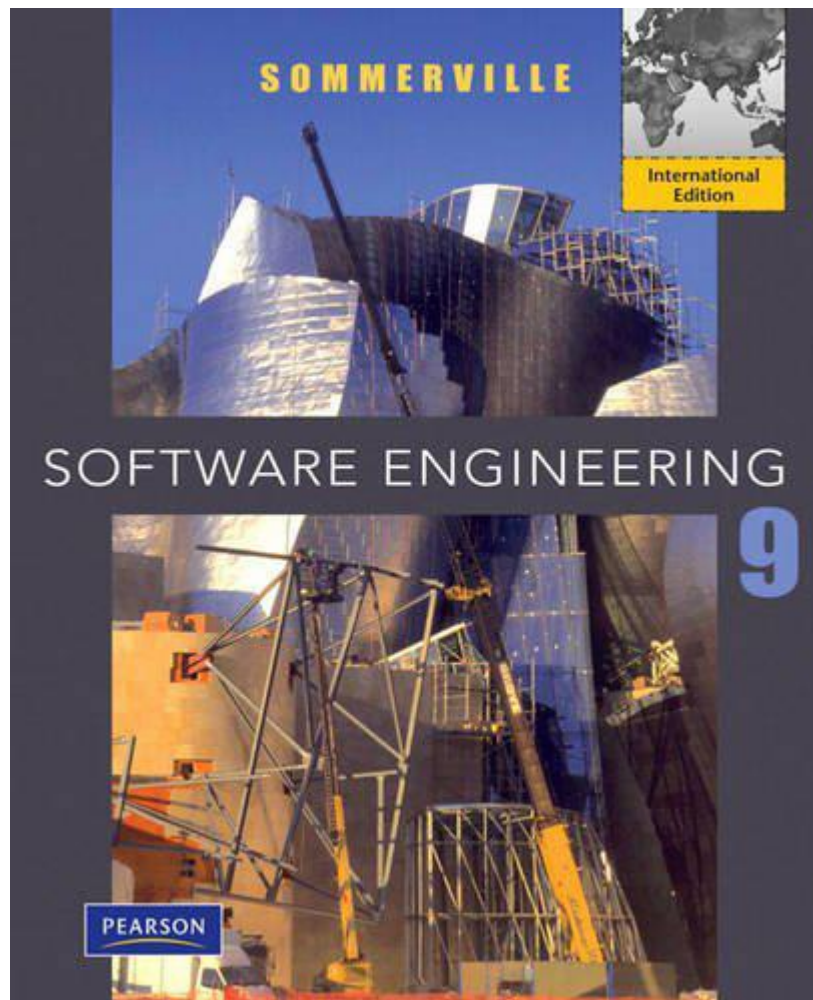


Ian Sommerville

Inxhinierimi i softuerit

përmbledhje



Zgjodhi dhe përktheu

Ridvan Bunjaku

Softa Shqiptare & |:(|:)

Gusht 2012

Përmbajtja

PJESA I: HYRJE NË INXHINIERIM TË SOFTUERIT	4
1. Hyrje	4
2. Proceset softuerike	5
3. Zhvillimi agjil (i shkathtë) i softuerit.....	6
4. Inxhinierimi i kërkesave	7
5. Modelimi i sistemit.....	8
6. Projektimi i arkitekturës.....	9
7. Projektimi (dizajnimi) dhe implementimi	10
8. Testimi i softuerit	11
9. Evolucion i softuerit	12
PJESA II: BESUESHMËRIA DHE SIGURIA.....	13
10. Sistemet socioteknike	13
11. Besueshmëria (mbështetshmëria) dhe siguria e informatave.....	14
12. Besueshmëria dhe specifikimi i sigurisë.....	15
13. Inxhinierimi i besueshmërisë	16
14. Inxhinierimi i sigurisë së informatave	17
15. Besueshmëria dhe garantimi i sigurisë	18
PJESA 3: INXHINIERIMI I AVANSUAR I SOFTUERIT.....	19
16. Ripërdorja e softuerit.....	19
17. Inxhinierimi softuerik i bazuar në komponenta.....	20
18. Inxhinierimi i softuerit të shpërndarë	21
19. Arkitektura e orientuar kah shërbimet (SOA)	22
20. Softueri i integruar në harduer (embedded)	23
21. Inxhinierimi i softuerit i orientuar kah aspektet	24

PJESA 4: MENAXHIMI I SOFTUERIT	25
22. Menaxhimi i projektit.....	25
23. Planifikimi i projektit	26
24. Menaxhimi i cilësisë	27
25. Menaxhimi i konfigurimit.....	28
26. Përmirësimi i procesit	29
Online	29

PJESA I: HYRJE NË INXHINIERIM TË SOFTUERIT

1. Hyrje

- Inxhinierimi i softuerit është disiplinë inxhinierike që merret me të gjitha aspektet e prodhimit të softuerit.
- Softueri nuk është vetëm program apo programe por e përfshin edhe dokumentimin. Atributet thelbësore të produktit softuerik janë mirëmbajtshmëria, besueshmëria, siguria, efikasiteti, dhe pranueshmëria.
- Procesi softuerik i përfshin të gjitha aktivitetet e përfshira në zhvillimin e softuerit. Aktivitetet e nivelit të lartë si specifikimi, zhvillimi, validimi, dhe evolucioni janë pjesë e të gjitha proceseve softuerike.
- Nocionet themelore të inxhinierimit të softuerit janë të zbatueshme universalisht për të gjitha llojet e zhvillimit të softuerit. Këto themele i përfshijnë proceset softuerike, besueshmërinë, sigurinë, kërkesat, dhe ripërdorjen.
- Ka disa lloje të ndryshme të sistemeve dhe secili kërkon vegla dhe teknika të duhura të inxhinierimit të softuerit për zhvillimin e tyre. Ka pak, nëse ka, teknika specifike të projektimit dhe të implementimit që janë të zbatueshme për të gjitha llojet e sistemeve.
- Idetë themelore të inxhinierimit të softuerit janë të zbatueshme për të gjitha llojet e sistemeve softuerike. Këto themele i përfshijnë proceset e menaxhuara të softuerit, besueshmërinë dhe sigurinë e softuerit, inxhinierimin e kërkesave dhe ripërdorjen e softuerit.
- Inxhinierët softuerikë kanë përgjegjësi ndaj profesionit të inxhinierimit dhe shoqërisë. Ata nuk duhet të merren thjesht vetëm me çështjet teknike.
- Shoqëritë profesionale publikojnë kode të mirësjelljes që i caktojnë standardet e sjelljes që priten nga anëtarët e tyre.

2. Proceset softuerike

- Proceset softuerike janë aktivitetet e përfshira në prodhimin e një sistemi softuerik. Modelet e procesit të softuerit janë paraqitje abstrakte të këtyre proceseve.
- Modelet e përgjithshme të procesit e përshkruajnë organizimin e proceseve softuerike. Shembuj të këtyre modeleve të përgjithshme përfshijnë modelin ujëvarë, zhvillimin rritës (incremental), dhe zhvillimin e orientuar kah ripërdorja.
- Inxhinierimi i kërkesave është procesi i zhvillimit të një specifikacioni të softuerit. Specifikacionet synohen t'ua komunikojnë zhvilluesve të sistemit nevojat sistemore të klientit.
- Proceset e projektimit dhe të implementimit merren me transformimin e një specifikacioni të kërkesave në sistem të ekzekutueshëm softuerik. Metodatat sistematike të projektimit mund të përdoren si pjesë e këtij transformimi.
- Validimi i softuerit është procesi i kontrollimit se sistemi përputhet me specifikacionin e tij dhe se i plotëson nevojat reale të përdoruesve të sistemit.
- Evolucionin softuerik ndodh kur i ndryshoni sistemet ekzistuese softuerike për t'i plotësuar kërkesat e reja. Ndryshimet janë të vazhdueshme dhe softueri duhet të evoluojë për të mbetur i dobishëm.
- Proceset duhet të përfshijnë aktivitete për t'ia dalë me ndryshimin. Kjo mund ta përfshijë një fazë të prototipimit që ndihmon të shmangen vendimet e dobëta në kërkesa dhe në projektim. Proceset mund të strukturohen për zhvillim dhe dorëzim iterativ ashtu që ndryshimet mund të bëhen pa e prishur sistemin si tërësi.
- Rational Unified Process është proces bashkëkohor i përgjithshëm që është i organizuar në faza (inception-zanafilla, elaboration-zhvillimi, construction-konstruktimi, dhe transition-tranzicioni) por i ndan aktivitetet (kërkesat, analizën dhe projektimin-dizajnin, etj.) nga këto faza.

3. Zhvillimi agjil (i shkathtë) i softuerit

- Metodatat agjile janë metoda rritëse të zhvillimit që përqendrohen në zhvillim të shpejtë, versione të reja të shpeshta të softuerit, zvogëlim të ngarkesës së procesit, dhe prodhim të kodit me cilësi të lartë. Ato e përfshijnë drejtpërdrejt klientin në procesin e zhvillimit.
- Vendimi se a të përdoret qasje agjile apo me plan ndaj zhvillimit duhet të varet nga lloji i softuerit që po zhvillohet, kapacitetet e ekipit zhvillues, dhe kultura e kompanisë që e zhvillon sistemin.
- Programimi ekstrem (extreme programming) është metodë e mirënjohur agjile që e integron një grup të praktikave të mira të programimit si versionet e shpeshta të reja të softuerit, përmirësimin e vazhdueshëm të softuerit, dhe pjesëmarrjen e klientëve në ekipin e zhvillimit.
- Një forcë e veçantë e programimit ekstrem është zhvillimi i testeve të automatizuara para se të krijohet një veçori e programit. Të gjitha testet duhet të ekzekutohen me sukses kur të integrohet një increment (iteracion) në sistem.
- Metoda Scrum është metodë agjile që ofron kornizë të menaxhimit të projektit. Ajo përqendrohet rreth një grupi të sprintave, që janë periudha me kohë fikse kur zhvillohet një increment (iteracion). Planifikimi bazohet në prioritizimin e një liste të punëve të pabëra (backlog) dhe zgjedhjen e punëve me prioritetin më të lartë për një sprint.
- Shkallëzimi i metodave agjile për sisteme të mëdha është i vështirë. Sistemet e mëdha kanë nevojë për projektim dhe dokumentim që në fillim. Integrimi i vazhdueshëm është praktikisht i pamundur kur ka disa ekipe të ndara të zhvillimit që punojnë në një projekt.

4. Inxhinierimi i kërkesave

- Kërkesat për një sistem softuerik e caktojnë se çka duhet të bëjë sistemi dhe i definojnë kufizimet që i ka operimi dhe implementimi i tij.
- Kërkesat funksionale janë deklarime të shërbimeve që duhet t'i ofrojë sistemi apo janë përshkrime se si duhet të bëhen disa llogaritje.
- Kërkesat jofunksionale shpesh e kufizojnë sistemin që po zhvillohet dhe procesin e zhvillimit që përdoret. Këto mund të jenë kërkesa të produktit, kërkesa organizacionale, apo kërkesa të jashtme. Ato shpesh ndërliken me vetitë emergjente të sistemit dhe prandaj vlejnë për sistemin si tërësi.
- Dokumenti i kërkesave softuerike është një deklaram i dakorduar i kërkesave të sistemit. Ai duhet të organizohet ashtu që të mund ta përdorin edhe klientët e sistemit edhe zhvilluesit e softuerit.
- Procesi i inxhinierimit të kërkesave e përfshin një studim të fizibilitetit (a realizohet dhe a ia vlen), nxjerrjen dhe analizën e kërkesave, specifikimin e kërkesave, validimin e kërkesave, dhe menaxhimin e kërkesave.
- Nxjerrja dhe analiza e kërkesave është proces iterativ që mund të paraqitet si spirale e aktiviteteve - zbulimi i kërkesave, klasifikimi dhe organizimi i kërkesave, negociimi i kërkesave, dhe dokumentimi i kërkesave.
- Validimi i kërkesave është procesi i kontrollimit të kërkesave për validitet, konsistencë, kompletshmëri, realizëm, dhe verifikueshmëri.
- Ndryshimet biznesore, organizacionale, dhe teknike pashmangshëm çojnë në ndryshim të kërkesave për një sistem softuerik. Menaxhimi i kërkesave është procesi i menaxhimit dhe i kontrollimit të këtyre ndryshimeve.

5. Modelimi i sistemit

- Modeli është pamje abstrakte e një sistemi që i injoron disa detale të sistemit. Mund të zhvillohen modele komplementare të sistemit për ta treguar kontekstin, ndërveprimet, strukturën, dhe sjelljen e sistemit.
- Modelet kontekstuale e tregojnë se si një sistem që po modelohet është i pozicionuar në një ambient me sistemet dhe proceset tjera. Ato ndihmojnë të definohen kufijtë e sistemit që do të zhvillohet.
- Diagramet e rasteve të përdorimit dhe diagramet sekuenciale përdoren për t'i përshkruar ndërveprimet ndërmjet sistemit që po projektohet dhe përdoruesve/sistemeve tjera. Rastet e përdorimit i përshkruajnë ndërveprimet ndërmjet një sistemi dhe aktorëve të jashtëm; diagramet sekuenciale iu shtojnë atyre më shumë informata duke i treguar ndërveprimet ndërmjet objekteve të sistemit.
- Modelet strukturale e tregojnë organizimin dhe arkitekturën e një sistemi. Diagramet e klasave përdoren për ta definuar strukturën statike të klasave në një sistem dhe asociacionet e tyre.
- Modelet e sjelljes përdoren për ta përshkruar sjelljen dinamike të një sistemi që ekzekutohet. Kjo mund të modelohet nga perspektiva e shënimeve që procesohen nga sistemi apo nga ngjarjet që i stimulojnë përgjigjet nga një sistem.
- Diagramet e aktivitetit mund të përdoren për ta modeluar procesimin e shënimeve, ku secili aktivitet e paraqet një hap të procesit.
- Diagramet e gjendjes përdoren për ta modeluar sjelljen e një sistemi si përgjigje ndaj ngjarjeve të brendshme apo të jashtme.
- Inxhinierimi i udhëhequr nga modelet është qasje ndaj zhvillimit softuerik në të cilën një sistem paraqitet si grup i modeleve që mund të transformohen automatikisht në kod të ekzekutueshëm.

6. Projektimi i arkitekturës

- Arkitektura softuerike është përshkrim se si është organizuar një sistem softuerik. Vetitë e një sistemi si performansa, siguria, dhe disponueshmëria ndikohen nga arkitektura e përdorur.
- Vendimet e projektimit të arkitekturës përfshijnë vendime për llojin e aplikacionit, shpërndarjen e sistemit, stilet arkitekturale që do të përdoren, dhe mënyrat në të cilat duhet të dokumentohet dhe vlerësohet (evaluohet) arkitektura.
- Arkitekturat mund të dokumentohen nga disa perspektiva apo pamje të ndryshme. Pamjet e mundshme përfshijnë pamjen konceptuale, pamjen logjike, pamjen e procesit, pamjen e zhvillimit, dhe pamjen fizike.
- Mostrat (patterns) arkitekturale janë mjet për ripërdorjen e dijes për arkitekturat e përgjithshme të sistemeve. Ato e përshkruajnë arkitekturën, e shpjegojnë se kur mund të përdoren, dhe i diskutojnë përparësitë dhe dobësitë e tyre.
- Mostrat arkitekturale të përdorura më shpesh përfshijnë Model-View-Controller (Modeli-Pamja-Kontrolluesi), Layered Architecture (Arkitektura e Shtresëzuar), Repository (Depoja), Client-server (Klient-shërbyes), dhe Pipe and Filter (Fut në gyp dhe filtro).
- Modelet e përgjithshme të arkitekturave të sistemeve aplikative na ndihmojnë ta kuptojmë operimin e aplikacioneve, t'i krahasojmë aplikacionet e llojit të njëjtë, t'i validojmë projektimet e sistemeve aplikative, dhe t'i vlerësojmë komponentat me shkallë të madhe për ripërdorje.
- Sistemet e procesimit të transaksioneve janë sisteme ndërvepruese që lejojnë që informata në një bazë të shënimeve të qaset nga distanca dhe të ndryshohet nga një numër i caktuar i përdoruesve. Sistemet informative dhe sistemet e menaxhimit të resurseve janë shembuj të sistemeve të procesimit të transaksioneve.
- Sistemet e procesimit të gjuhës përdoren për t'i përkthyer tekstet nga një gjuhë në një tjetër dhe për t'i kryer instruksionet e specifikuar në gjuhën hyrëse. Ato e përfshijnë një përkthyes dhe një makinë abstrakte që e ekzekuton gjuhën e gjeneruar.

7. Projektimi (dizajnimi) dhe implementimi

- Projektimi dhe implementimi i softuerit janë aktivitete të gërshetuara. Niveli i detaleve në projektim varet nga lloji i sistemit që po zhvillohet dhe a jeni duke përdorur qasje të udhëhequr nga plani apo agjile.
- Procesi i projektimit të orientuar kah objektet i përfshin aktivitetet për ta projektuar arkitekturën e sistemit, të identifikohen objektet në sistem, të përshkruhet projektimi duke përdorur modele të ndryshme të objekteve, dhe të dokumentohen interfejsat e komponentave.
- Gjatë një procesi të projektimit të orientuar kah objektet mund të prodhohet një grup i modeleve të ndryshme. Këto i përfshijnë modelet statike (modelet e klasave, modelet e përgjithësimit, modelet e asociimit) dhe modelet dinamike (modelet sekuenciale, modelet e makinës me gjendje).
- Interfejsat e komponentave duhet të definohen saktësisht ashtu që objektet tjera të mund t'i përdorin ato. Për t'i definuar interfejsat mund të përdoret një stereotip UML për interfejs.
- Kur zhvillohet softuer, duhet gjithmonë ta konsideroni mundësinë e ripërdorjes së softuerit ekzistues, si komponenta, si shërbime, apo si sisteme të plota.
- Menaxhimi i konfigurimit është procesi i menaxhimit të ndryshimeve në një sistem softuerik evolues. Është thelbësor kur një ekip i njerëzve janë duke bashkëpunuar për të zhvilluar softuer.
- Shumica e zhvillimit të softuerit është zhvillim nikoqir-cak (host-target). E përdorni një IDE (Rrethinë Zhvilluese të Integruar) në një makinë nikoqire për ta zhvilluar softuerin, i cili bartet te një makinë cak për ekzekutim.
- Zhvillimi me kod të hapur e përfshin qitjen në dispozicion publikisht të kodit burimor të një sistemi. Kjo domethënë se shumë njerëz mund të propozojnë ndryshime dhe përmirësime për softuerin.

8. Testimi i softuerit

- Testi mund vetëm ta tregojë prezencën e gabimeve në një program. Ai nuk mund ta demonstrojë se nuk ka më të meta tjera.
- Testimi i zhvillimit është përgjegjësi e ekipit të zhvillimit të softuerit. Një ekip i veçantë duhet të jetë përgjegjës për testimin e një sistemi para se të instalohet te klientët. Në procesin e testimit nga përdoruesit, klientët apo përdoruesit e sistemit ofrojnë shënime testuese dhe e kontrollojnë se a janë testet të suksesshme.
- Testimi i zhvillimit e përfshin testimin e njësisë (unit testing), në të cilin i testoni objektet dhe metodat individuale; testimin e komponentave (component testing), në të cilin testoni grupe të ndërlidhura të objekteve; dhe testimin e sistemit (system testing), në të cilin testoni sisteme të pjesshme apo të plota.
- Kur e testoni softuerin, duhet të provoni ta 'thyeri' softuerin duke e përdorur përvojën dhe udhëzimet për t'i zgjedhur llojet e rasteve të testimit që kanë qenë efektive në zbulimin e defekteve në sistemet tjera.
- Kurdo që është e mundur, duhet të shkruani teste të automatizuara. Testet janë të integruara në një program që mund të ekzekutohet sa herë që është bërë një ndryshim në sistem.
- Zhvillimi testet-së-pari (test-first development) është qasje ndaj zhvillimit ku testet shkruhen para kodit që do të testohet. Bëhen ndryshime të vogla në kod dhe kodi rifaktorohet deri sa të gjitha testet ekzekutohen me sukses.
- Testimi me skenarë (scenario testing) është i dobishëm sepse e replikon përdorimin praktik të sistemit. Ai e përfshin shpikjen e një skenari tipik të përdorimit dhe përdorjen e kësaj për t'i nxjerrë rastet e testimit.
- Testimi i pranimi është një proces i testimit nga përdoruesit ku qëllimi është të vendoset se a është softueri mjaft i mirë që të instalohet dhe të përdoret në ambientin e tij operacional.

9. Evolucionit i softuerit

- Zhvillimi i softuerit dhe evolucioni mund të mendohen si proces i integruar, iterativ që mund të paraqitet duke e përdorur një model spiral.
- Për sistemet specifike (të porositura enkas), kostot e mirëmbajtjes zakonisht i kalojnë kostot e zhvillimit të softuerit.
- Procesi i evolucionit të softuerit udhëhiqet nga kërkesat për ndryshime dhe e përfshin analizën e ndikimit të ndryshimit, planifikimin e instalimit të versionit të ri, dhe implementimin e ndryshimit.
- Ligjet e Lehmanit, si nocioni se ndryshimi është i vazhdueshëm, e përshkruajnë një numër të mençurive të nxjerra nga studimet afat-gjata të evolucionit të sistemit.
- Janë tri lloje të mirëmbajtjes së softuerit: rregullimi i defekteve, ndryshimi i softuerit për të punuar në ambient të ri, dhe implementimi i kërkesave të reja apo të ndryshuara.
- Riinxhinierimi i softuerit merret me ristrukturimin dhe ridokumentimin e softuerit për ta bërë më të lehtë të kuptohet dhe të ndryshohet.
- Rifaktorimi, bërja e ndryshimeve të vogla në program që e ruajnë funksionalitetin, mund të mendohet si mirëmbajtje parandaluese.
- Duhet të vlerësohen vlera biznesore e një sistemi të vjetër dhe cilësia e softuerit aplikativ dhe ambientit të tij për ta përcaktuar se a duhet të zëvendësohet, të transformohet, apo të mirëmbahet sistemi.

PJESA II: BESUESHMËRIA DHE SIGURIA

10. Sistemet socioteknike

- Sistemet socioteknike e përfshijnë harduerin kompjuterik, softuerin, dhe njerëzit dhe ndodhen brenda një organizate. Ato projektohen për t'i përkrahur qëllimet dhe objektivat e organizatës apo të biznesit.
- Faktorët njerëzorë dhe organizacionalë siç është struktura e organizatës dhe politika kanë efekt të konsiderueshëm në operimin e sistemeve socioteknike.
- Vetitë emergjente të një sistemi janë karakteristika të sistemit si tërësi e jo të pjesëve përbërëse të tij. Ato përfshijnë veti si performansa, besueshmëria (mbështetshmëria), përdorshmëria, siguria fizike, dhe siguria e informatave. Suksesi apo dështimi i një sistemi shpesh varet nga këto veti emergjente.
- Proceset themelore të inxhinierimit të sistemeve janë prokurimi/blerja e sistemit, zhvillimi i sistemit, dhe operimi i sistemit.
- Prokurimi i sistemit i mbulon të gjitha aktivitetet e përfshira në vendosjen se çfarë sistemi të blihet dhe kush duhet ta furnizojë atë sistem. Si pjesë e procesit të prokurimit zhvillohen kërkesat e nivelit të lartë.
- Zhvillimi i sistemit e përfshin specifikimin e kërkesave, projektimin, konstruktimin, integrimin, dhe testimin. Integrimi i sistemeve, ku nënsistemet nga më shumë se një furnizues duhet të bëhen të punojnë bashkë, është veçanërisht kritik.
- Kur një sistem vihet në përdorim, proceset operationale dhe vetë sistemi duhet të ndryshojë që t'i reflektojë kërkesat ndryshuese të biznesit.
- Gabimet njerëzore janë të pashmangshme dhe sistemet duhet të përfshijnë barriera për t'i detektuar këto gabime para se ato të çojnë në dështimin e sistemit. Modeli i Djathit Zvicëran i Reason-it e shpjegon se si gabimi njerëzor plus defektet e fshehura në barriera mund të çojnë në dështimin e sistemit.

11. Besueshmëria (mbështetshmëria) dhe siguria e informatave

- Dështimi i sistemeve kritike kompjuterike mund të çojë në humbje të mëdha ekonomike, humbje serioze të informatave, dëmtim fizik, apo kërcënime ndaj jetës njerëzore.
- Besueshmëria e një sistemi kompjuterik është veti e sistemit që e reflekton shkallën e besimit të përdoruesit në sistemin. Përmasat më të rëndësishme të besueshmërisë janë disponueshmëria, mbështetshmëria (reliability), siguria fizike, dhe siguria e informatave.
- Disponueshmëria e një sistemi është gjasa që sistemi do të jetë në gjendje t'ua ofrojë shërbimet përdoruesve të vet kur kërkohet ta bëjë këtë. Mbështetshmëria është gjasa që shërbimet e sistemit do të ofrohen siç është specifikuar.
- Mbështetshmëria e perceptuar ndërlidhet me gjasën që një gabim të ndodhë në përdorim operacional. Një program mund të përmbajë defekte të njohura mirëpo prapë mund të përjetohet si i mbështetshëm nga përdoruesit e tij. Ata mund të mos i përdorin kurrë veçoritë e sistemit që ndikohen nga defektet.
- Siguria fizike e një sistemi është atribut i sistemit që e reflekton aftësinë e sistemit të operojë, normalisht apo abnormalisht, pa lëndime te njerëzit apo dëme në ambient.
- Siguria e informatave e reflekton aftësinë e një sistemi ta mbrojë veten ndaj sulmeve të jashtme. Defektet e sigurisë mund të çojnë në humbje të disponueshmërisë, ta dëmtojnë sistemin apo shënimet e tij, apo rrjedhjen e informatave te njerëzit e paautorizuar.
- Pa nivel të pranueshëm të sigurisë së informatave, disponueshmëria, mbështetshmëria, dhe siguria fizike e sistemit mund të komprometohen nëse sulmet e jashtme e dëmtojnë sistemin. Nëse një sistem është i pambështetshëm, është vështirë të garantohet siguria fizike apo e informatave, meqë ato mund të komprometohen nga defektet e sistemit.

12. Besueshmëria dhe specifikimi i sigurisë

- Analiza e rreziqeve është aktivitet i rëndësishëm në specifikimin e kërkesave të sigurisë dhe besueshmërisë. Ajo e përfshin identifikimin e rreziqeve që mund të rezultojnë në aksidente apo incidente. Kërkesat e sistemit pastaj gjenerohen ashtu që të sigurojnë që këto rreziqe nuk ndodhin dhe, nëse ndodhin, se nuk çojnë në incident apo aksident.
- Qasja e udhëhequr nga rreziqet (hazard-driven) mund të përdoret për t'i kuptuar kërkesat e sigurisë fizike për një sistem. I identifikoni rreziqet potenciale dhe i shpërbëni ato (duke i përdorur metodat si analiza e pemës së defekteve) për t'i zbuluar shkaqet e tyre rrënjë. Pastaj i specifikoni kërkesat për t'iu shmangur apo për t'u rikëndellur nga këto probleme.
- Kërkesat e besueshmërisë mund të definohen në mënyrë sasiore në specifikimin e kërkesave të softuerit. Metrikat e mbështetshmërisë e përfshijnë gjasën e dështimit në momentin e kërkesës së shërbimit (probability of failure on demand, POFOD), shkallën e paraqitjes së dështimit (rate of occurrence of failure, ROCOF), dhe disponueshmërinë (availability, AVAIL).
- Është me rëndësi të mos mbispecifikohet mbështetshmëria e kërkuar e sistemit meqë kjo çon në kosto shtesë të panevojshme në zhvillim dhe në procese të validimit.
- Kërkesat e sigurisë së informatave janë më të vështira të identifikohen se sa kërkesat e sigurisë fizike sepse një sulmues i sistemit mund ta përdorë njohurinë e pikave të dobëta të sistemit për ta planifikuar një sulm të sistemit, dhe mund të mësojë për pikat e dobëta nga sulmet e pasuksesshme.
- Për t'i specifikuar kërkesat e sigurisë, duhet t'i identifikoni asetet që duhet të mbrohen dhe ta definoni se si duhet të përdoren teknikat dhe teknologjitë e sigurisë për t'i mbrojtur këto aset.
- Metodatat formale të zhvillimit të softuerit mbështeten në një specifikim të sistemit që shprehet si model matematikor. Zhvillimi i një specifikacioni formal e ka përfitimin kryesor të stimulimit të një kontrollimi dhe analize detale të kërkesave të sistemit.

13. Inxhinierimi i besueshmërisë

- Besueshmëria në një program mund të arrihet duke iu shmangur paraqitjes së defekteve, duke i detektuar dhe hequr defektet para instalimit të sistemit, dhe duke përfshirë pajisje të tolerancës ndaj defekteve që ia lejojnë sistemit të mbesë operacional pasi që një defekt ka shkaktuar dështim të sistemit.
- Përdorja e redundansës (përsëritjes) dhe diversitetit në harduer, procese softuerike, dhe sisteme softuerike është thelbësore në zhvillimin e sistemeve të besueshme.
- Përdorja e një procesi të definuar mirë e të përsëritshëm është thelbësore nëse duhet të minimizohen dështimet në një sistem. Procesi duhet t'i përfshijë aktivitetet e verifikimit dhe të validimit në të gjitha fazat, nga definimi i kërkesave deri te implementimi i sistemit.
- Arkitektura të besueshme të sistemit janë arkitekturat e sistemit që janë projektuar për tolerancë ndaj defekteve. Ka një numër të stileve arkitekturale që e përkrahin tolerancën ndaj dështimeve përfshirë sistemet mbrojtëse, arkitekturat vetë-monitoruese, dhe programimi N-versional.
- Diversiteti i softuerit arrihet vështirë sepse është praktikisht e pamundur të garantohet se secili version i softuerit është vërtet i pavarur. Programimi i besueshëm mbështetet në përfshirjen e redundansës në një program për ta kontrolluar vlefshmërinë e hyrjeve dhe vlerave të ndryshoreve të programit.
- Disa konstrukte dhe teknika të programimit, si urdhrat go-to, pointerët, rekursionin, trashëgimia, dhe numrat me pikë lëvizëse, janë në thelb të prira për gabime. Duhet të provoni t'iu shmangeni këtyre konstrukteve kur zhvilloni sisteme të besueshme.

14. Inxhinierimi i sigurisë së informatave

- Inxhinierimi i sigurisë përqendrohet në atë se si të zhvillohen dhe të mirëmbahen sistemet softuerike që mund t'iu rezistojnë sulmeve keqbërëse të synuara për ta dëmtuar një sistem të bazuar në kompjuter apo shënimet e tij.
- Kërcënimet e sigurisë mund të jenë kërcënime ndaj konfidencialitetit, integritetit, apo disponueshmërisë së një sistemi apo shënimeve të tij.
- Menaxhimi i rreziqeve të sigurisë e përfshin vlerësimin e humbjeve që mund të ndodhin nga sulmet në një sistem, dhe nxjerrjen e kërkesave të sigurisë që e kanë synim eliminimin apo zvogëlimin e këtyre humbjeve.
- Projektimi për siguri e përfshin projektimin e një arkitekture të sigurt të sistemit, ndjekjen e praktikës së mirë për projektim të sistemeve të sigurt, dhe përfshirjen e funksionalitetit për ta minimizuar gjasën e përhapjes së pikave të dobëta kur të instalohet sistemi.
- Çështjet kryesore gjatë projektimit të një arkitekture të sistemeve të sigurt e përfshijnë organizimin e strukturës së sistemit për t'i mbrojtur asetet kryesore dhe shpërndarjen e aseteve të sistemit për t'i minimizuar humbjet nga një sulm i suksesshëm.
- Udhëzimet e projektimit të sigurisë i sensibilizojnë projektuesit e sistemit për çështjet e sigurisë që mund të mos i kenë konsideruar. Ato ofrojnë bazë për krijimin e listave për kontrollim të sigurisë.
- Për ta përkrahur instalimin e sigurt duhet ta ofroni një mënyrë të shfaqjes dhe analizimit të konfiguracioneve të sistemit, t'i lokalizoni rregullimet e konfiguracionit ashtu që konfigurimet e rëndësishme nuk harrohen, t'i minimizoni privilegjet e paracaktuara për përdoruesit e sistemit, dhe të ofroni mënyra për t'i riparuar pikat e dobëta të sigurisë.
- Mbijetueshmëria e sistemit e reflekton aftësinë e sistemit të vazhdojë t'i ofrojë shërbimet thelbësore biznesore apo kritike për misionin për t'i legjitimuar përdoruesit ndërsa është nën sulm, apo pasi të jetë dëmtuar një pjesë e sistemit.

15. Besueshmëria dhe garantimi i sigurisë

- Analiza statike është qasje ndaj V & V (verifikimi dhe validimi) që e shqyrton kodin burimor (apo ndonjë paraqitje tjetër) të një sistemi, duke kërkuar gabime apo anomali. Ajo lejon të kontrollohen të gjitha pjesët e një programi, jo vetëm ato pjesë që janë provuar nga testet e sistemit.
- Kontrollimi i modelit është qasje formale ndaj analizës statike që i kontrollon në mënyrë shteruese të gjitha gjendjet në një sistem duke kërkuar gabime potenciale.
- Testimi statistikor përdoret për ta vlerësuar besueshmërinë softuerike. Ai mbështetet në testimin e sistemit me një grup testues të shënimeve që e reflekton profilin operacional të softuerit. Shënimet testuese mund të gjenerohen automatikisht.
- Validimi i sigurisë është i vështirë sepse kërkesat e sigurisë deklarojnë se çka duhet të mos ndodhë në një sistem, në vend se çka duhet të ndodhë. Pos kësaj, sulmuesit e sistemit janë inteligjentë dhe mund të kenë më shumë kohë të kërkojnë pika të dobëta se sa që është në dispozicion për testimin e sigurisë.
- Validimi i sigurisë mund të bëhet duke përdorur analizë të bazuar në përvojë, analizë të bazuar në vegla, apo 'ekipe tigr' që i simulojnë sulmet në një sistem.
- Është me rëndësi të keni proces të definuar mirë, të certifikuar për zhvillim të sistemeve që janë kritike për sigurinë fizike. Procesi duhet ta përfshijë identifikimin dhe monitorimin e rreziqeve potenciale.
- Rastet e sigurisë fizike dhe të besueshmërisë e grumbullojnë tërë evidencën që demonstroi se një sistem është fizikisht i sigurt dhe i besueshëm. Rastet e sigurisë fizike kërkohen kur një rregullator i jashtëm duhet ta certifikojë sistemin para se të përdoret.
- Rastet e sigurisë fizike zakonisht bazohen në argumente të strukturuar. Argumentet e strukturuar të sigurisë fizike tregojnë se një gjendje apo kusht i identifikuar nuk mund të ndodhë kurrë duke i konsideruar të gjitha rrugët e programit që çojnë në kushtin apo gjendjen e pasigurt, dhe duke treguar se kushti apo gjendja nuk mund të ndodhë.

PJESA 3: INXHINIERIMI I AVANSUAR I SOFTUERIT

16. Ripërdorja e softuerit

- Shumica e sistemeve të reja softuerike të biznesit tash zhvillohen duke e ripërdorur njohurinë dhe kodin nga sistemet e implementuara më parë.
- Ka mënyra të ndryshme për ta ripërdorur softuerin. Këto shkojnë nga ripërdorja e klasave dhe e metodave në librari deri te ripërdorja e sistemeve të plota aplikative.
- Përparësitë e ripërdorjes së softuerit janë kostot më të ulëta, zhvillimi më i shpejtë i softuerit, dhe rreziqet më të vogla. Besueshmëria e sistemit rritet. Specialistët mund të përdoren në mënyrë më efikase duke e përqendruar ekspertizën e tyre në projektimin e komponentave të ripërdorshme.
- Kornizat e aplikacioneve (application frameworks) janë koleksione të objekteve konkrete dhe abstrakte që janë projektuar për ripërdorje përmes specializimit dhe shtimit të objekteve të reja. Ato zakonisht përfshijnë praktika të mira të projektimit përmes mostrave të projektimit (design patterns).
- Vijat e produktit softuerik janë aplikacione të ndërlidhura që janë zhvilluar nga një apo më shumë aplikacione bazë. Një sistem i përgjithshëm adaptohet dhe specializohet për t'i plotësuar kërkesat specifike për funksionalitet, platformë të shtuar, apo konfigurim operacional.
- Ripërdorja e produkteve COTS (Commercial Off the Shelf - komerciale, të gatshme për shitje) merret me ripërdorjen e sistemeve të shkallës së madhe, të gatshme për shitje. Këto ofrojnë shumë funksionalitet dhe ripërdorja e tyre mund t'i zvogëlojë rrënjësisht kostot dhe kohën e zhvillimit. Sistemet mund të zhvillohen duke e konfiguruar një produkt të vetëm, të përgjithshëm COTS apo duke i integruar dy apo më shumë produkte COTS.
- Sistemet për Planifikimin e Resurseve të Ndërmarrjes (Enterprise Resource Planning, ERP) janë shembuj të ripërdorjes së shkallës së madhe të COTS. E krijoni një instancë të një sistemi ERP duke e konfiguruar një sistem të përgjithshëm me informata për proceset dhe rregullat e biznesit të klientit.
- Problemet potenciale me ripërdorjen e bazuar në COTS përfshijnë mungesën e kontrollit ndaj funksionalitetit dhe performansës, mungesën e kontrollit ndaj evolucionit të sistemit, nevojën për përkrahje nga furnizuesit e jashtëm, dhe vështirësitë në garantimin se sistemet mund të ndëroperojnë.

17. Inxhinierimi softuerik i bazuar në komponenta

- Inxhinierimi softuerik i bazuar në komponenta është qasje e bazuar në ripërdorje për definimin, implementimin dhe kompozimin në sisteme të komponentave të pavarura të çiftuara lirshëm (dobët).
- Komponenta është njësi softuerike funksionaliteti dhe varësitë e të cilës janë të definuara plotësisht nga një grup i interfejsave publikë. Komponentat mund të kompozohen me komponenta tjera pa njohuri për implementimin e tyre dhe mund të instalohen si njësi e ekzekutueshme.
- Komponentat mund të implementohen si njësi të programit që përfshihen në një sistem apo si shërbime të jashtme që referohen nga brendia e një sistemi.
- Modeli i komponentave e definon një grup të standardeve për komponenta, përfshirë standardet e interfejsave, standardet e përdorjes, dhe standardet e instalimit. Implementimi i modelit të komponentave e ofron një grup të shërbimeve të përbashkëta që mund të përdoret nga të gjitha komponentat.
- Gjatë procesit CBSE (component-based software engineering - inxhinierimi softuerik i bazuar në komponenta), duhet t'i gërshetoni proceset e inxhinierimit të kërkesave dhe të projektimit të sistemit. Duhet t'i balansoni kërkesat e dëshirueshme me shërbimet që janë në dispozicion nga komponentat ekzistuese të ripërdorshme.
- Kompozimi i komponentave është procesi i 'lidhjes me tela' të komponentave për ta krijuar një sistem. Llojet e kompozimit e përfshijnë kompozimin sekuencial, kompozimin hierarkik, dhe kompozimin shtues.
- Gjatë kompozimit të komponentave të ripërdorshme që nuk janë shkruar për aplikacionin tuaj, mund të ju duhet të shkruani adaptues apo 'kod ngjitës' për t'i pajtuar interfejsat e komponentave të ndryshme.
- Kur i zgjedhni kompozimet, duhet ta konsideroni funksionalitetin e kërkuar të sistemit, kërkesat jofunksionale dhe lehtësinë me të cilën mund të zëvendësohet një komponentë kur ndryshohet sistemi.

18. Inxhinierimi i softuerit të shpërndarë

- Përfitimet e sistemeve të shpërndara janë se ato mund të shkallëzohen për t'ia dalë me kërkesat që rriten, mund të vazhdojnë të ofrojnë shërbime për përdoruesit (edhe nëse disa pjesë të sistemit dështojnë), dhe mundësojnë të bashkëpërdoren resurset.
- Çështjet që duhet të konsiderohen në projektimin e sistemeve të shpërndara e përfshijnë transparencën, të qenit i hapur, shkallëzueshmërinë, sigurinë, cilësinë e shërbimit, dhe menaxhimin e dështimit.
- Sistemet klient-server janë sisteme të shpërndara në të cilat sistemi është i strukturuar në shtresa, me shtresën e prezantimit të implementuar në kompjuterin klient. Serverët ofrojnë shërbime të menaxhimit të shënimeve, të aplikacionit, dhe të bazës së shënimeve.
- Sistemet klient-server mund të kenë disa shtresa lidhëse, ku shtresa të ndryshme të sistemit janë të shpërndara në kompjuterë të ndryshëm.
- Mostrat arkitekturale (architectural patterns) për sistemet e shpërndara përfshijnë arkitekturat zotërues-skllav (master-slave), arkitekturat dy-shtresore dhe shumë-shtresore klient-server, arkitekturat e komponentave të shpërndara, dhe arkitekturat shok-me-shok (peer-to-peer).
- Sistemet me komponenta të shpërndara kërkojnë softuer ndërmjetësues (middleware) për t'i trajtuar komunikimet e komponentave dhe për t'ua lejuar komponentave të shtohen dhe të hiqen nga sistemi.
- Arkitekturat shok-me-shok janë arkitektura të decentralizuara në të cilat nuk ka klientë dhe serverë të dalluar. Llogaritjet mund të shpërndahen nëpër shumë sisteme në organizata të ndryshme.
- Softueri si shërbim (software as a service, SaS) është mënyrë e shpërndarjes së aplikacioneve si sisteme të lehta klient-server, ku klienti është shfletues i web-it.

19. Arkitektura e orientuar kah shërbimet (SOA)

- Arkitektura e orientuar kah shërbimet (serviset) është qasje ndaj inxhinierimit të softuerit ku shërbimet e ripërdorshme, të standardizuara janë blloqe themelore të ndërtimit për sisteme aplikative.
- Interfejsat e shërbimeve mund të definohen në një gjuhë të bazuar në XML të quajtur WSDL. Një specifikacion WSDL e përfshin definicionin e llojeve dhe operacioneve të interfejsit, protokollin lidhës të përdorur nga shërbimi dhe lokacionin e shërbimit.
- Shërbimet mund të klasifikohen si shërbime vegla që ofrojnë funksionalitet me qëllim të përgjithshëm, shërbime biznesore që implementojnë pjesë të një procesi të biznesit, apo shërbime të koordinimit që e koordinojnë ekzekutimin e shërbimeve tjera.
- Procesi i inxhinierimit të shërbimeve e përfshin identifikimin e shërbimeve kandidat për implementim, definimin e interfejsit të shërbimit dhe implementimin, dhe testimin dhe instalimin e shërbimit.
- Interfejsat e shërbimit mund të definohen për sisteme të vjetra softuerike që vazhdojnë të jenë të dobishme për një organizatë. Funksionaliteti i sistemit të vjetër mund të ripërdoret pastaj në aplikacione tjera.
- Zhvillimi i softuerit duke përdorur shërbime bazohet rreth idesë se programet krijohen duke kompozuar dhe konfiguruar shërbime për të krijuar shërbime të reja kompozite.
- Modelet e proceseve biznesore i definojnë aktivitetet dhe shkëmbimin e informatave që ndodh në një proces biznesor. Aktivitetet në procesin biznesor mund të implementohen nga shërbimet ashtu që modeli i procesit biznesor e paraqet një kompozim të shërbimeve.

20. Softueri i integruar në harduer (embedded)

- Sistemi softuerik i integruar është pjesë e një sistemi harduer/softuer që reagon ndaj ngjarjeve në ambientin e tij. Softueri është 'i integruar' në harduer. Sistemet e integruara normalisht janë sisteme në kohë reale.
- Sistemi në kohë reale është sistem softuerik që duhet të reagojë ndaj ngjarjeve në kohë reale. Korrektësia e sistemit nuk varet vetëm nga rezultatet që i prodhon, por edhe nga koha kur janë prodhuar këto rezultate.
- Sistemet në kohë reale zakonisht implementohen si grup i proceseve komunikuese që reagojnë ndaj stimuleve për të prodhuar përgjigje.
- Modelet e gjendjes janë paraqitje e rëndësishme e projektimit për sistemet e integruara në kohë reale. Ato përdoren për të treguar se si reagon sistemi ndaj ambientit të tij ndërsa ngjarjet nxisin ndryshime të gjendjes në sistem.
- Ka disa mostra standarde që mund të vëzhgohen në llojet e ndryshme të sistemeve të integruara. Këto e përfshijnë një mostër për monitorimin e ambientit të sistemit për ngjarje të pafavorshme, një mostër për kontroll të aktivizuesve dhe një mostër për procesim të shënimeve.
- Projektuesit e sistemeve në kohë reale duhet të bëjnë analizë të koordinimit kohor, që udhëhiqet nga afatet për procesimin dhe përgjigjen ndaj stimuleve. Ata duhet të vendosin se sa shpesh duhet të ekzekutohet secili proces në sistem dhe për kohën e ekzekutimit të rastit më të keq për proceset.
- Një sistem operativ në kohë reale është përgjegjës për menaxhimin e proceseve dhe resurseve. Ai gjithmonë e përfshin një planifikues të orarit, që është komponenta përgjegjëse për vendosjen se cili proces duhet të planifikohet për ekzekutim.

21. Inxhinierimi i softuerit i orientuar kah aspektet

- Përfitimi kryesor i qasjes së orientuar kah aspektet ndaj zhvillimit të softuerit është se ajo e përkrah ndarjen e interesimeve (separation of concerns). Duke i paraqitur interesimet ndërprerëse si aspekte, interesimet individuale mund të kuptohen, ripërdoren, dhe ndryshohen pa i ndryshuar pjesët tjera të programit.
- Lëmshi ndodh kur një modul në një sistem përfshin kod që përfshin kërkesa të ndryshme të sistemit. Fenomeni i ndërlidhur i shpërndarjes ndodh kur implementimi i një interesimi të vetëm është i shpërndarë nëpër disa komponenta në një program.
- Aspektet e përfshijnë një pikë prerëse - një deklaratë që e definon se ku do të thuret aspekti në program, dhe këshillën - kodin për ta implementuar interesimin ndërprerës. Pikat e bashkimit janë ngjarjet që mund të referohen në një pikë prerëse.
- Për ta garantuar ndarjen e interesimeve, sistemet mund të projektohen si sistem bërthamë që i implementon interesimet parësore të palëve të interesit, dhe një grup të zgjerimeve që i implementojnë interesimet dytësore.
- Për t'i identifikuar interesimet, mund të përdorni qasje të orientuar kah pikëpamjet ndaj inxhinierimit të kërkesave për t'i nxjerrë kërkesat e palëve të interesit (stakeholders) dhe për ta identifikuar cilësinë ndërprerëse të shërbimit dhe çështjet e politikave.
- Tranzicioni nga kërkesat në projektim mund të bëhet duke i identifikuar rastet e përdorimit, ku secili rast i përdorimit e përfaqëson një interesim të palës së interesit. Projektimi mund të modelohet duke përdorur version të zgjeruar të UML me stereotipe aspekt.
- Problemet e inspektimit dhe nxjerrjes së testeve për programet e orientuara kah aspektet janë barrierë e konsiderueshme për adoptimin e zhvillimit të softuerit të orientuar kah aspektet në projekte të mëdha softuerike.

PJESA 4: MENAXHIMI I SOFTUERIT

22. Menaxhimi i projektit

- Menaxhimi i mirë i projektit softuerik është thelbësor nëse projektet e inxhinierimit të softuerit do të zhvillohen me orar dhe brenda buxhetit.
- Menaxhimi i softuerit dallohet nga menaxhimi tjetër i inxhinierimit. Softueri është i paprekshëm. Projektet mund të jenë të reja apo inovative kështu që nuk ka trup të përvojës për ta udhëzuar menaxhimin e tyre. Proceset softuerike nuk janë aq të pjekura sa proceset tradicionale të inxhinierimit.
- Menaxhimi i rrezikut tash njihet si njëri nga detyrat më të rëndësishme të menaxhimit të projektit.
- Menaxhimi i rrezikut e përfshin identifikimin dhe vlerësimin e rreziqeve të mëdha të projektit për ta caktuar gjasën se ato do të ndodhin dhe pasojat për projektin nëse ai rrezik ndodh. Duhet të bëni plane për t'iu shmangur, për t'i menaxhuar, apo për t'u marrë me rreziqet e mundshme nëse apo kur të ndodhin.
- Njerëzit motivohen me ndërveprim me njerëz tjerë, vlerësim nga menaxhmenti dhe kolegët, dhe duke iu dhënë shanse për zhvillim personal.
- Grupet e zhvillimit të softuerit duhet të jenë të vogla dhe kohezive. Faktorët kryesorë që ndikojnë në efektivitetin e një grupi janë njerëzit në atë grup, mënyra si është organizuar grupi, dhe komunikimi ndërmjet anëtarëve të grupit.
- Komunikimet brenda një grupi ndikohen nga faktorët si statusi i anëtarëve të grupit, madhësia e grupit, përbërja gjinore e grupit, personalitetet, dhe kanalet e disponueshme të komunikimit.

23. Planifikimi i projektit

- Çmimi i kërkuar për një sistem nuk varet vetëm nga kostot e tij të vlerësuara të zhvillimit dhe fitimi i kërkuar nga kompania e zhvillimit. Faktorët organizacionalë mund ta paracaktojnë që çmimi të rritet për shkak të rrezikut të rritur apo të zvogëlohet për të fituar përparësi konkurruese.
- Softuerit shpesh i caktohet çmimi për ta fituar një kontratë dhe funksionaliteti i sistemit pastaj rregullohet që të përputhet me çmimin e vlerësuar.
- Zhvillimi i udhëhequr nga plani (plan-driven) organizohet përreth një plani të plotë të projektit që i definon aktivitetet e projektit, angazhimet e planifikuara, orarin e aktiviteteve, dhe kush është përgjegjës për secilin aktivitet.
- Orarizimi i projektit e përfshin krijimin e paraqitjeve të ndryshme grafike të një pjese të planit të projektit. Diagramet me shirita, që e tregojnë kohëzgjatjen dhe stafin, janë paraqitjet më të përdorura të orarit.
- Guri kilometrik i projektit është një rezultat i parashikueshëm i një aktiviteti apo një grupi të aktiviteteve. Në secilin gur kilometrik, menaxhmentit duhet t'i prezantohet një raport apo progres formal. Rezultati i pritshëm (deliverable) është një produkt i punës që i dorëzohet klientit të projektit.
- Loja e planifikimit e XP (programimit ekstrem) e përfshin tërë ekipin në planifikimin e projektit. Plani zhvillohet rritshëm (inkrementalisht) dhe, nëse dalin probleme, rregullohet ashtu që funksionaliteti i softuerit zvogëlohet në vend se të vonohet dorëzimi i një iteracioni.
- Teknikat e vlerësimit për softuer mund të jenë të bazuara në përvojë, ku menaxherët e gjejnë angazhimin e kërkuar, apo algoritmike, ku angazhimi i kërkuar llogaritet nga parametra tjerë të vlerësuar të projektit.
- Modeli i kostove COCOMO II është model i pjekur algoritmik i kostos që i merr parasysh atributet e projektit, produktit, harduerit, dhe të personelit kur e formulon një vlerësim të kostos.

24. Menaxhimi i cilësisë

- Menaxhimi i cilësisë së softuerit merret me garantimin që softueri ka numër të ulët të defekteve dhe që i arrin standardet e kërkuara të mirëmbajtshmërisë, besueshmërisë, bartshmërisë, dhe kështu me radhë. Ai e përfshin definimin e standardeve për proceset dhe produktet dhe vendosjen e proceseve për të kontrolluar se këto standarde janë ndjekur.
- Standardet softuerike janë të rëndësishme për sigurimin e cilësisë meqë përfaqësojnë identifikim të 'praktikës më të mirë'. Kur zhvillohet softuer, standardet ofrojnë themel solid për ndërtimin e softuerit të cilësisë së mirë.
- Duhet ta dokumentoni një grup të procedurave të sigurimit të cilësisë në një doracak të cilësisë të organizatës. Kjo mund të bazohet në modelin e përgjithshëm për doracak të cilësisë të sugjeruar në standardin ISO 9001.
- Kontrollimet e rezultateve të pritshme të procesit softuerik e përfshijnë një ekip të njerëzve që e kontrollojnë se po ndiqen standardet e cilësisë. Kontrollimet janë teknika më e përdorur për vlerësimin e cilësisë.
- Në inspektimin e programit apo kontrollimin nga kolegët, një ekip i vogël e kontrollon sistematikisht kodin. Ata e lexojnë kodin detalisht dhe kërkojnë gabime apo harresa të mundshme. Problemet e detektuara pastaj diskutohen në një takim të kontrollimit të kodit.
- Matja e softuerit mund të përdoret për të mbledhur shënime sasiore për softuerin dhe procesin softuerik. Mund të jeni në gjendje t'i përdorni vlerat e metrikave të softuerit që janë grumbulluar për të nxjerrë përfundime për cilësinë e produktit dhe të procesit.
- Metrikat e cilësisë së produktit janë veçanërisht të dobishme për nxjerrjen në pah të komponentave anomalike që mund të kenë probleme të cilësisë. Këto komponenta duhet pastaj të analizohen më detalisht.

25. Menaxhimi i konfigurimit

- Menaxhimi i konfigurimit është menaxhimi i një sistemi evolues softuerik. Kur mirëmbahet një sistem, vendoset një ekip i MK për të siguruar se ndryshimet përfshihen në sistem në mënyrë të kontrolluar dhe se regjistrimet mirëmbahen me detalet e ndryshimeve që janë implementuar.
- Proceset kryesore të menaxhimit të konfigurimit merren me menaxhimin e ndryshimit, menaxhimin e versioneve, ndërtimin e sistemit, dhe menaxhimin e shpërndarjes (instalimit). Ka në dispozicion vegla softuerike për t'i përkrahur të gjitha këto procese.
- Menaxhimi i ndryshimit e përfshin vlerësimin e propozimeve për ndryshime nga klientët e sistemit dhe palët tjera të interesit (stakeholders) dhe vendosjen se a është efektive për kosto implementimi i tyre në një version të ri të sistemit.
- Menaxhimi i versioneve e përfshin mbajtjen e shënimeve për versionet e ndryshme të komponentave të softuerit që krijohen dhe ndryshimet që iu janë bërë atyre.
- Ndërtimi i sistemit është procesi i montimit të komponentave të sistemit në një program të ekzekutueshëm për t'u ekzekutuar në një sistem kompjuterik të synuar.
- Softueri duhet të rindërtohet shpesh dhe të testohet menjëherë pasi të jetë ndërtuar një version i ri. Kjo e bën më të lehtë të detektohen defektet dhe problemet që janë paraqitur që nga ndërtimi i fundit.
- Versionet instaluese (releases) të sistemit përfshijnë kod të ekzekutueshëm, fajlla të shënimeve, fajlla të konfigurimit, dhe dokumentim. Menaxhimi i instalimit e përfshin marrjen e vendimeve për datat e instalimit të sistemit, përgatitjen e të gjitha informatave për shpërndarje, dhe dokumentimin e secilit version të sistemit.

26. Përmirësimi i procesit

- Qëllimet e përmirësimit të procesit janë cilësia më e lartë e produktit, kostot e zvogëluara të procesit, dhe dorëzimi më i shpejtë i softuerit.
- Qasjet kryesore ndaj përmirësimit të procesit janë qasjet agjile, të përshtatura për zvogëlimin e ngarkesës së procesit, dhe qasjet e bazuara në pjekuri të bazuara në menaxhim më të mirë të procesit dhe përdorim të praktikës së mirë të inxhinierimit të softuerit.
- Cikli i përmirësimit të procesit e përfshin matjen e procesit, analizën dhe modelimin e procesit, dhe ndryshimin e procesit.
- Modelet e procesit, që i tregojnë aktivitetet në një proces dhe relacionet e tyre me produktet softuerike, përdoren për përshkrim të procesit. Në praktikë, megjithatë, inxhinierët e përfshirë në zhvillimin e softuerit gjithmonë i adaptojnë modelet ndaj rrethanave të tyre lokale.
- Matja duhet të përdoret për t'iu përgjigjur pyetjeve specifike për procesin softuerik të përdorur. Këto pyetje duhet të bazohen në qëllimet organizacionale të përmirësimit.
- Tri lloje të metrikave të procesit të përdorura në procesin e matjes janë metrikat e kohës, metrikat e shfrytëzimit të resurseve, dhe metrikat e ngjarjeve.
- Modeli CMMI i pjekurisë së procesit është model i integruar i përmirësimit të procesit që e përkrah edhe përmirësimin me faza edhe atë të vazhdueshëm të procesit.
- Përmirësimi i procesit në modelin CMMI bazohet në arritjen e një grupi të qëllimeve të lidhura me praktikën e mirë të inxhinierimit të softuerit dhe përshkrimin, standardizimin, dhe kontrollimin e praktikave të përdorura për t'i arritur këto qëllime. Modeli CMMI i përfshin praktikatat e rekomanduara që mund të përdoren, mirëpo këto nuk janë të obligueshme.

Online

Ky publikim mund të gjendet online te:

Softa Shqiptare: <http://sites.google.com/site/softashqiptare/>

Web-faqja e Ridvan Bunjakut: <http://sites.google.com/site/kalabuli/>