

# Sissejuhatus tarkvaraarendusse

Martti Raavel

[martti.raavel@tlu.ee](mailto:martti.raavel@tlu.ee)

# Kolmas loeng

- Meenutame eelmist loengut
- Eelmise loengu töö
- Kuidas on läinud Githubiga?
- Kuidas on läinud ESLint tööriistaga?
- Koduste tööde näitamine ja arutelu
- Rollid tarkvaraarenduses
- Dokumentatsioon

**Kodustest töödest**

**Githubi kogemus**

**ESLint kogemus**

# Rollid tarkvaraarenduses

Tarkvara valmimise seisukohast olulisim inimeste rühm (inimene) on tellija. Tellijaks võib olla ärimees, firma, omavalitsus, asutuse teine osakond, eraisik. Tarkvaraarenduse protsessi kirjeldamisel nimetatakse seda poolt äriks (*bussiness*).

# Äripoolised rollid

- Tooteomanik (*product owner*)
- Äriarhitekt (*business architect*)
- Ärianalüütik (*business analyst*)
- Tarkvara kasutaja (*software user*)

## Tooteomanik (*product owner*)

Tooteomanik on tellija esindaja, kes tunneb vajadust tarkvara järele ning otsustab, et toode tuleb luua. Olenevalt asutuse suurusest võib tooteomanikuks olla äriomanik (*business owner*) ise või mitte.



## Äriarhitekt (*business architect*)

Äriarhitekt on inimene, kes näeb ning paneb kirja (joonestab) suure pildi, nt kaardistab tarkvaravajadused vastavas äriprotsessis.

## Ärianalüütik (*business analyst*)

Ärianalüütik aitab panna kirja, mida tahetakse saavutada (esmane tõlge inimkeelest IT-keelde).

## Tarkvara kasutaja (*software user*)

Tarkvara kasutaja on valmiva toote lõppkasutaja (võib olla ka tooteomanik).

Alati ei pruugi äri poolel äriarhitekti ja ärianalüütikut olla. Sellisel juhul võib neid ülesandeid täita vastav spetsialist projekti täitja poolel.

# Rollid arenduses

- Tarkvaraarhitekt (*software architect*)
- Süsteemianalüütik (*system analyst*)
- Programmeerija (*programmer*)
- Arendaja (*developer*)
- Tarkvarainsener (*software engineer*)
- Testija (*quality assurance tester*)
- Süsteemiadministraator (*system administrator*)
- Andmebaasi administraator (*database administrator*)
- Kasutajatugi (*helpdesk*)
- Koolitaja (*training*)
- Projektijuht (*project manager*)
- Arenduse juht (*development manager*)

## Tarkvaraarhitekt (*software architect*)

Tarkvaraarhitekt planeerib tarkvara ülesehituse: millistest osadest tarkvara koosneb, kuidase need omavahel ja välismaailmaga suhtlevad.

## Süsteemianalüütik (*system analyst*)

Süsteemianalüütik aitab "tõlkida" ärisoovid arendajatele sobivateks juhisteks.

Tarkvaraarhitekti ja süsteemianalüütiku plaanide järgi saab tarkvara loomisega alustada. Tarkvara kirjutamise ehk programmeerimisega on seotud mitmed ametinimetused. Tasub märkida, et neid nimetusi kasutatakse paindlikult, erinevalt ja teineteisega vahetatavalt, mistõttu ei ole need alati siintoodud tähendusega.



## Programmeerija (*programmer*)

Programmeerija on keegi, kes kirjutab programmi. Kõige lihtsamalt võib see olla keegi, kes instruktsioonide järgi programmi kirjutab, ise programmi ehitusele vähe mõeldes. Üldnimetusena võib see tähistada ükskõik millist programmi kirjutamisega seotud ametit.

## Arendaja (*developer*)

Arendaja on üldmõiste töötaja kohta, kes tarkvara loob. Arendustööks on vaja arusaamist programmeerimiskeelest ja nõuetest, kuid keerukamate ülesannete jaoks võib olla tarvis laialdasemaid teadmisi. Oskuste, kogemuste ning vastutuse põhjal eristatakse vahel nooremarendajat (*junior developer*), kellele usaldatakse enamasti kergemad ülesanded, ja vanemarendajat (*senior developer*), kes saab hakkama nõudlikumate töödega.

## Tarkvarainsener (*software engineer*)

Tarkvarainsener tunneb programmeerimisega seotud infotehnoloogilisi võimalusi (raamistikke, malle, arhitektuure, protsesse, tehnikaid jne.) ning oskab neid otstarbekohaselt rakendada. Tihti võib see kattuda arendaja või vanemarendaja töökirjeldusega.

## Testija (*quality assurance tester*)

Testija kontrollib, et tarkvara teeks seda, mida ta peab tegema, ehk et ärisoovid oleksid täidetud ning tarkvaras ei oleks vigu. Seda rolli saab võrrelda järelevalveametniku rolliga ehituses.

## Süsteemiadministraator (*system administrator*)

Süsteemiadministraator vastutab arvutisüsteemide eest; hangib, paigaldab ja uuendab tarkvara ja riistvarakomponendid, ehk kannab hoolt, et kõik süsteemid oleksid töökorras.

## Andmebaasi administraator (*database administrator*)

Andmebaasi administraator haldab andmebaasi, vastutab andmete säilitamise, rakenduse töötamise eest.

## Kasutajatugi (*helpdesk*)

Kasutajatugi aitab kasutajatel tarkvara kasutada.

## Koolitaja (*training*)

Koolitaja õpetab lõppkasutajat tarkvara kasutama, vajadusel vastutab kasutamiseks vajalike dokumentide loomise eest. Ka maja ehitamisel saab omanik palju "koolitajatega" kokku.



## Projektijuht (*project manager*)

Projektijuht juhib projekti töökorraldust, hoolitseb lõpptulemuse saavutamise eest, jälgib eelarve täitmist.

## Arenduse juht (*development manager*)

Arenduse juht koordineerib koostööd projektide vahel, saab projekti katkestada

# Dokumentatsioon

Tarkvaraarenduse dokumentatsioon viitab kirjalike või visuaalsete materjalide loomise ja korrastamise protsessile, mis kirjeldavad tarkvaratoote või -süsteemi disaini, arendust ja kasutamist.

## Tehnilised kirjeldused

Dokumendid, mis kirjeldavad tarkvarasüsteemi, sealhulgas andmestruktuuride, algoritmide ja liideste kavandamist ja rakendamist.

## Kasutusjuhendid

Dokumendid, mis kirjeldavad tarkvaratoote kasutamist, sealhulgas installi-, konfiguratsiooni- ja kasutusjuhiseid.

# Väljalaskemärkmed

Dokumendid, mis kirjeldavad tarkvaratoote iga versiooni muudatusi ja täiustusi.

## Koodikommentaarid

Kommentaarid, mis lisatakse tarkvaratoote lähtekoodile, et kirjeldada, kuidas koodi erinevad osad töötavad ja kuidas need on kasutamiseks mõeldud.

# API dokumentatsioon

Dokumendid, mis kirjeldavad tarkvarasüsteemi poolt avaldatavaid liideseid, sealhulgas iga liidese sisendeid, väljundeid ja käitumist.



# Dokumentatsiooni loomine

Tarkvaraarenduse dokumentatsiooni koostamine võib olla keeruline protsess, kuna olenevalt projekti vajadustest võib olla vaja koostada palju erinevat tüüpi dokumentatsiooni. Siin on aga mõned üldised sammud, mida saab tõhusa tarkvaraarenduse dokumentatsiooni loomiseks järgida

## **Tehke kindlaks dokumentatsiooni eesmärk ja ulatus**

Enne dokumentatsiooni loomise alustamist on oluline kindlaks teha, millist tüüpi dokumentatsiooni on vaja, kes on sihtrühm ja millised on dokumentatsiooni eesmärgid.

## Dokumentatsiooni kavandamine

See samm hõlmab dokumentatsiooni struktuuri, vormingu ja sisu määratlemist. Peaksite kaaluma dokumentatsiooni (nt tekst, diagrammid või videod) parimat vormingut ja otsustama, milline sisu tuleb kaasata.

## Dokumentatsiooni kirjutamine

Kui planeerimine on lõpetatud, võite alustada dokumentatsiooni kirjutamist. Kasutage selget ja ülevaatlikku keelt ning veenduge, et dokumentatsioon oleks kergesti loetav ja arusaadav. Peaksite lisama ka kõik asjakohased diagrammid, pildid ja viited, mis aitavad sisu selgitada.

## Dokumentatsiooni testimine ja ülevaatamine

Oluline on dokumentatsiooni testida, et olla kindel, et see on täpne ja hõlpsasti jälgitav. See võib hõlmata seda, et teised arendajad, sidusrühmad või lõppkasutajad vaatavad tagasiside andmiseks dokumentatsiooni üle.

## Dokumentatsiooni värskendamine ja hooldamine

Tarkvaraarendus on pidev protsess, mistõttu on oluline dokumentatsiooni projekti edenedes uuendada ja säilitada. See võib hõlmata uue teabe lisamist, olemasoleva teabe värskendamist ja aegunud teabe eemaldamist.

# Näpunäiteid

- Kasutage kogu dokumentatsioonis ühtset vormingut ja stiili
- Keeruliste mõistete selgitamiseks kasutage diagramme ja pilte
- Hoidke dokumentatsioon ajakohasena
- Kasutage selget ja kokkuvõtlikku keelt
- Kaaluge sihtrühma vajadusi ja vaatenurki

# Kodune töö

- Lisage oma eelmistele kodutöödele nimed
- Koolile dokumentatsiooni loomise eeskirjad:
  - Tehniline dokumentatsioon
  - Kasutusjuhendid
  - Väljalaskemärkmed
  - API dokumentatsioon
  - Issue