

Universiteti AAB

Shkenca Kompjuterike

Inxhinieria e Softverit

**Hyrje në Inxhinierinë e Softverit dhe Modelet e
proceseve softverike**

Fatime Gashi



Hyrje në Inxhinierinë e Softverit

▶ **1. Çka është softveri?**

- ▶ Programet kompjuterike dhe dokumentacionet e asocuara. Produktet softverike mund të zhvillohen për ndonjë konsumator të caktuar apo për një treg të përgjithshëm.

▶ **2. Cilat janë atributet e softverit të mirë?**

- ▶ Softveri i mirë duhet ta sigurojë për përdoruesin funksionalitetin dhe performancën e kërkuar dhe duhet të jetë i mirëmbajtshëm, i varur dhe i përdorshëm.

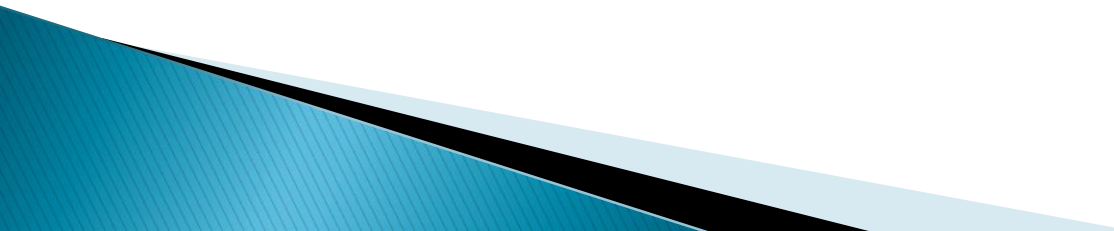
▶ **3. Çka është inxhinieria e softverit?**

- ▶ Inxhinieria e softverit është disiplinë e inxhinierisë që ka të bëjë me të gjitha aspektet e prodhimit të softverit.

Hyrje në Inxhinierinë e Softverit

- ▶ **4. Cilat janë aktivitetet fundamentale të inxhinierisë së softverit?**
 - ▶ Specifikacioni i softverit, zhvillimi i softverit, validimi i softverit dhe evoluimi i softverit.
- ▶ **5. Cili është dallimi ndërmjet inxhinierisë së softverit dhe shkencës kompjuterike?**
 - ▶ Shkenca kompjuterike përqëndrohet në teori dhe fundamente, kuse inxhinieria e softverit merret me praktikën e zhvillimit dhe dorëzimit të softverit.

Hyrje në Inxhinierinë e Softverit

- ▶ **6. Cili është dallimi ndërmjet inxhinierisë së softverit dhe inxhinierisë së sistemit?**
 - ▶ Inxhinieria e sistemit ka të bëjë me të gjitha aspektet e zhvillimit të sistemeve, të bazuara në kompjuter, duke e përfshirë inxhinierinë e hardverit, softverit dhe procesit. Inxhinieria e softverit është pjesë e këtij procesi më të përgjithshëm.
 - ▶ **7. Cilat janë sfidat kryesore me të cilat ballafaqohet inxhinieria e softverit?**
 - ▶ Përballja me diversitetin e shtuar, kërkesat për afate kohore të shkurtuara të dorëzimit, dhe zhvillimi i softverit të besueshëm.
- 

Hyrje në Inxhinierinë e Softverit

▶ **8. Cilat janë kostot e inxhinierisë së softverit?**

- ▶ Përafërsisht 60 % të shpenzimeve të softverit janë shpenzime të zhvillimit, 40 % janë shpenzime të testimeve.

▶ **9. Cilat janë teknikat dhe metodat më të mira të inxhinierisë së softverit?**

- ▶ Përderisa të gjitha projektet e softverit duhet të menaxhohen dhe zhvillohen profesionalisht, teknika të ndryshme janë të përshtatshme për lloje të ndryshme të sistemeve. P.sh. lojërat gjithmonë duhet të zhvillohen duke përdorur seri të prototipeve, kurse sistemet kontrolluese me rrezikshmëri kritike kërkojnë specifikacion të plotë dhe të analizueshëm. Prandaj nuk ekziston një metodë më e mirë se tjetra.

Hyrje në Inxhinierinë e Softverit

- ▶ **10. Çfarë ndryshimesh i ka shkaktuar Web-i inxhinierisë së softverit?**
- ▶ Web-i ka sjellë në dispozicion shumë shërbime softverike si dhe ka mundësuar zhvillimin e sistemeve të bazuara në-shërbime. Poashtu zhvillimi i sistemeve të bazuara në Web ka qar drejt avansimeve të rëndësishme në gjuhët programuese dhe ripërdorimin e softverit.

Hyrje në Inxhinierinë e Softverit

- ▶ **11. Cilat janë etiket e inxhinierisë së softverit?**
- ▶ **Konfidencialiteti** - Zakonisht duhet ta respektoni konfidencialitetin e punëtorëve dhe klientëve tuaj pa marrë parasysh se a është nënshkruar një marrëveshje formale e konfidencialitetit.
- ▶ **Kompetenca** - Ju nuk duhet ta keqprezentoni nivelin tuaj të kompetencave. Ju nuk duhet ta pranoni punën e cila është jashtë kompetencave tuaja.

Hyrje në Inxhinierinë e Softverit

(vazhdim)

- ▶ **Të drejtat e pronësisë intelektuale** - Ju duhet t'i njihni ligjet lokale për përdorimin e pronësisë intelektuale siç janë patentimet dhe të drejtat e autorit. Duhet të jeni të kujdesshëm që të siguroni mbrojtjen e pronësisë intelektuale të punëtorëve dhe klientëve.
- ▶ **Keqpërdorimi i kompjuterit** - Ju nuk duhet t'i përdorni shkathtësitë e juaja teknike për t'i keqpërdorur kompjuterët e njerëzve tjerë. Keqpërdorimi i kompjuterëve mund të jetë trivial (lojërat në kompjuter) apo shumë serioz (shpërndarja e virusëve).

Modelet e proceseve softverike

- ▶ Modeli i procesit softverik është reprezentim i thjeshtuar i procesit softverik. Secili model i procesit e paraqet procesin nga një perspektivë partikulare, prandaj jep vetëm informata të pjesërishme për atë proces. Për shembull modeli i aktiviteteve të procesit i tregon aktivitetet dhe sekuencën e tyre, por nuk i tregon rolet e njerëzve të involvuar në këto aktivitete.
- ▶ Këtu do t'i paraqesim **kater modele** të përgjithshme të proceseve, pra do të mund ta shohim *kornizën* e procesit por jo edhe detalet e aktiviteteve specifike. Këto modele nuk janë përshkrime përfundimtare të proceseve softverike, por janë abstragime të procesit të cilat mund të përdoren për të sqaruar qasje të ndryshme në zhvillimin e softverit.

Modelet e proceseve softverike

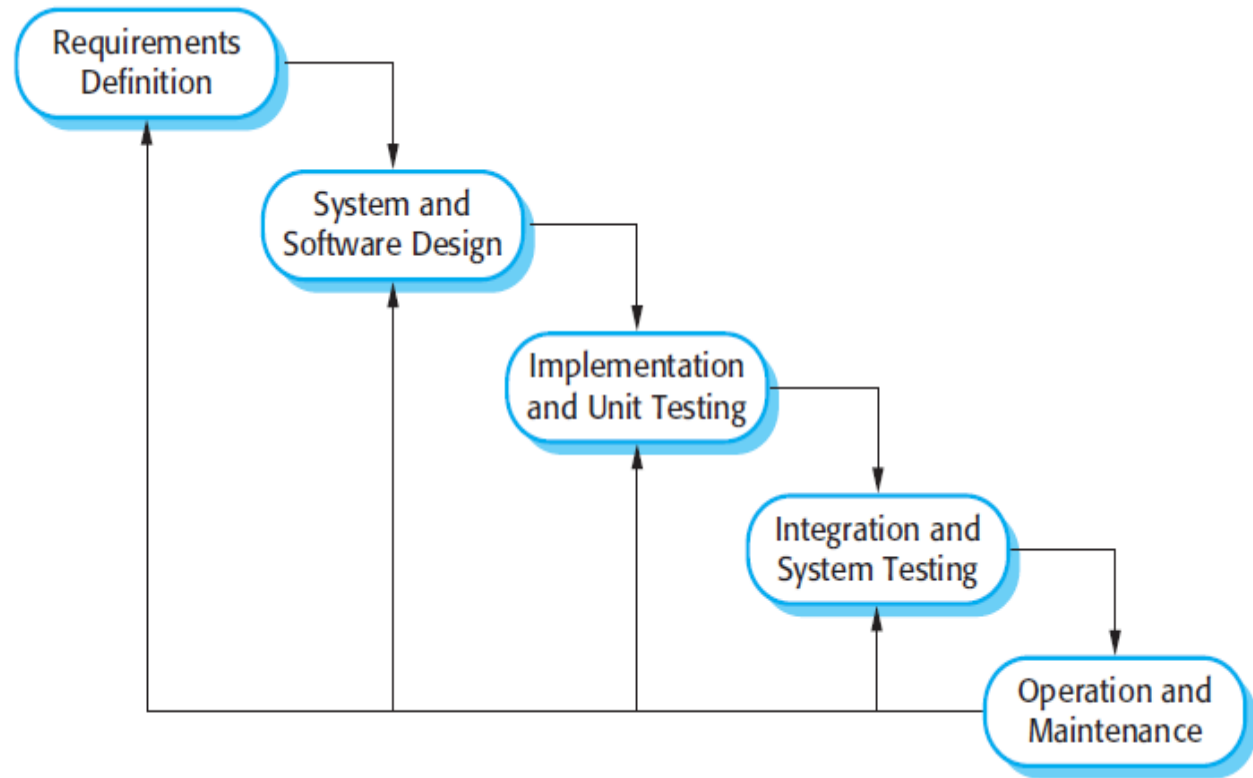
Modelet e proceseve që do t'i paraqesim janë:

- ▶ **1. Modeli Ujëvarë** Ky model i merr aktivitetet fundamentale të procesit për specifikim, zhvillim, validim dhe evoluim dhe i paraqet ato si faza të ndara të procesit, siç janë specifikimi i kërkesave, dizajni softverik, implementimi, testimi dhe kështu me rradhë.
- ▶ **2. Zhvillimi Inkremental** Kjo qasje i ndërlidh aktivitetet e specifikimit, zhvillimit dhe validimit. Sistemi zhvillohet si seri e versioneve (inkrementeve) dhe secili version i shton funksionalitet verzionit paraprak.

Modelet e proceseve softverike

- ▶ **3. Inxhinieria softverike e orientuar-në-ripërdorim** Kjo qasje bazohet në ekzistencën e një numri të madh të komponenteve të ripërdorshme. Dhe procesi i zhvillimit të sistemit përqëndrohet në integrimin e këtyre komponenteve në një system dhe jo t'i zhvilloj ato prej fillimit.
- ▶ **4. Modeli Spiral i Boehmit** – është model për ruajtje nga rreziqet
- ▶ Këto modele nuk e përjashtojnë njëra tjetrën dhe shpesh përdoren së bashku, sidomos për zhvillimin e sistemeve të mëdha. Për sisteme të mëdha kombinohen disa nga vetitë më të mira të modeleve ujëvarë dhe e zhvillimit inkremental. Pjesët e sistemit që kuptohen mirë mund të zhvillohen me modelin ujëvarë, kurse pjesët që janë të vështira për t'u kuptuar zhvillohen me modelin e zhvillimit inkremental.

Modeli ujëvarë



Modeli ujëvarë

- ▶ Fazat kryesore të modelit ujëvarë janë:
- ▶ 1. *Analiza dhe definicioni i kërkesave* Shërbimet, kufizimet dhe qëllimet e sistemit janë të vendosura në konsultim me përdoruesit e sistemit. Ato pastaj definohen në detale dhe shërbejnë si specifikacione të sistemit.

Modeli ujëvarë

- ▶ *2. Dizajnimi i sistemit dhe softverit* Dizajnimi i sistemit ia alokon kërkesat sistemeve hardverike apo softverike, duke e vendosur/krijuar arkitekturën e përgjithshme të sistemit. Dizajni i softverit përfshin identifikimin dhe përshkrimin e abstragimeve fundamentale të sistemit të softverit dhe lidhjeve të tyre.
- ▶ *3. Implementimi dhe testimi i njësisë* Gjatë kësaj faze dizajnimi i softverit realizohet si tërësi e programeve apo e njësive të programit. Me testimin e njësisë verifikohet se a i plotëson specifikacionet e saj secila njësi.

Modeli ujëvarë

- ▶ 4. *Integrimi dhe testimi i sistemit* Njësitë e programit apo programet integrohen dhe testohen si sistem i kompletuar që të sigurohet se kërkesat e softverit janë plotësuar. Dhe pas testimit, sistemi i softverit i dorëzohet konsumatorit.
- ▶ 5. *Operacioni dhe mirëmbajtja* Zakonisht kjo është faza më e gjatë e ciklit jetësor. Sistemi instalohet dhe vëhet në përdorim praktik. Mirëmbajtja përfshin korigjimin e gabimeve që nuk janë zbuluar në fazat e mëhershme të ciklit jetësor, duke e përmirësuar implementimin e njësive të sistemit dhe duke i avancuar shërbimet e sistemit.

Modeli ujëvarë

- ▶ Parimisht, si rezultat i secilës fazë përgatitet një dokument i cili miratohet. Dhe faza e ardhshme nuk duhet të fillojë para se të mbarojë ajo paraprake. Gjatë dizajnit paraqiten probleme me kërkesa. Gjatë kodimit paraqiten probleme me dizajn e kështu me rradhë. Pas disa përsëritjeve është normale që të stopohen pjesë të zhvillimit, siç është specifikacioni, dhe të vazhdohet me fazat e mëvonshme të zhvillimit.
- ▶ Do me thënë problemet lehen për rezolucion të mëvonshëm. Ky stopim i parakohshëm nënkupton se sistemi nuk do të funksionojë ashtu siç është pritur, e pastaj kjo mund të qojë deri te një sistem i strukturuar keq, dhe në fazën finale softveri veht në përdorim kur edhe zbulohen gabimet. Prandaj sistemi duhet të evoluojë që të mbetet i përdorshëm.

Modeli ujëvarë

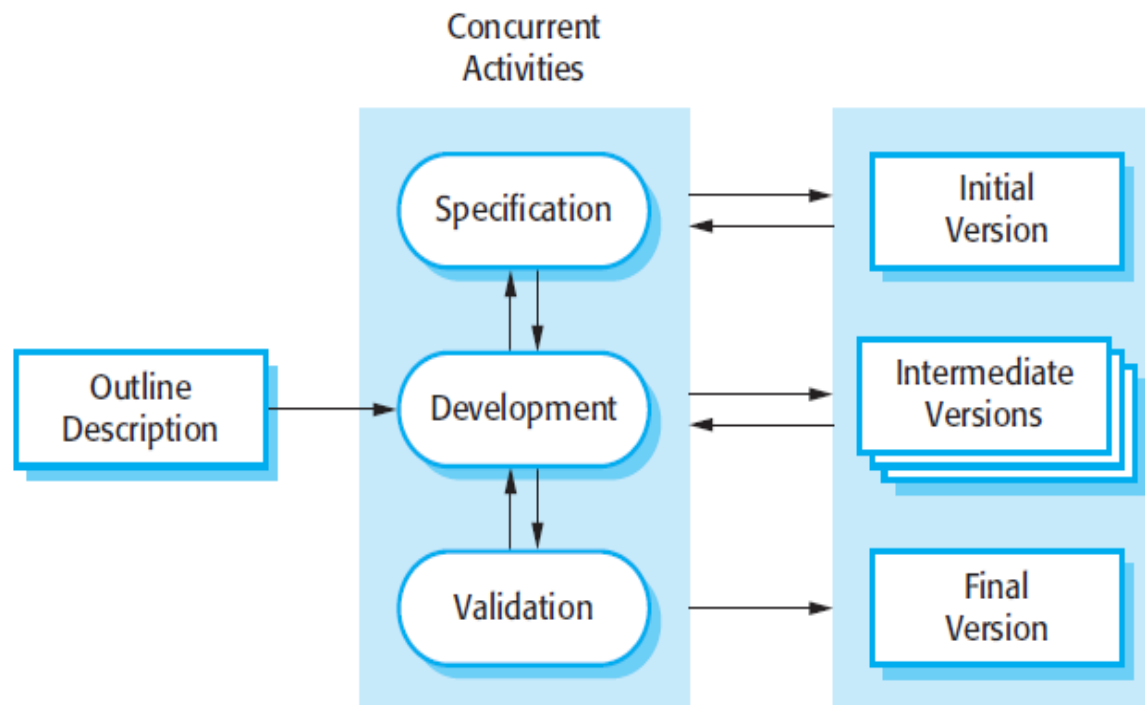
- ▶ Problem kryesor është particioni jofleksibil i projektit në faza të dalluara. Përkushtimet duhet të bëhen në faza të hershme gjatë procesit, dhe kjo e vështirëson përgjigjën ndaj kërkesave të ndryshueshme të konsumatorëve. Prandaj në parim modeli ujëvarë duhet të përdoret vetëm kur kërkesat të jenë kuptuar mirë dhe nuk ndryshojnë radikalisht gjatë zhvillimit të sistemit.

Modeli ujëvarë

Situatat ku modeli ujëvarë aplikohet më së shumti janë:

- ▶ Aplikacionet e drejtpërdrehta me rrezik të ulët – dobësia e modelit ujëvarë është se nuk siguron asnjë mekanizëm për t'u kthyer prapa në ndonjë fazë të mëhershme, nëse janë gjetur problemet. Përderisa në disa raste, kur aplikacionet kuptohen mirë, nuk paraqitet ndonjë problem.
- ▶ Stafi me eksperiencë – stafi me eksperiencë kanë më shumë gjasa t'i parashohin ndryshimet.
- ▶ Kërkesat e qarta – dmth kur nuk ekziston arsyeja për përmirësime, arsyeja më e shpeshtë për kthim prapa (në ndonjë fazë të mëhershme të procesit) janë kërkesat e pakompletuara apo të keqkuptuara.
- ▶ Kërkesat nuk kanë gjasa të ndryshojnë – si mëlartë
- ▶ Teknologjia stabile – kur përdoret hardveri, softveri apo teknikat më të zhvilluara (state of art) atëherë ka më shumë gjasa të ripunohet në fazat e mëhershme, pasi që teknologjia ndryshon.
- ▶ Nëse këto rrethana nuk zotërojnë, atëherë duhet të zgjedhet një model tjetër për zhvillim.

Zhvillimi inkremental



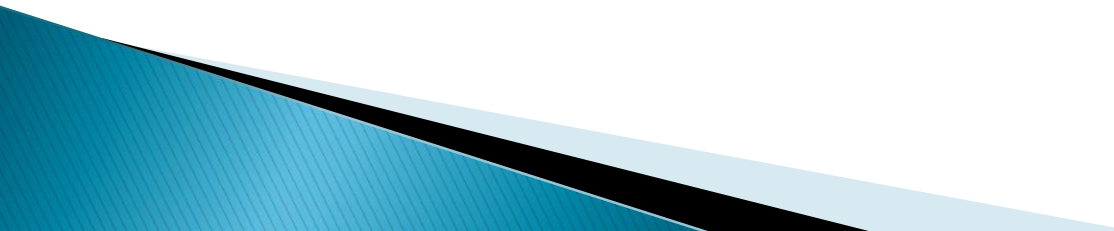
Zhvillimi inkremental

- ▶ Ky model bazohet në idenë që të zhvillohet një implementim fillestar dhe ky pastaj t'i ekspozohet komenteve të përdoruesve, dhe pastaj të evoluojë dhe të krijohen disa versione derisa të zhvillohet sistemi adekuat.
- ▶ Zhvillimi inkremental është model më i mirë se ai i ujëvarës për shumicën e sistemeve biznesore, e-komerc dhe personale. Zhvillimi inkremental e reflekton edhe mënyrën se si i zgjidhim problemet. Në rrallë herë e gjejmë tërë zgjidhjen e problemit paraprakisht, por shkojmë kah zgjidhja në disa hapa, duke u kthyer prapa, kur e kuptojmë se kemi gabuar.
- ▶ Duke e zhvilluar softverin në mënyrë inkrementale është më lirë dhe më lehtë të bëhen ndryshimet në softver, përderisa ende është duke u zhvilluar. Secili inkrement përfaqëson një funksion. Zakonisht inkrementet e para paraqesin funksion më të rëndësishëm apo të kërkuar më urgjentisht. Dhe kjo d.m.th. se konsumatori mund ta vlerësojë në fazë shumë të hershme se a ofron sistemi atë që kërkohet.

Zhvillimi inkremental

- ▶ Zhvillimi inkremental i ka tri përparësi në krahasim me modelin e ujëvarës:
- ▶ Kostoja e akomodimit të kërkesave të ndryshueshme të konsumatorëve është zvogëluar. Volumi i analizës dhe dokumentacionit, i cili duhet të bëhet përsëri, është shumë më i vogël se sa te modeli i ujëvarës.
- ▶ Është më lehtë të merret vlerësimi nga konsumatori rreth punës zhvilluese që është bërë. Konsumatorët mund të komentojnë rreth softverit dhe të shohin se në çfarë mase është implementuar.
- ▶ Me këtë model bëhet dorezim më rapid i softverit tek konsumatorët, edhe nëse nuk është përfshirë i tërë funksionaliteti. Konsumatorët mund ta përdorin softverin më shpejt se sa me modelin e ujëvarës.

Zhvillimi inkremental

- ▶ Nga perspektiva e menaxhmentit modeli inkremental i ka dy probleme:
 - ▶ Menaxherët duhet që rregullisht të marrin dokumentacione, që ta masin progresin e bërë dhe ky dokumentacion është i *kushtueshëm*.
 - ▶ Struktura e sistemit tenton të degradohet duke u shtuar inkrementët e ri.
- 

Studim i rastit (agjil dhe ujëvarë)

- ▶ Dy projekte shumë të ngjajshme janë ekzekutuar nga ekipi i njejtë i projektit, për të njejtin përdorues. Projekti i parë e ka përdorur metodologjinë waterfall dhe ka rezultuar në moskryerje të projektit në afatin e caktuar dhe dështim për t'i përmbushur kërkesat e përdoruesve. Projekti i dytë e ka përdorur metodën Agile, dhe përderisa ka pasur probleme fillestare projekti ka pasur sukses të madh.
- ▶ **Skenari**
- ▶ Bill Holst, president dhe inxhinier i i softverit për “Prescient Software Engineering”, i ka menaxhuar dy projekte për “Colorado Springs Utilities” – projekt elektrik dhe projektin për gaz. Të dy projektet kanë qenë sisteme për dizajnimin e distribuimit dhe shumë të ngjajshme.

Studim i rastit (agjil dhe ujëvarë)

- ▶ Holsti ka kontraktuar punën zhvillimore dhe e ka përdorur ekipin e njejtë për të dy projektet. Projekti elektrik ka qenë ofertë me çmim të fiksuar dhe është bërë duke e përdorur qasjen tradicionale ujëvarë. Projekti i gazit është bërë me ekipin e njejtë zhvillimor (ekip pa eksperiencë me metodën agjile). Në të dy rastet konsumatorët e projektit kanë qenë inxhinier fushor.
- ▶ Sipas qasjes ujëvarë kërkesat kompletohen para se të fillojë kodimi. Kryesisht pasi të përfundojë faza e kërkesave, përdoruesit nuk involvohen deri sa të përfundojë projekti. Përderisa me metodën Agjile përdoruesit vazhdojnë të jenë të involvuar gjatë tërë ciklit jetësor, me rishqyrtime dhe përditësime të rregullta ndaj kërkesave.

Studim i rastit (agjil dhe ujëvarë)

- ▶ **Përdorimi i metodës “ujëvarë” tregon rezultate jo të kënaqshme**
- ▶ Holsti ka vërejtur se projekti elektrik, i cili ishte bërë i pari, kishte shumë mangësi. Kishte shumë vonesa ndërmjet përfundimit të pritur dhe përfundimit aktual të projektit. Softveri nuk kryente atë që konsumatorët kishin kërkuar. Ka pasur shumë tundje gjatë projektit me rezultate zhgënjyese. Dhe frustrimi është ndier çdokund.
- ▶ **Reagimi fillestar “Ne nuk e njohim metodën agjile”**
- ▶ Ekipi e dinte se kishin nevojë të bënin diçka ndryshe, prandaj vendosën ta provojnë metodën agjile për projektin e gazit. Ata i punësuan disa trajnerë me ekspertizë në metodën agjile për t’i udhëzuar ata, por fillimisht kishte shumë rezistencë nga ekipi. Inxhinierëve fushor nuk u pëlqeu kjo, duke menduar se kjo metodologji ishte shumë e ndjeshme.

Studim i rastit (agjil dhe ujëvarë)

- ▶ Projekti kaloi nëpër shumë probleme. Ekipi e bëri një vendim të rreptë: Stop dhe rigrupohuni.
- ▶ Ky ndryshim i drejtimit u bë i mundur vetëm për shkak të metodës agjile. “Dhe çmimi për ta ndryshuar sistemin kaq vonë do të ishte shumë i lartë duke e përdorur modelin ujëvarë” ka thënë Holsti. Të pasurit e një modeli fleksibil për nga çmimi, bashkë me mundësinë për t’i ndryshuar kërkesat kah gjysma e projektit ka qenë çelësi kryesor i suksesit.

Studim i rastit (agjil dhe ujëvarë)

▶ **Rezultatet**

- ▶ Me kërkesat e rishqurtuara, ekipi filloi ta përdorë metodën agjile. Shpejtësia filloi të rritet pasi që dokumentimet filluan të zvogëlohen. Edhe me dy javë vonesë për rigrupim, projekti u përfundua herët dhe brenda bugjetit.
- ▶ Në të gjitha aspektet projekti u përfundua me sukses. Në krahasim me projektin e parë dhe nëse kjo kishte qenë garë, metoda agjile do të vlerësohej më shumë në të gjitha kategoritë.
- ▶ Cilësia ka qenë më e mirë dhe testimet kanë qenë më të qarta/pastra”. Përdorshmëria ishte shumë më e mirë, gjithashtu.
- ▶ Gjëja kryesore ka qenë se projekti agjil u realizua herët, brenda bugjetit dhe përdoruesve u pëlqeu.

Studim i rastit (agjil dhe ujëvarë)

- ▶ **Çelësi i suksesit**
- ▶ Holsti tha se arsyeja e suksesit kanë qenë dy faktorë kryesor: trajnimi agjil dhe rigrupimi. Trajnimi ka qenë me rëndësi sepse asnjëri nga ekipi nuk kishte punuar më parë me metodën agjile.
- ▶ **A do të ngadhnjejë çdoherë metoda agjile?**
- ▶ Edhe pse në këtë rast metoda agjile ishte fituese (në krahasim me metodën ujëvarë), nuk do me thënë se çdo projekt agjil do të ketë sukses apo se metoda agjile është metodë më e mirë. Edhe projekti agjil filloi me shumë vështirësi, por aftësia për inspektim dhe adaptim, duke marrur kohë për rigrupim dhe për t'i rishqyrtuar kërkesat, ua mundësoi këtij ekipi të kthehet “në binarë”.

Inxhinieria softverike e orientuar në ripërdorim

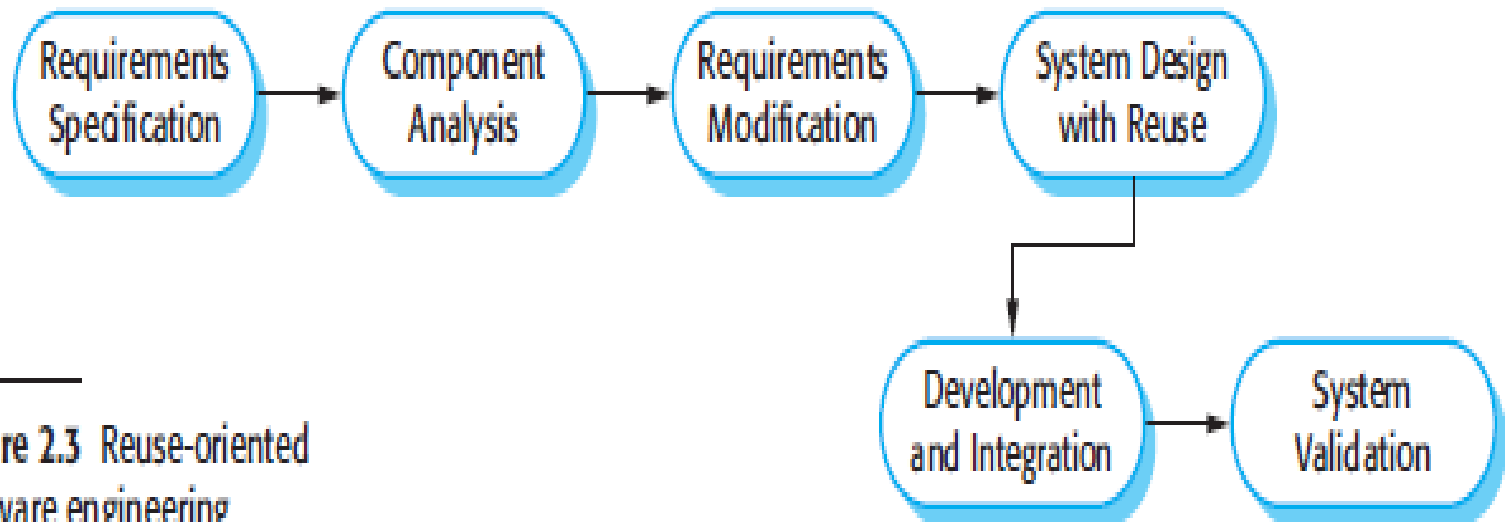


Figure 2.3 Reuse-oriented software engineering

Inxhinieria softverike e orientuar në ripërdorim

- ▶ Në shumicën e projekteve softverike, ka ripërdorim të softverit. Kjo zakonisht ndodh joformalisht kur njerëzit që punojnë në projekt dijnë dizajne apo kode që janë të ngjajshme me ato që kërkohen. Proceset e zhvillimit softverik që përqëndrohen në ripërdorimin e softverit ekzistues janë duke u përdorur shumë.
- ▶ Edhe pse fazat fillestare si specifikimi i kërkesave dhe validimi janë të krahasueshme me modelet e tjera, megjithatë fazat tjera të këtij modeli janë të ndryshme.

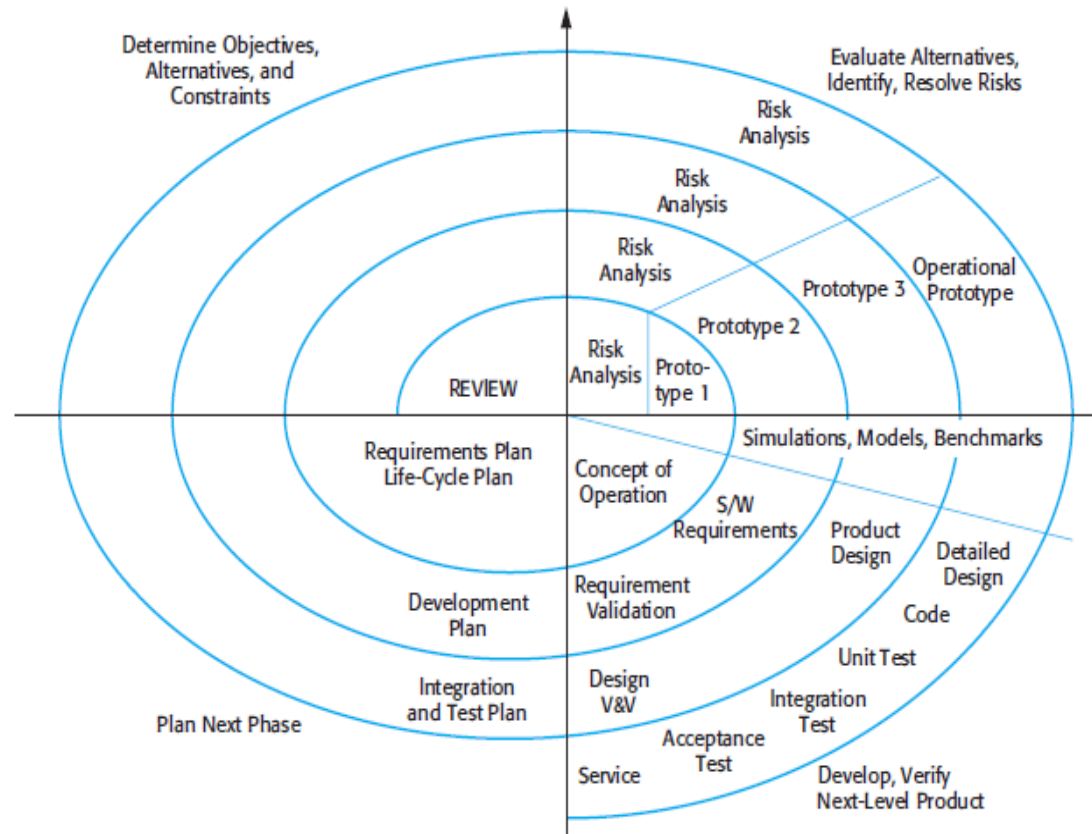
Inxhinieria softverike e orientuar në ripërdorim

- ▶ Këto faza janë:
- ▶ **Analiza e komponenteve** – Kur dihen specifikimet e kërkesave atëherë kërkohen komponentet për t'i implementuar ato specifikime. Zakonisht nuk ekziston përputhshmëri e plotë dhe komponentet që mund të përdoren e ofrojnë vetëm një pjesë të funksionalitetit.
- ▶ **Modifikimi i kërkesave** – Gjatë kësaj faze analizohen kërkesat dhe pastaj modifikohen varësisht nga komponente që janë në dispozicion.
- ▶ **Dizajni i sistemeve me ripërdorim** – Gjatë kësaj faze dizajnohet korniza e sistemit apo ripërdoret korniza ekzistuese. Dizajnerët i marrin në konsiderim komponentet që ripërdoren dhe e organizojnë kornizën.
- ▶ **Zhvillimi dhe integrimi** Zhvillohet software-i që nuk mund të blehet në mënyrë eksterne, dhe komponentet dhe sistemet komerciale integrohen për ta krijuar një sistem të ri dhe integrimi i sistemit mund të jetë pjesë e procesit zhvillues e jo aktivitet i ndarë.

Inxhinieria softverike e orientuar në ripërdorim

- ▶ Ekzistojnë tri lloje të komponenteve të softverit që mund të përdoren gjatë procesit të orientuar-në-ripërdorim.
- ▶ **1.** Web shërbimet që zhvillohen duke u bazuar në standardet e shërbimeve.
- ▶ **2.** Koleksioni i objekteve që zhvillohen si tërësi për tu integruar me kornizën e komponenteve siq është .NET apo J2EE.
- ▶ **3.** Sistemet softverike “stand-alone” konfigurohen për t’u përdorur në ambiente të veçanta.
- ▶ Inxhinieria softverike e orientuar në ripërdorim e ka aftësinë e reduktimit të vëllimit të softverit që duhet të zhvillohet, dhe i zvogëlon koston dhe rreziqet. Ajo poashtu e mundëson përfundimin më të shpejtë të softverit, mirëpo edhe kompromiset e kërkesave janë të pashmangshme dhe kjo mund të qoj deri te sistemi që nuk i plotëson kërkesat e vërteta të përdoruesve.

Modeli spiral i Boehm-it



Modeli spiral i Boehm-it

- ▶ Modeli spiral është propozuar nga Boehm (1988). Ky model është paraqitur në figurë. Procesi softverik është paraqitur në formë të spiralës, e jo si sekuencë e aktiviteteve me pak kthim prapa nga njëri aktivitet në tjetrin. Secila unazë në spirale e paraqet një fazë të procesit softverik. Prandaj unaza më e brendshme mund të paraqesë fizibilitetin, e ardhshmja përcaktimin e kërkesave, pastaj dizajnin e kështu me rradhë.
- ▶ Ky model e kombinon **shmangien nga gabimet** me **tolerancën ndaj gabimeve**. Modeli spiral supozon se ndryshimet rezultojnë nga rreziqet e projektit prandaj përfshin aktivitete eksplicite për menaxhimin e rreziqeve për t'i zvogëluar këto rreziqe.

Modeli spiral i Boehm-it

- ▶ Secila unazë në spirale është e ndarë në katër sektore:
- ▶ 1. **Settings-at e objektivave** Këtu përcaktohen objektivat, kufizimet në proceset dhe në produkte, identifikohen rreziqet e projektit, mund të planifikohen strategji alternative, varësisht nga këto rreziqe.
- ▶ 2. **Vlerësimi dhe reduktimi i rreziqeve** Për çdo rrezik të identifikuar të projektit kryhet një analizë e detalizuar. Pastaj ndërmerren hapa për ta zvogëluar rrezikun.
- ▶ 3. **Zhvillimi dhe validimi** Pas evaluimit të rrezikut, zgjedhet një model i zhvillimit për sistem.
- ▶ 4. **Planifikimi** Projekti shqyrtohet dhe merret vendimi se a të vazhdohet me unazën e ardhshme të spiralës. Nëse po atëherë hartohen planet për fazën e ardhshme të projektit.

Modeli spiral i Boehm-it

- ▶ Dallimi kryesor ndërmjet modelit spiral dhe modeleve tjera është njohja eksplicite e rreziqeve. Jo formalisht me rrezik nënkuptohet diçka që mund të shkojë gabim. Për shembull nëse tentohet të përdoret një gjuhë e re programuese atëherë rreziku është se kompajlerët në dispozicion nuk janë të besueshëm apo nuk prodhojnë me efikasitet të mjaftueshëm kod të objekteve.
- ▶ Rreziqet qojnë deri te ndryshimet e propozuara të softverit dhe problemet e projektit siç janë tejkalimi i planprogramit dhe i kostos, prandaj minimizimi i rrezikut është aktivitet shumë i rëndësishëm i menaxhimit të projektit.