Universiteti i Prishtinës "Hasan Prishtina" Kosovë

Inxhinierimi softverik

Pjesa 1 - Hyrje

Prof. Asoc. Dr. Ermir Rogova

Objektivat

- Analizimi i disa nga çështjeve të përfshira në krijimin e një program të thjeshtë:
  - Kërkesat (funsionale dhe jo-funksionale)
  - Kufizimet dhe vendimet e dizajnit
  - Testimi
  - Vlerësimi i përpjekjeve
  - Detalet e implementimit
- Kuptimi i aktiviteteve të përfshira në shkrimin e një programi të thjeshtë

Hyrje (Krijimi i një programi)

- Ne të gjithë "fillojmë" duke mësuar si të kodojmë një njërën nga gjuhët programuese.
  - Me një problem të vogël, hipotetik dhe të definuar mire.
  - Zakonisht kodi gjendet brenda një moduli të vetem.

- Pastaj mësojmë që programi zakonisht nuk punon herën e parë, ose të dytën - ndoshta as të pestën apo të gjashtën!
  - Mësojmë për "testimin" e programit.
  - Mësojmë për ri-leximin dhe ri-mendimin e kërkesave (të problemit) me kujdes — dhe gjejmë se mund të mos i kemi të gjitha përgjigjet.
  - Mësojmë për tracing dhe "debugging" e programit.
  - Pastaj vendosim që është "mjaft mirë!"

Deklarata e problemit

- Të paramendojmë që na është dhënë një problem i thjeshtë si në vijim:
  - "Marrë një koleksion të linjave të tekstit (strings) të ruajtura në një fajll, rënditi sipas alfabetit dhe shkruaj në një fajll tjetër."
- Kjo deklaratë nuk e specifikon problemin në tërësi. Duhet të qartësohen kërkesat në mënyrë që të krijohet një program i duhur.
- Duhen kuptuar të gjitha kërkesat e programit (program requirements) dhe kufizimet në dizajn (design constraints) të imponuara nga klienti.

Kërkesat dhe kufizimet

- Kërkesat e programit (program requirements)
  - Janë deklarata të cilat definojnë dhe kualifikojnë atë që programi duhet të bëjë
  - Ndahen në:
    - Kërkesa funksionale (functional requirements)
    - Kërkesa jofunksionale (nonfunctional requirements)
- Kufizimet në dizajn (design constraints)
  - Janë deklarata të cilat kufizojnë mënyrat se si softveri mund të dizajnohet dhe implementohet

Kërkesat funksionale

- Çka duhet të bëjë programi?
  - Në rastin tone rënditja e tekstit (me të gjitha detajet e nevojshme)
- Matja e rezultatit tenton të jetë Boolean.
  - Kërkesa është plotësuar ose jo
- Në rastin tone:
  - Formati i inputit? – latin/grek/kinez, shkronja ekstra, 1 apo 2 bajt për shkrojnë, etj
  - Rënditja? – a→z apo z→a, fillon me A apo a, A apo 0 (zero)
  - Rastet speciale? – Rreshtat e zbraztë, fajllat e zbraztë, trajtimi i gabimeve

Kërkesat jofunksionale

- Mënyra në të cilën kërkesat funksionale duhet të arrihen
- Matja e rezultatit tenton të ketë shkalë lineare
- Disa nga to janë:
  - Performanca
  - Ndryshueshmëria (modifiability)
  - Përdorshmëria (usability)
  - Konfigurishmëria (configurability)
  - Besueshmëria (reliability)
  - Disponueshmëria (availability)
  - Siguria (security)
  - Shkallzueshmëria (scalability)

Kërkesat jofunksionale – rasti ynë

- Përformanca
  - programi duhet të përfundojë të gjithë inputet brenda kohës së caktuar. <1 minutë për fajll me 100 rreshta secila me nga 100 shkronja
- Koha reale (real-time)
  - Kur programi duhet të punojë në kohë reale, performanca ka rëndësi. Përzgjedhja e algoritmit të sportimit (quick sort, heap sort, merge sort)
- Ndryshueshmëria
  - Duhet ditur sa do të jetë jetëgjatësia e programit. Sa më e gjatë ajo, aq më shumë duhet të kemi kujdes që të japim mundësi modifikimi
- Siguria
  - Clienti dhe zhvilluesit duhen të pajtohen në definimet e sigurisë që rrjedhin nga politikat biznesore të clientit. Psh. Zhvilluesi mund të kërkujë që sistemi të jetë i mbrojtur nga DOS në mënyrë që të kryej punën si duhet
- Përdorshmëria
  - Kjo kërkesë bazohet në interaksionin ndërmjet programit dhe shfrytëzuesit.

Kufizimet në dizajn

- Ndërfaqja e shfrytëzuesit (user interface)
  - Është ajo çka shfrytëzuesi shikon, ndien dhe dëgjon nga sistemi
  - GUI, CLI.
- Madhësitë hyrëse tipike dhe maximale
  - A mund të qëndrojnë në RAM apo duhet të përdret algoritëm më efektiv për lexim nga disku
- Platformat
  - Arkitektura
  - Sistemi operativ
  - Libraritë që janë në dispozicion
- Orari
  - Afati për përfundimin e projektit vjen nga klienti
  - Mund të ketë hapsirë për ndryshime por edhe ndryshime në kosto.

Vendimet në dizajn

- Hapat dhe mendimet lidhur me vendimet në dizajn për problemin tonë mund të përmblidhen si:
  - Gjuha programuese - Zakonisht ky është vendim teknik, edhepse nuk është e jashtëzakonshme të jepet si kufizim në dizajn.
  - Algoritmet – kur implementohen sisteme, zakonisht disa gjëra mund të influencohen nga përzgjedhja e algoritmeve. Në rastin tone mund të zgjedhim nga një gamë e gjërë e algoritmeve
  - Gjuha që do të përdoret dhe libraritë në dispozicion ndikojnë shumë në përzgjedhje

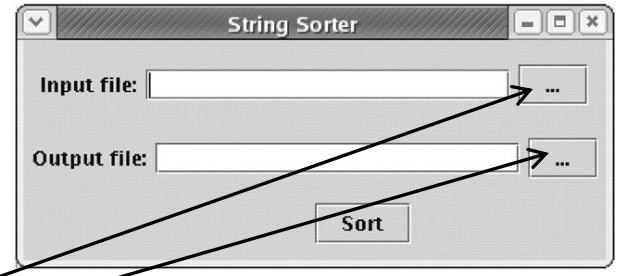
Testimet

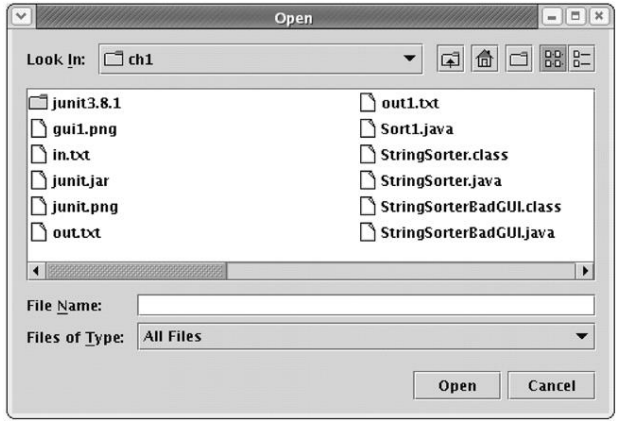
- Kur testohet
  - Derisa programi definohet
  - Derisa programi zhvillohet
  - Pasi programi të kompletohet
- Llojet e testeve
  - Pranimi (validimi)
  - Verifikimi
    - Testimi i njësive
    - Black box (kutia e zezë)
    - White box (kutia e bardhë)

Vlerësimi i mundit/punës

- Një nga aspektet më të rëndësishme të një projekti softverik është vlerësimi se sa mund/punë do të konzumojë
- Ky vlerësim nevojitet për të prodhuar parallogaritjen e qmimit dhe kohës
- Shembulli në vijim ilustron këtë pikë

Version 1. Vlerësimi i minutave në total





- Shkruaj në program i cili lexon rreshtat nga një fajll dhe i shkruan rreshtat e rënditur në një fajll tjetër. Supozo që rutinën për rënditje (sortim) do ta shkruani vetë dhe të implementoni një GUI të thjeshtë (figura 1).
- Shtypja e njërit nga dy butonat paraqet një dialog për hapjen e fajllëve (figura 2), ku shfrytëzuesi mund të navigojë sistemin e fajllave të kompjuterit dhe të zgjedhë një fajll.
- Supozo që mund të punoni vetëm në këtë detyrë, pa ndonjë ndërprerje. Brenda një minuti bëj një vlerësim.
- Hapi 1. Vlerësimi ideal i kohës totale:

  _____

Version 2. Vlerësimi i kohës kalendarike

- A është real supozimi që do të mund të punoni në këtë detyrë prej fillimit deri në mbarim, pa ndërprerje?
- A do të keni nevojë të shkoni në tualet, apo të pini ujë/kafe?
- A mund të shpenzoni kohën e duhur në këtë detyrë?
- Nëse ju kërkohet të kryeni këtë detyrë sa më parë që është arsyeshmërisht e mundur, duke filluar nga tani, a mund të vlerësoni se kur do të përfundoni?
- Po të filloni tani, vlerësoni se kur mendoni që do ta keni këtë program të gatshëm për tia dhënë klientit.
- Poashtu, bëj një vlerësim të kohës gjatë të cilës nuk do të punoni në këtë detyrë (p.sh. Ushqyerje, fjetje, lëndë të tjera, etj.).
- Hapi 2. Koha kalendarike e fillimit: _____ përfundimit:_____ pauzat:_____

Version 3. Vlerësimi i nëndetyrave

- Ndaje të gjithë programin në detyra zhvillimore të veqanta; këto detyra mund të ndahen në disa nëndetyra.
- Detyrë aktuale është planifikuese, nëndetyrë e së cilës është vlerësimi.
- Kur mendoni për kërkesat e projektit, supozoni që do të krijoni një klasë që quhet StringSorter, me tri metoda publike: Read, Write, and Sort.

- Për rutinën e rënditjes, supozoni që algoritmi juaj përfshin gjetjen e elementit më të madh, vendosjen e tij në fund të vektorit dhe pastar të vazhdojë me rënditjen e vektorin me të njëjtin mekanizëm.
- Supozoni që do të krijoni një metodë që quhet IndexOfBiggest e cila kthen indeksin e elementit më të madh në vektor.

Ideal Total Time	Calendar Time
Planning	
IndexOfBiggest	
Sort	
Read	
Write	
GUI	
Testing	
Total	

Koha aktuale në krijimin e programit

- Tani dizajnoni dhe implementoni zgjidhjen tuaj duke mbajtur shënim kohën.

	Started:	Ended:	Breaks:	Time
Planning (Estimation)				
IndexOfBiggest				
Sort				
Read				
Write				
GUI				
Testing				
TOTAL:				

Një numër i hapash të thjeshtë

- 1) Kuptoni problemin – kërkesat
  - Funksionale
  - Jofunksionale: performanca, siguria, ndryshueshmëria,etj.
- 2) Bëj një dizajn – bazuar në kërkesat
  - Organizo funksionalet në një sekuencë; mundësisht me ndihmën e ndonjë diagrami.
  - Fokusohu në hyrje/daljet (të dhënat, formatet, organizimi).
  - Mendo për disa kufizime si shpejtësia, dukja e ndërfaqes, gjuha programuese, mvarshmërisë, etc.
  - Ndonjë algoritëm specifik dhe përmirësime në sekuencën e funksionaleve.

Një numër i hapash të thjeshtë (2)

- 3) Kodo/Implemento – transformo dizajnin në kod burimor
  - Mvarësish se sa nga dizajni është kompletuar, mund ose të punohet direkt në konvertim në kod (i mvarshëm prej gjuhës) ose punohet me vazhdim të dizajnit.
    - A) Konverto hyrje/daljet në ndërfaqje apo format specifik.
    - B) Sequencoje procesimin në rënditjen e dëshiruar.
    - C) Konvertoje algoritmin procesues në konstruktin e gjuhës së përzgjedhur.
    - D) Gjej se si të përdoret në mënyrë korrekte libraria gjuhësore.

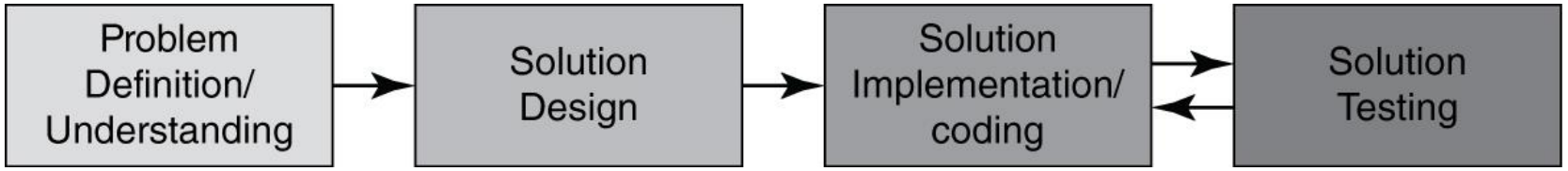
Një numër i hapash të thjeshtë (3)

- 4) Verifiko/Testo programin – kontrollo rezultatet e programit (përmes daljeve) me disa hyrje të paracaktuara.
  - Hyrjet e paracaktuara janë "raste testuese" dhe kërkojnë pak mendim.
  - Nëse rezultatet nuk përkojnë me atë që pritej:
    - "Debug" (gjej gabimet)
    - Fix (rregulloi)
    - Retest — reverify (ritesto – riverifiko)
  - Ndalu atëherë kur të gjitha rastet testuese prodhonë rezultatet e pritura.

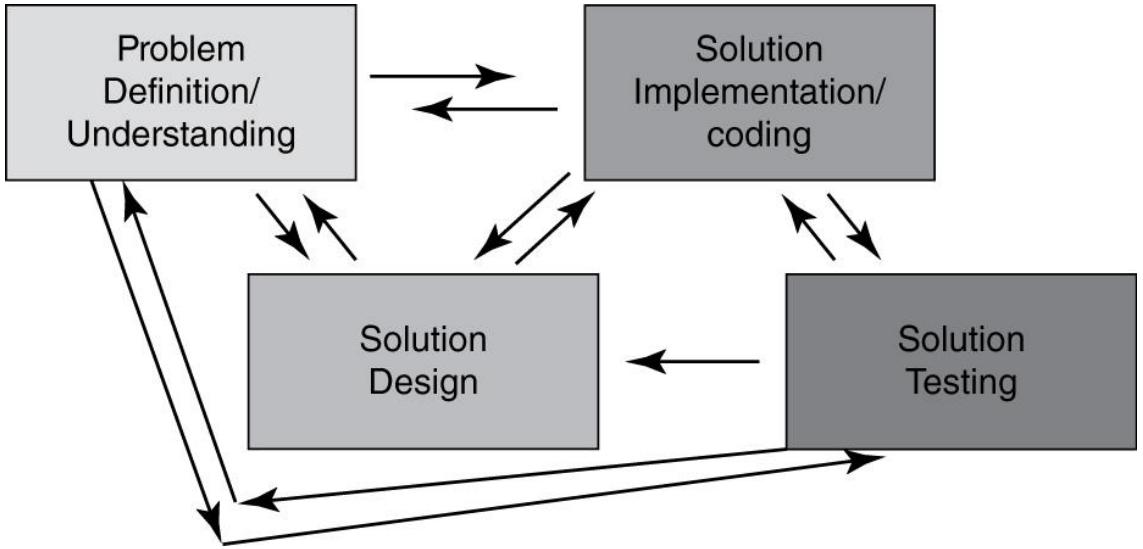
Pyetje: Sa raste testuese duhet të zhvillojmë dhe ekzekutojmë?

Çfarë ndodh me të vërtetë?

"E imagjinuar" – Situatë ideale

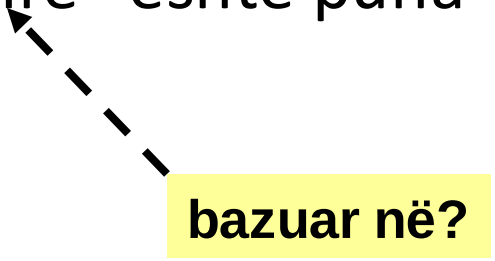


"Aktuale" – Ajo që ndodh me të vërtetë



Kodimi u krye! Çka tjetër ka rëndësi?

- Sa gjatë (kohë që ka kaluar) ka marrë për tu përfunduar puna?
- Sa mund/punë (person orë) është shpenzuar për tu kryer puna?
- A është zgjidhur tërësisht problemi?
- Sa "e mirë" është puna — (kodi, dizajni, dokumentimi, testimi, etj.)?



Konsidero një problem "të thjeshtë"

- Shkruani një "program" në gjuhën tuaj të preferuar i cili do të pranojë numra si hyrje, të llogarisë mesataren, dhe të paraqesë përgjigjen.

Përgjigjuni këtyre pyetjeve në klasë:

1. **Sa gjatë (kohë që ka kaluar) do tju merrte juve për të implementuar këtë zgjidhje?**
2. **Sa mund/punë totale (në person orë) do të merr kjo?**
3. **A do ti përshtatet problemit zgjidhja juaj?**
4. **Sa i mire është kodi/dizajni/dokumentimi/testimi juaj?**

Përgjigje reale nga ligjëratat

- Sa gjatë (kohë që ka kaluar) do tju merrte juve për të implementuar këtë zgjidhje?
  - Përgjigje në klasë: 10 min, 15 min, 1 orë
- Sa mund/punë totale (në person orë) do të merr kjo?
  - Përgjigje në klasë: 10 person min.; 15 person min.; 1 person orë; 3 person orë
- A do ti përshtatet problemit zgjidhja juaj?
  - Përgjigje në klasë: PO!
- Sa i mire është kodi/dizajni/dokumentimi/testimi juaj?
  - Përgjigje në klasë: I mrekullueshëm!

Ta shohim të dhënat "reale" individuale.

Rezultate reale

- Sa kohë mendoni që do tju merr detyra?
  - 1 orë — 7 persona
  - 2 orë — 6 persona
  - 3 orë — 2 persona
  - 10 orë — 3 persona
- The dhëna reale:
  - Koha e kaluar: prej 46 minutave deri në 5 ditë — kryesisht ndërmjet 1 dhe 3 orë.
  - Mundi/puna: prej 40 person minuta deri në 8 person orë — kryesisht ndërmjet 1 person orë dhe 3 person orë.

Pyetje???