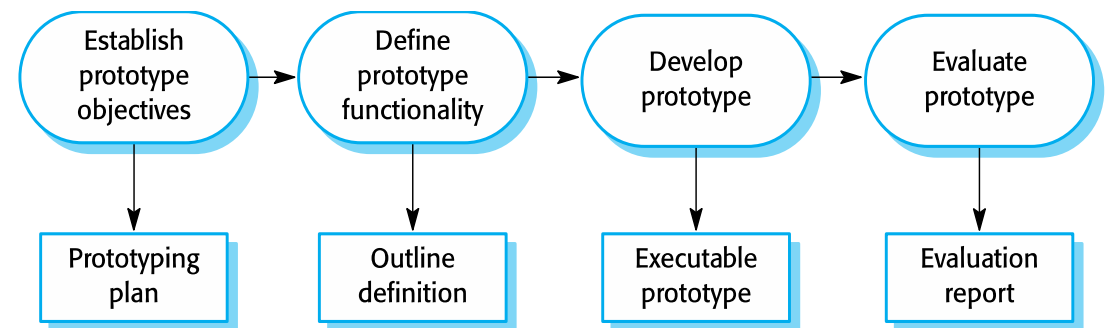
Modeli i Krijimi të prototipit (1)

- ❑ **Modeli Prototipi** paraqet një version *inicial* të një sistemi softuerikë që përdoret për të **demonstruar koncepte**, duke përdorë opsionet themelore dhe duke **zbuluar problemet** dhe zgjidhjet e mundshme.
- ❑ Zhvillimi i shpejtë **ri-përsëritës** i **prototipit** është esencial për **zvogëlimin e shpenzimeve të ri-bërjes** së zhvillimit.
- ❑ Prototip mund të përdoret/ndihmojë në:

Në procesin e **inxhinierisë së kërkesave** dhe **validon kërkesat** e sistemit ;

Në procesin e **dizajnit të sistemit**, dhe për të **analizuar zgjidhjen përkatëse të softuerit** dhe të ndihmojë **dizajnin e ndërfaqes (interface-UI)**;

Në procesin e **testimit** për të ekzekutur **testet një-pas-një**.





Përparësitë e modelit të krijimit të prototipimit

☐ Përmirësimi i **cilësisë së dizajnit**

- Ide të reja për zhvillim.
- Një nderlidhje më e afërt me nevojat reale të përdoruesve.
- Ide të reja për dizajn

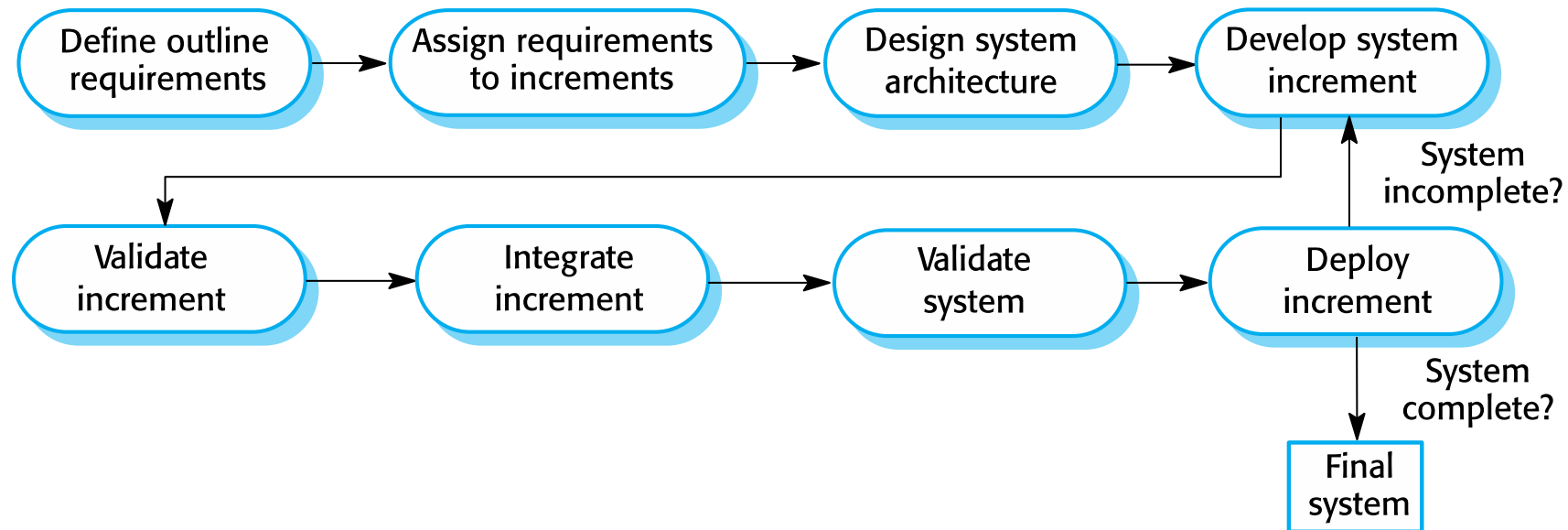
☐ Përmirësimi në **mirëmbajtje** të sistemit.

☐ Kohë më të **vogël për trajnim** (redukton përpjekjet në zhvillimit).

modeli: dorëzimi Rritës/Inkremental(1)

- Dorëzimi rritës paraqet një **qasje** të zhvillimit të softuerit ku **disa nga funksionet** e zhvilluara i **dorëzohen klientit për përdorim**.
 - Në këtë proces **klienti identifikon funksionet** dhe i **radhit** ato në mënyrë që të zhvillohen **sipas radhës (prioriteteve)**.
 - Gjatë zhvillimit **çdo funksion** që **ndryshon** ndikon në **zhvillimin e funksionit tjetër** dhe në këtë mënyrë **i paraprin** ndryshimeve të funksioneve të **zhvilluara**.
- Çdo **version** posedon **funksionet paraprake** të **modifikuara** dhe **përshtatura**. Po ashtu ndryshimet e kërkuara përfshihen në funksionet **që janë në zhvillim ende** pa fillua zhvillimi

modeli: dorëzimi Rritës/Inkremental(2)



përparsit e modeli: dorëzimi Rritës/Inkremental(1)

- ❑ **Klientët** mund të përdorin *versionet e para si 'prototip'* dhe të *fitojnë eksperiencë* që mundeson zhvillimin e **mëtejshëm** të *kërkesës*.
 - **Ndryshe** nga prototipi *këto versione* janë pjese e *sistemit real* dhe për këtë arsye nuk ka *ri-mësim* pasi që sistemi ka filluar të **përdoret** me **versionet** fillestare.
- ❑ **Klientët nuk duhet** të presni derisa *i tërë sistemi* është *implementuar* para klientët mund të *përfitojnë vlerë* nga sistemi.
- ❑ **Verzioni i parë** përmbush *kërkesat e tyre më të rëndësishme* në mënyrë që ata mund të përdorin softuer menjëherë.
- ❑ **Rrezik më të ulët** të *dështimit* të projektit në përgjithësi.
- ❑ **Shërbimet** e sistemit me *prioritet më të lartë* tentohet të kenë testimin më të theksuar.



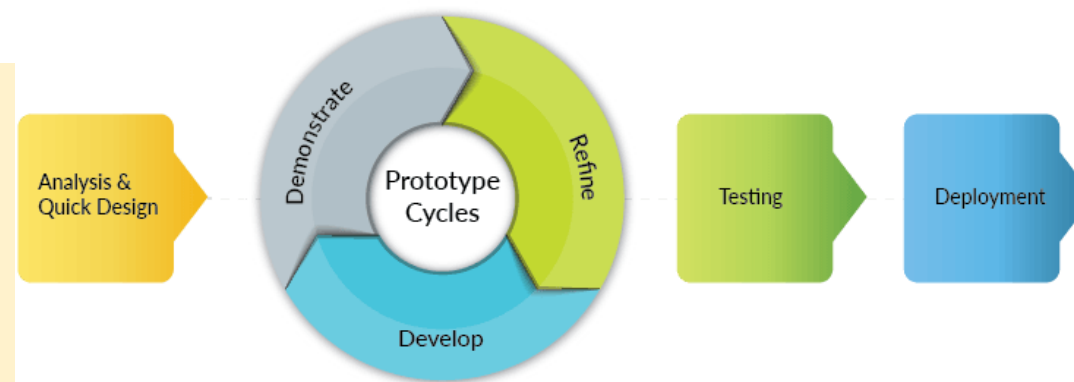
mangësit e modeli: dorëzimi Rritës/Inkremental(1)

- ☐ Meqë **kërkesat** nuk janë përcaktuar në detaje **përderisa një versionion** është implementuar , mund të jetë e vështirë të **identifikohen lehtësishtë nevojat** e përbashkëta që nevojiten nga të gjitha verzionet.
- ☐ **Thelbi i proceseve iterative** është se **specifikimi** është **zhvilluar** në **bashkëpunim** me **softuerin**.
- ☐ Megjithatë, kjo bie ndesh me **modelin e prokurimit** të shumë organizatave, ku **specifikimi i plotë** i sistemit është **pjesë e kontratës** së zhvillimit të sistemit.

Modeli RAD (Rapid Application Development)

- ❑ **Modeli i RAD** synon të ndërtojë aplikacione më të shpejta duke përdorur ***një proces zhvillimi përsëritës*** - përsëritjen e fazave të zhvillimit siç kërkohet, ku ekziston një divergjencë e zhvilluesve dhe konceptimeve të konsumatorëve.
- ❑ Modeli RAD ndjek metodat e implementimit të shpejtë ku karakteristikat/funksionet ekspozohen gradualisht dhe ndryshimet janë inkorporuar menjëherë.

❑ **Megjithëse projektet RAD** janë më të shpejta në zhvillim, mund të mos jetë e përshtatshme për ***projekte kritike me mision me mekanizma kompleksë*** dhe të dhëna të volumit ose nëse arkitektura teknike është e paqartë dhe mjetet e përdorura janë komplekse.

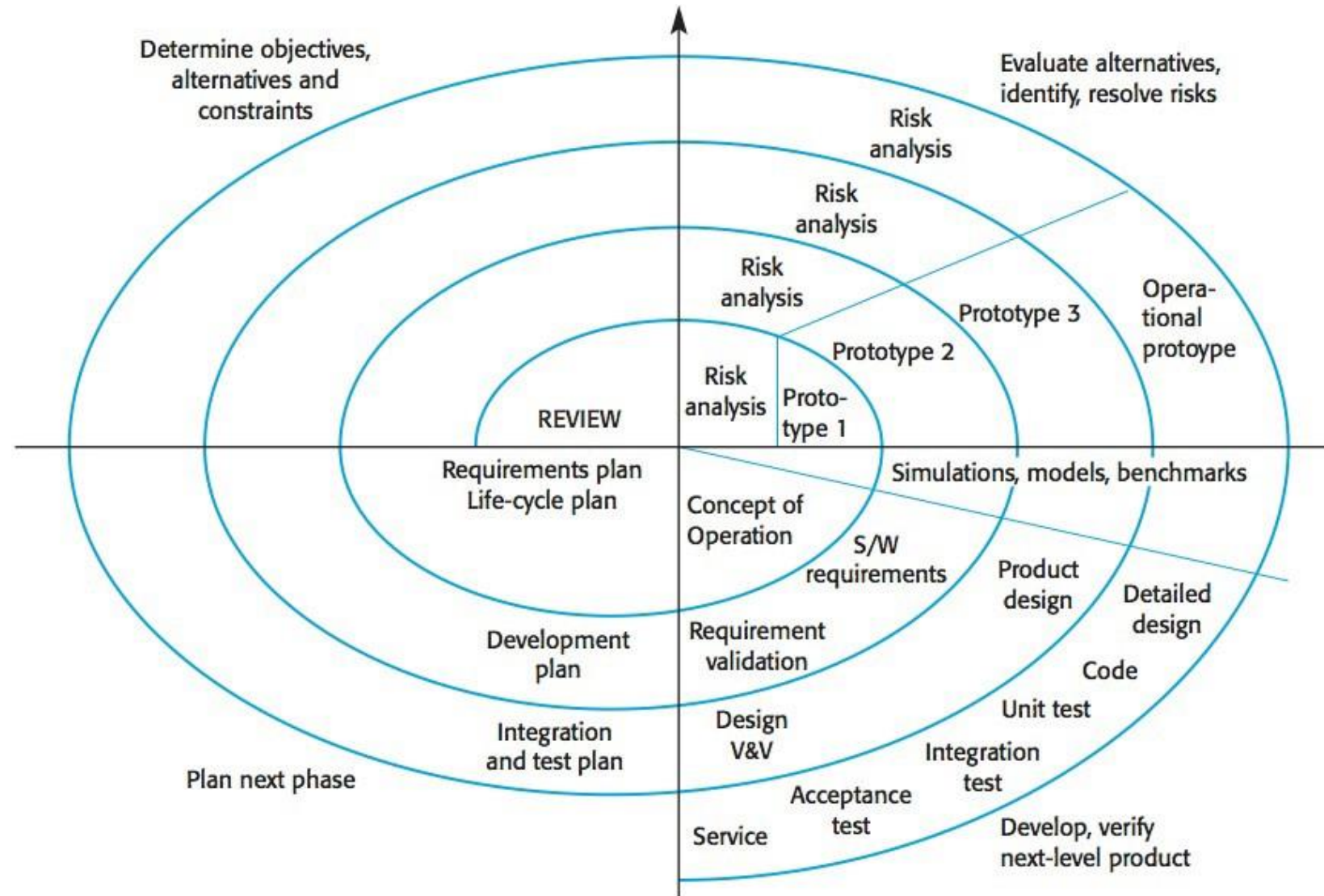


Modeli spiral i Boehm-it

- ❑ **Modeli i spiralës**, është një model i zhvillimit të sistemeve (SDLC) që përdoret në teknologjinë e informacionit (IT).
- ❑ Ky model i zhvillimit kombinon *natyre iterative të modelit prototipit* dhe modelit *ujëvarë*. **Modeli spiral** është i favorizuar për *projekte të mëdha*, të *shtrenjta* dhe të *ndërlikuara*.
- ❑ **Kur të përdorim** modelin spiral:
 - Kur vlersohet **kostoja** dhe **rrezikut** është i rëndësishëm.
 - Kur krijimi i **një prototipi** është i përshtatshëm
 - Kërkesat janë **komplekse**
 - Për projektet nga **rrisku mesëm** në të **lartë**
 - Priten **ndryshime** të **rëndësishme/të mëdha**
 - Përdoruesit janë të **pasigurt** për nevojat e tyre
- ❑ **Përparësitë** e modelit spirale:
 - **Feedback** të **hershme** dhe të shpeshta nga përdoruesit
 - Përdoruesit e **shohin sistemin herët** për shkak të të krijimit të prototipeve
 - Ofron tregues të hershëm të **rreziqeve**.
 - **Përdoruesit** mund të jetë i **lidhur ngushtë** me të gjitha hapat e ciklit të jetës

Modeli spiral i Boehm-it

Modeli **spiral** i Boehm-it



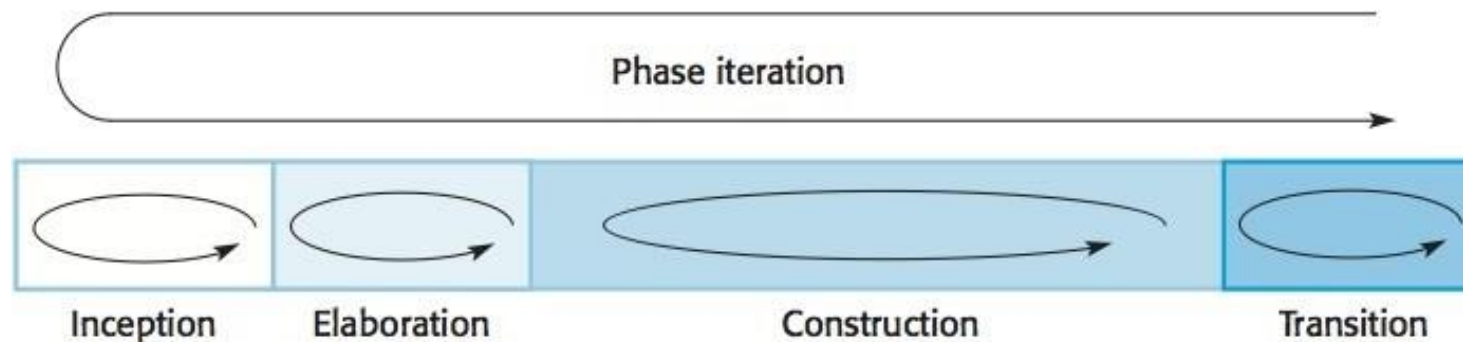


Modeli: RUP-Procesi i Unifikuar Racional (1)

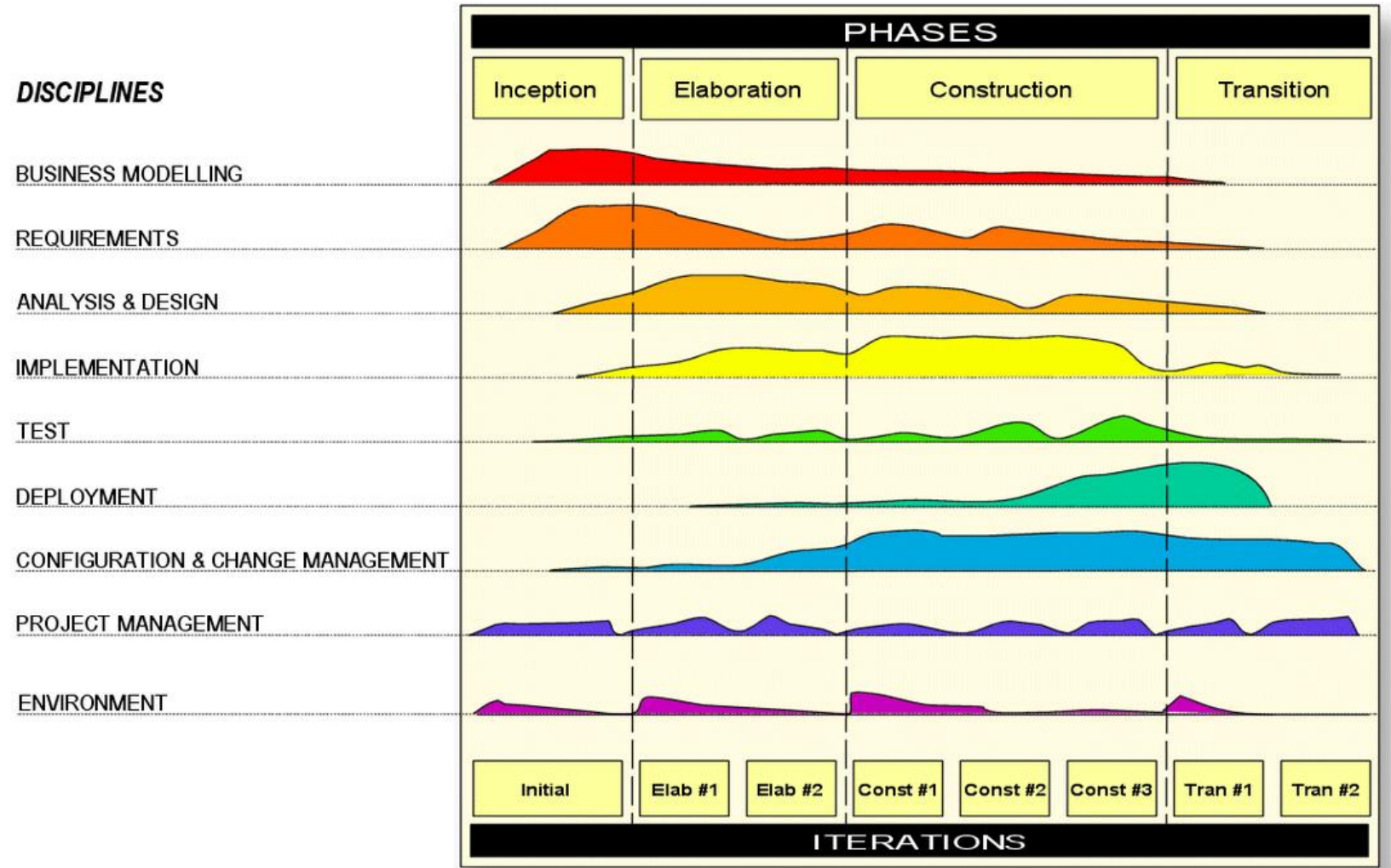
- ❑ Modeli Racional i Unifikuar (The Rational Unified Process - RUP))(Krutchen, 2003) është një shembull i **procesit modern**, i cili ka derivuar nga puna në **UML** dhe lidhet me Procesin e Unifikuar të Zhvillimit të Softuerit (Rumbaugh, et al., 1999; Arlow and Neustadt, 2005).
- ❑ Paraqet një **model hibrid** që sjellë elemente nga të gjitha **proceset gjenerale** duke përfshi praktikatat e mira të krijimit të **prototipit** dhe **dorëzimit rritës**
- ❑ RUP pranon që proceset **konvecionale** parqyrojnë procese nga një këndvështrim.
- ❑ RUP tenton që të **përshkruaj proceset** nga **tri perspektiva**:
 1. Një perspektivë **dinamike**, e cila tregon fazat e modelit me **kalimin e kohës**.
 2. Një perspektivë **statike**, e cila tregon **aktivitetet** e procesit që janë **miratuar**.
 3. Një perspektivë **praktike**, e cila **sugjeron praktika të mira** që do të përdoren gjatë procesit.

Fazat e modelit RUP

- 1) **Inception (Zanafilla)**, ka për qëllim të vendos *këkesat e biznesit* për sistemin. Gjatë kësaj faze *identifikohen entitetet e jashtme* (njerëzit dhe sistemet) që do të ndër veprojnë me sistemin që po e ndërtojmë
- 2) **Elaboration (Elaburimi)**, ka për qëllim të *kuptuarit e problemit*, *krijimin konceptual i arkitekturës* së sistemit dhe krijimin e *planit të zhvillimit*.
- 3) **Construction (Ndërtimi)**, është fazë ku *dizajnohet*, *programohet* dhe *testohet* sistemi
- 4) **Transition (Trazicioni)** është faza finale e **RUP** ku produkti *kalon* në *shfrytëzim* (deployed)



Fazat e modelit RUP





Çështjet me proceset e modeleve tradicionale

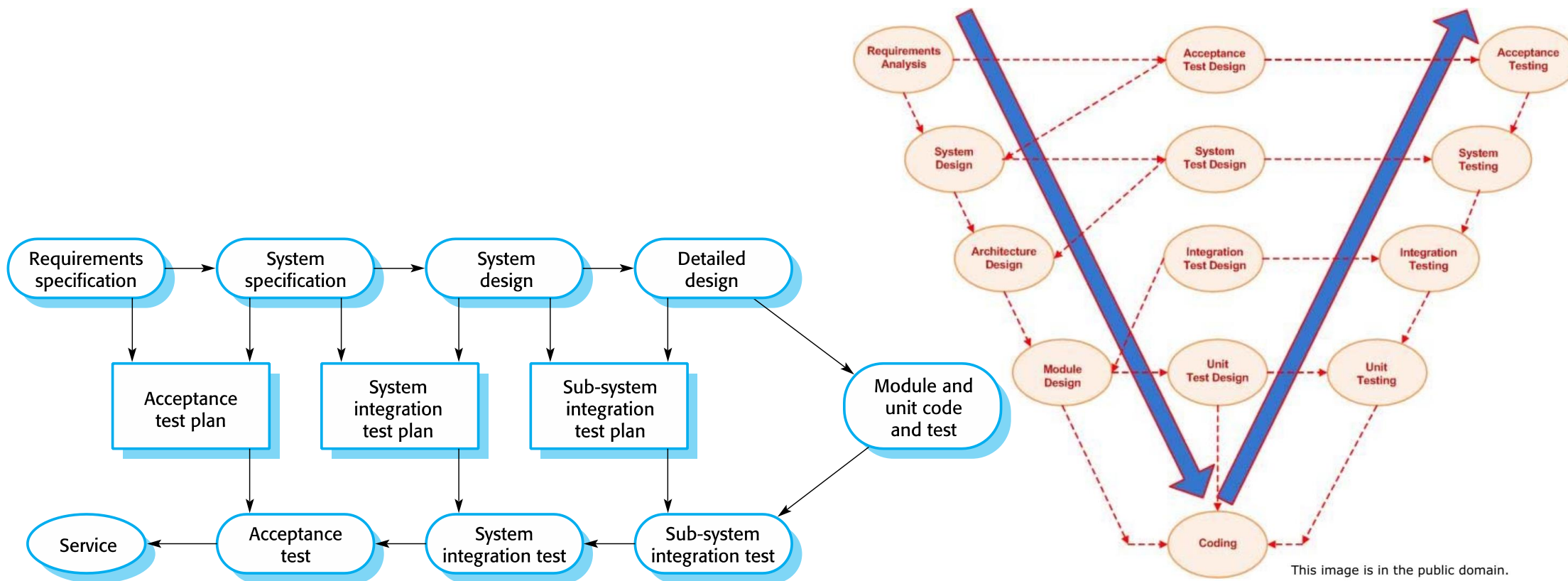
Issues With Traditional Processes

- ❑ **Sa kohë** keni **nevojë** për ta përfunduar këtë?
 - Vlerësimi i bazuar në **asnjë të dhënë**
 - Pyetje pa përgjigje
- ❑ Analizë → Dizajn → Implementim
 - analiza dhe dizajni **paraprakisht** kërkojnë **shumë kohë**
 - **jo kohë** të mjaftueshme për **implementim**
- ❑ "Unë dua që kjo **veçori** shtesë të shtohet!"
 - **nuk ka kohë** për të (këtë veçori)
 - **shkatërrojnë** dizajnin "perfect"

Zhvillimi i shpejtë (Rapid) i softuerit

- Zhvillimi dhe zbatimi i **shpejtë** tani është shpesh **kërkesa më e rëndësishme** për sistemet softuerike
 - Bizneset **operoj** në ndryshimet të **shpejtë/shëpershta të kërkesë** dhe është praktikisht e **pamundur** të prodhohet një grup i **kërkesave të qëndrueshme** të softuerit
 - Softueri duhet të **evluojë** shpejt për të **reflektuar** ndryshimet e nevojave të biznesit.
- Zhvillimi sipas **model i drejtuar-në-planifikim** është thelbësor për disa lloje të sistemit, por nuk i plotëson këto **nevoja të biznesit të sotshëm**
- Metodat e zhvillimit **Agile** u shfaqën në fund të viteve 1990, qëllimi i të cilave ishte **reduktimi rrënjësor** i **kohës së dorëzimit** për sistemet softuerike.

Fazat e testimit in a plan-driven software process (V-model)



This image is in the public domain.

tests developed in early phases, applied in later phases



Referencat

☐ Kapitulli 2: Software Processes

