



Inxhinierimi softverik

Pjesa 3

Prof. Asoc. Dr. Ermir Rogova

Inxhinierimi i softverit

- Me rritjen e madhësisë dhe kompleksitetit të projekteve softverike filloi të rritet dhe numri i projekteve të dështuara
- “Inxhinierimi” i softverit u mendua të jetë ilaçi:
 - Vendosja e një disipline në “programing”
 - Të bëhet më shumë se vetëm kodim/programim
 - “Studimi” (modelimi/matja), “Kuptimi” (analizimi), dhe “Përmirësimi” (ndryshimi) i kësaj fushe!

Chaos Report dhe sukseset/dështimet e projekteve softverike

- The Standish Group është një organizatë këshilluese/hulumtuese me fokus në performancën e projekteve softverike
- Chaos Report (1995) hulumtuan rreth 300 projekte softverike dhe raportuan që vetëm rreth 16% të tyre “janë përfunduar,” “me kohë,” dhe “brenda buxhetit”!

Kjo dmth që 84% e projekteve kanë dështuar!

- Chaos Report (2009) ka deklaruar që projektet softverike janë përmirësuar sepse 32% “janë përfunduar,” “me kohë,” dhe “brenda buxhetit”!

Kjo dmth që ende 68% e projekteve —dështim!

Chaos Report (2018)

- 30% projekte softverike të sukseshme
- 51% projekte softverike të sfiduara
 - (vonim, mbi buxhet, ose më pak funksionale)
- 19% projekte softverike të dështura
 - (të anuluar ose të papërdorshme)

Kjo dmth kemi ende 70% dështim të projekteve

Faktorët për sukses dhe dështim (Chaos Report)

- Karakteristikat e projekteve që kanë pasur “sukses”
 - Përfshirja e shfrytëzuesit
 - Mbështetja e menaxhmentit ekzekutiv
 - Kërkesa të qarta
 - Planifikim i mirë
- Karakteristikat e projekteve “të sfiduara” (të përfunduara dhe funksionale – por mbi buxhet dhe me vonesë)
 - Mos marrja e të dhënave nga shfrytëzuesit
 - Kërkesa dhe specifikacione jo-komplete nga shfrytëzuesit
 - Kërkesa dhe specifikacione që ndryshojnë
- Karakteristika e projekteve “të pa-afta dhe përfundimisht të anuluarat”
 - Kërkesa jo-komplete
 - Mungesë e përfshirjes së shfrytëzuesëve
 - Mungesë e burimeve (resurseve)

Dështimet e produktit softverik (Capers Jones Study)

- Gabimet në kod: 38.33%
- Gabimet në dizajn: 24.17%
- Gabimet në dokumentacion: 13.33%
- Gabimet në kërkesa: 12.50%
- Gabimet në rregullime: 11.67%

Të gjitha gabimet mund të jenë serioze dhe shumë të kushtueshme por

A duhet të kujdesemi më shumë për kodim apo për kërkesa – Pse?

Gabimet në kërkesa janë shumë të kushtueshme nëse nuk detektohen dhe lihen në produkt – Pse?

Koordinimi dhe shqetësimet jo-teknike

- Me rritjen në madhësi dhe kompleksitet të projekteve softverike, problemet tejkaluan kodin dhe softverin
- U paraqitën disa qeshtje të tjera “jo-teknike”:
 - Angazhimi dhe udhëheqja ekzekutive
 - Planifikimi tërësor i proceseve teknike dhe të biznesit
 - Personeli i aftë dhe me eksperiencë
 - Fokusi menaxherial dhe monitorimi i projektit
 - Gatishmëria për ndryshime dhe përshtatje

Raporti për senatin nga US General Accounting Office Report (2004)

- Tri strategji “kyqe” për të siguruar ofrimin e softverit:
 - (a) Qualitativ
 - (b) Në kohë, dhe
 - (c) Brenda buxhetit
- Vëmendje e fokusuar në mjedisin e zhvillimit softverik (njerëzit/veglat/menaxhimi/etj.)
- “Diciplinë” në procesin e zhvillimit
- Përdorim metodik i metrikave (matje të kuantifikueshme) për matjen e objektivave të kostos, orarit, dhe performansës funksionale

“Lindja” e Inxhinieringut softverik

- Eksperiencat e hershme në shkrimin e programeve të vështira por të vogla nuk na kanë dhënë një rrugë të qartë kur kemi filluar të ndërtojmë sisteme operative, databaza, sisteme komerciale, etj. të “mëdha”
- Çfarë na nevojitet për të zhvilluar produkte softverike të mëdha dhe komplekse dhe çfarë është e nevojshme për të kontrolluar projekte të tilla?
- Në këtë fushë nevojitet më shumë “diciplinë”
 - “SOFTWARE ENGINEERING!!!”

(NATO conference – 1968)

Çka është Inxhinierimi softverik?

- David Parnas – “ndërtimi i një softveri me shumë versione nga shumë persona”
- Sommerville – “një diciplinë inxhinierike, fokusi i së cilës është zhvillimi i sistemeve softverike të kualitetit të lartë dhe me kosto efektive”
- Pfleeger – aplikimi i veglave kompjuterike për zgjidhjen e problemeve
- CMU/SEI-90-TR-003 – “formë e inxhinierimit e cila aplikon principet e shkencës kompjuterike dhe matematikës për të arritur zgjidhje me kosto efektive për probleme softverike”
- IEEE std 610-1990 – “aplikimi i një qasjeje sistematike, të disiplinuar dhe quantifikuese në i) zhvillimin, ii) operimin, dhe iii) mirëmbajtjen e softverit.”

Inxhinierimi softverik (Tsui and Karam)

- Inxhinierimi softverik është një fushë e gjërë e cila prek të gjitha aspektet e (a) zhvillimit dhe (b) mbështetjes së një sistemi softverik, duke përfshirë këto zona kyçe:

- 1. Proceset teknike dhe biznesore***
- 2. Metodologjitë dhe teknikat specifike***
- 3. Karakterizimi i produktit dhe metrikat për matje***
- 4. Aftësitë e njerëzve dhe puna grupore***
- 5. Koordinimi dhe menaxhimi i projektit***

Rëndësia e Inxhinieringut softverik

- Softveri është një biznes serioz
 - Ka mbërri \$388.98 bilion në 2020 botërisht
 - Ka prezencë gjithpërfshirëse
- Softveri është një mall vlera e të cilit vazhdimisht rritet
- Biznesi i softverit ka evoluar prej një operimi nga “garazha” në një profesion “ndërmarrje” — përfshirë këtu “Facebook”, “Amazon”, etj.
- Ne duhet ta trajtojmë inxhinieringun softverik si një profesion inxhinierik
- 15 universitete kanë marrë akreditim (2009) nga bordi akreditues i inxhinierisë dhe teknologjisë

Profesionistët e Inxhinierimit softverik

- Ende nuk egziston titulli “inxhinier profesional” (PE) për inxhinierët softverik
 - Përveq në Texas ku bordi i inxhinierëve professional ka adoptuar inxhinierimin softverik si diciplinë specifike në të cilën mund të merret licence inxhinierike

Informacion shtesë

- Industria softverike parashikohet të arrin vlerën prej \$650 bilion në 2025
- (2014) 22 universitete në USA me programe të akredituara të inxhinierimit softverik



Pra — si mund të bëhemi më “profesional” në Inxhinierim softverik?

Raporti IEEE-CS/ACM Versioni 5.2 (1)

- 8 principet** për etikë dhe praktika profesionale në inxhinieri softverike
 - Inxhinierët e softverit do të veprojnë në përputhje me interesin publik
 - Inxhinierët e softverit do të veprojnë në mënyrë e cila është me interes të klientit dhe punëdhënësit të tyre, në përputhje me interesin publik
 - Inxhinierët e softverit do të sigurohen që produktet e tyre dhe modifikimet e këtyre të fundit plotësojnë standardet më të larta të mundshme
 - Inxhinierët e softverit do ta mbajnë integritetin dhe pavarësinë në opinionet e tyre profesionale

**** Nën secilin princip madhor, egzistojnë nën-principe të mëtutjeshme.**

Raporti IEEE-CS/ACM Versioni 5.2 (2)

- 8 principet** për etikë dhe praktika profesionale në inxhinieri softverike (vazhdim)
 - Menaxherët dhe udhëheqësit e inxhinierisë softverike duhet të promovojnë dhe të pajtohen me një qasje etike në menaxhimin e zhvillimit dhe mirëmbajtjes së softverit
 - Inxhinierët e softverit do ta avansojnë integritetin dhe reputacionin e profesionit në përputhje me interesin publik
 - Inxhinierët e softverit do të jenë të drejtë karshi kolegëve të tyre dhe ti mbështesin ata
 - Inxhinierët e softverit do të marrin pjesë në mësimin gjatë gjithë jetës lidhur me praktikën e profesionit të tyre dhe do ta promovojnë një qasje etike ndaj praktikimit të profesionit të tyre.

**** Nën secilin princip madhor, egzistojnë nën-principe të mëtutjeshme.**

Një listë më e thjeshtë e rregullave të sjelljeve

- Respekto të tjerët
- Përpiqu të jesh i drejtë (fer)
- Përdor aftësitë maksimale tuaja
- Respekto ligjin

“Principet e përgjithshme”

- Me dallim prej diciplinat e tjera inxhinierike siç janë ato mekanike and elektrike, në inxhinierim softverik nuk egziston një listë e “principeve universale” për të cilat pajtohen të gjithë
- Nuk egziston as ndonjë “ligj” i inxhinierisë softverike sikur p.sh. ligji i Njutonit për lëvizjen e trupave
- Sidoqoftë, egzistojnë disa të cilat janë të pranueshme dhe respektohen
 - Principet e Davis-it
 - Principet e Royce-it
 - Konceptet e Wasserman-it

Principet e Davis-it

- Kualiteti është i pari
- Softveri me kualitet të lartë është i mundur
- Jepju klientëve produktet para kohe
- Përcaktoje problemin para shkrimit të kërkesave
- Merr parasysh alternativa në dizajn
- Përdor model të duhur procesi
- Përdor gjuhë të ndryshme për faza të ndryshme
- Minimizo distancat intelektuale
- Vendos teknikat para veglave
- Bëje si duhet para se ta bësh më të shpejtë
- Inspekto kodin
- Menaxhimi i mirë është më i rëndësishëm se teknologjia e mirë
- Njerëzit janë çelësi i suksesit
- Përcjell me kujdes
- Merr përgjegjësinë

Principet më moderne të Royce-it

- Bazo procesin në qasjen arkitektura është e para
- Vendos process përsëritës – merru herët me rrezikun
- Thekso zhvillim të bazuar në komponente për të zvogluar punën
- Vendos menaxhim të ndryshimeve
- Përdor inxhinierim round-trip – një formë e procesit përsëritës
- Për dizajn përdor shënime të bazuara në model dhe të procesueshme nga makinat
- Vendos proces për kontroll të kualitetit dhe vlerësim të projektit
- Përdor qasje që lejon demonstrimin e hershëm të artifakteve
- Planifiko të keni lëshime (releases) inkrementale
- Vendos një proces të konfigurueshëm që plotëson nevojat

Konceptet themelore të Wasserman-it

- Abstrakimi (një formë e thjeshtimit)
- Analiza si dhe metodat dhe shënimet e dizajnit
- Prototipet e ndërfaqeve të shfrytëzuesit
- Modulariteti dhe arkitektura
- Ripërdorimi
- Cikli jetësor dhe procesi
- Metrikat
- Veglat dhe mjedisi i integruar

Këto principe adresojnë ato që përmendëm më parë

- Tri strategji “kyqe” për të siguruar ofrimin e softverit:
 - (a) Qualitativ
 - (b) Në kohë, dhe
 - (c) Brenda buxhetit
- 1. Vëmendje e fokusuar në mjedisin e zhvillimit softverik (njerëzit/veglat/menaxhimi/etj.)
- 2. “Diciplinë” në procesin e zhvillimit
- 3. Përdorim metodik i metrikave (matje të kuantifikueshme) për matjen e objektivave të kostos, orarit, dhe performansës funksionale



Pyetje???