

Tallinna Ülikool
Informaatika Instituut

Avatud lähtekoodiga projekti loomine ja võimalused

Seminaritöö

Autor: Timo Vainu

Juhendaja: Kaido Kikkas

Autor: „ „ 2009

Juhendaja: „ „ 2009

Instituudi direktor: „ „ 2009

Tallinn 2009

Sisukord

Sisukord.....	2
Sissejuhatus	4
1. Avatud lähtekoodiga tarkvara	5
1.1 Avatud lähtekoodiga tarkvara kriteeriumid.....	6
1.2 Avatud lähtekoodi litsentsid.....	6
1.3 Avatud lähtekoodi kogukonnad	7
1.3.1 Liikmed	7
1.3.2 Motivatsioon.....	7
2. Koodi avalikustamise võimalused ja eelised.....	8
2.1 Kvaliteet	8
2.1.1 Distsipliin	8
2.1.2 Väljaspoolne abi.....	8
2.2 Vestlused	9
2.2.1 Suhtluskanalid	9
2.2.2 Täiendavad suhtluskanalid	9
2.3 Levimine.....	10
2.3.1 Projekti levitamine	10
2.3.2 Uue tehnoloogia levitamine	10
2.4 Turg	11
2.4.1 Turu ehitamine	11
2.4.2 Turg varalistele toodetele	11
2.5 Kasutajatepoolne abi	12
2.5.1 Projekteerimise abi.....	12
2.5.2 Arendamise abi.....	12
2.5.3 Kasutajatugi.....	13
2.6 Standardite areng.....	13
2.7 Piiratud ressursid	14
3 Avatud lähtekoodiga projekti algatamine.....	15
3.1 Vajalik infrastruktuur avatud lähtekoodi jaoks	15
3.1.1 Avaliku koodi arhiiv.....	15
3.1.2 Projekti dokumentatsioon.....	16
3.1.3 Vigade andmebaas.....	16

3.1.4 Avatud meililist	17
3.1.5 Projekti veebileht.....	17
3.2 Tarkvara protsess.....	17
3.2.1 Planeerimine.....	18
3.2.2 Arendajate panuste ühendamine lähtekoodiga	18
3.2.3 Järgud	18
3.2.4 Testimine	19
3.2.5 Väljalasked	19
3.2.6 Uue mooduli või alamprojekti lisamine	20
4. Soovitused	21
Kokkuvõte	22
Allikate loend	23

Sissejuhatus

Kui viieteist aasta eest oli avatud lähtekoodiga tarkvara vaid üksikute ringkondade ning IT-meeste huviobjekt, siis nüüdseks on ajad muutunud ning avatud lähtekoodiga tarkvara on hakkanud tänapäevases maailmas tugevalt kanda kinnitama.

Siiski tekitab avatud lähtekoodiga tarkvara inimestes kahtlusi. Kas saab usaldada tarkvara, mille ärilised eesmärgid on tihti paljudele arusaamatud ning mille arendamisele kaasa aitavad vabatahtlikud oma töö ja ajakulu eest rahalist kompensatsiooni ei saa?

Käesoleva seminaritöö eesmärk on välja selgitada, millised võimalused kaasnevad avatud lähtekoodiga tarkvara arendamisel ettevõtetele ning mida on vaja avatud lähtekoodiga tarkvara loomiseks.

Omapoolse panusena koostab autor nimekirja soovitustest, mille järgimine oleks aluseks edukale avatud lähtekoodiga tarkvara projektile.

1. peatükis vaatame, mis teeb avatud lähtekoodist avatud lähtekoodi ja millised on avatud lähtekoodi litsentsid. Tutvume ka avatud lähtekoodi kogukondadega ja vaatleme põhjuseid, miks inimesed võtavad osa avatud lähtekoodiga projektidest.

2. peatükis vaatame, milliseid võimalusi ja eeliseid pakub avatud lähtekoodiga projekti arendamine. Tutvume nii projekti kvaliteeti suurendavate asjaoludega, suhetega inimeste ja ettevõtete vahel, vaatleme kuidas projekt levib, millised võimalused on projektil turul ja tutvume ka kasutajatepoolse abiga projektile.

3. peatükis vaatame, mida läheb vaja avatud lähtekoodiga tarkvara projekti alustamiseks ja kuidas see käib. Tutvume põhjalikult vajamineva infrastruktuuriga ja uurime arenguprotsessi etappe.

4. peatükk koosneb antud seminaritöö autori koostatud soovitustest avatud lähtekoodiga projektile.

1. Avatud lähtekoodiga tarkvara

Avatud lähtekoodiga tarkvara on vägagi perspektiivikas suure hulga inimeste töö tulemus, mis on levinud tänapäeval nii üksikisikute, väikeettevõtete kui ka hiiglaslike äriimpeeriumite seas (näiteks Apache on enimkasutatud veebilehe server). [Wheeler] Avatud lähtekoodiga tarkvara fenomen seisneb selles, et avatud lähtekoodiga tarkvara lähtekood on kõigile kasutajatele avalikult nähtav. Lähtekood avalikustatakse koos tarkvaraga, igaüks saab seda omale sobival moel ilma piiranguteta kasutada, kopeerida, uurida, muuta ja levitada.

1.1 Avatud lähtekoodiga tarkvara kriteeriumid

Antud seminaritöös räägime avatud lähtekoodiga tarkvarast. Et tarkvara saaks lugeda avatud lähtekoodiga tarkvaraks, peab ta vastama järgmistele kriteeriumitele [OSI]:

- on vabalt levitav;
- lähtekood on kättesaadav;
- modifitseerimine ja edasiarendamine on võimalik;
- muudetud tarkvara levitamist võib keelata ainult siis, kui on lubatud tarkvara muutmiseks luua paikasid ja levitada tarkvara originaalversiooni koos paikadega;
- puuduvad kitsendused isikutele ja gruppidele;
- puuduvad kasutuspiirangud;
- kasutuslitsents on piiramatu;
- litsents ei tohi olla produktipõhine;
- litsents ei tohi piirata teisi tarkvarasid;
- litsents peab olema tehnoloogia suhtes neutraalne.

1.2 Avatud lähtekoodi litsentsid

Täpsemad avatud lähtekoodiga tarkvara kasutusõigused kirjeldab ära selle valitud litsents. Stallmani järgi on kõigile litsentsidele omane põhimõte, et igal inimesel on õigus lähtekoodi kasutada, kopeerida ja jagada. Woods ja Guliani peavad litsentsi tingimuste kattuvust ja kokkusobimatuste mõistmist väga tähtsaks aspektiks, millega tuleks igal ettevõttel avatud lähtekoodiga projekti alustamisel arvestada.

Tutvustan lühidalt olemasolevaid tuntumaid avaliku lähtekoodi litsentse:

- **GPL (GNU General Public License)** - GNU Üldine Avalik Litsents. Sisaldab copyleft-printsiipi. Copyleft annab õiguse litsentsiobjekti vabalt kasutada, paljundada ja muuta senikaua kuni kasutaja ei hakka saadud õigusi omaltpoolt kitsendama. [Stallman]
- **LGPL (GNU Lesser General Public License)** - Lubab seda litsentsi kasutavaid programmeerimiseke ehk alamprogrammide kogumikke ühendada mittevabade rakendusega. Muus osas sarnaselt GPLi'ga kehtib copyleft-i põhimõte. [Kikkas]
- **BSD (Berkeley Software Distribution License)** - Vaba tarkvara litsents, GPL litsentsi järel üks populaarsemaid. Tegemist on enimkasutatava mitte-copyleft litsentsiga. [Kikkas]
- **Apache (Apache License)** - Peamiselt loodud Apache veebiserveri jaoks 1995 Aastal. Jaanuaris 2004 kinnitati Apache License v 2.0, mis Woodsi ja Gualiani kohaselt annab litsentsi valdajale kõik autori- ja patendiõigused.
- **MPL (Mozilla Public License)** - Loodud Netscape Communicationsi poolt, mil nad loobusid oma brauseri müügist. Tänapäeval üks populaarsemaid alternatiive GPL-ile. [Woods&Guliani]

1.3 Avatud lähtekoodi kogukonnad

Edukas avatud lähtekoodiga projekt on kogukonna töö tulemus. [Goldman&Gabriel]

1.3.1 Liikmed

Woodsi ja Guliani sõnul võib avatud lähtekoodiga kogukonna struktuur olla erinev. Kogukond võib omada projektijuhti, kes kordineerib kogu tegevust. Projektijuht on sealjuures tihti ka peamiseks arendajaks. Osades kogukondades puudub projektijuht ja on ainult arendajad. Projekti edenedes kujunevad aga osadest arendajatest välja liidrid. Lisaks liidritele koosneb kogukond erinevate vajaduste ja oskustega inimestest:

- kasutajad
- kujundajad
- programmeerijad

Iga kogukonna liige täidab kogukonnas erinevat rolli. Kõik liikmed kokku moodustavadki terviku – kogukonna. Kommunikatsiooni kogukonna liikmete vahel võib lugeda kogukonna kõige tähtsamaks koostisosaks. [Lakhani&Wolf]

1.3.2 Motivatsioon

Lakhani ja Wolfi sõnul ei saa 87% inimestest avatud lähtekoodiga projektis osalemise eest mingit finantsilist kompensatsiooni. Seega tekib küsimus – miks nad seda teevad? Järgnevad motivatsioonipõhjused on refereeritud Goldmani ja Gabrieli raamatust.

- **Vajadus toote järele** - luua, kohandada või täiustada toodet enda ja teiste vajaduste järgi.
- **Nauding, lõbu ja tahtmine midagi luua ja täiustada** - inimesed leiavad, et tarkvara arendamine on loominguline ja huvitav.
- **Ühtekuuluvustunne** - inimestele meeldib suhelda ja veeta aega mõttekaaslastega.
- **Väärtused ja ideoloogia** - levitamaks kindlaid ideaale nagu näiteks vaba tarkvara filosoofia.
- **Harjutamine, õppimine, reputatsioon väljaspool kogukonda** - arendamiseks oma oskusi lootused, et selline enesetäiendamine avab teed paremale töökohale.
- **Õiglustunne** - et anda omaltpoolt midagi tagasi programmi kasutamise või kogukonnalt abi saamise eest.
- **Lootus muuta asju paremaks** - leida või luua juba olemasolevatele paremaid lahendusi.
- **Tagasiside** - saada kommentaare ja juhiseid oma töö ja oskuste kohta.

2. Koodi avalikustamise võimalused ja eelised

Avaliku lähtekoodiga projekt saab alguse inspiratsioonist lahendamaks probleemi, mis on piisavalt suur ja levinud, et meelitada teisi inimesi kogukonnaga liituma ja töötama projekti kallal tasuta. [Woods&Guliani] Enne kui otsustada avatud lähtekoodi kasuks, tuleks tutvuda avatud lähtekoodiga kaasnevate võimaluste ja eelistega projekti tulevikku ja eesmärke silmas pidades.

Käesolevas peatükis käsitleme lähemalt avatud lähtekoodi eripäraseid ning vaatleme võimalusi, mida koodi avalikustamine pakub. Illustreerimaks seda, kasutan Goldmani ja Gabrieli poolt toodud näiteid NetBeansi, OpenOffice'i, Gini, Java.net'i ja teiste projektide kohta.

2.1 Kvaliteet

2.1.1 Distsipliin

Kõik ettepanekud kavandamise ja rakendamise kohta avatud lähtekoodiga projektis on esitletud kirjalikult. Kogukond seejärel arutleb tehtud ettepaneku üle ja lõppotsus on kokku võetud samuti kirjalikult. Kirja pandud materjalid lähevad arhiivi ja on avalikud kõigile. Sellisel tasemel distsipliini rakendades kasvab kavandamise ja rakendamise kvaliteet. Kõige üleskirjutamine loob vajaduse väljendada probleeme selgel viisil, mis sageli viib põhjuste parema mõistmiseni. Parem mõistmine aga omakorda viib suurema kvaliteediga kavandamiseni. [Goldman&Gabriel]

2.1.2 Väljaspoolne abi

Väljalasete kättesaadavus kogukonnale juba varajases arendamisjärgus suurendab vigade leidmise tõenäosust. Kasutajad ja arendajad annavad leitud vigadest teada ja need parandatakse teiste arendajate poolt või siis võib vea leidja omades piisavalt oskusi ise vea parandada. Sedasi saab parandada hulga vigu, mis muidu jänuks parandamata ja oleksid alles hiljem välja tulnud. Väljaspoolsed arendajad võivad samuti lisada väikseid funktsioone mida tuumikrühm soovib programmis näha, säästes aega. Nii saab tuumikrühm keskenduda olulisemale ja raskemale tööle. Kõik see väljendub ametlike väljalasete suuresti paranenud kvaliteedis. [Goldman&Gabriel]

2.2 Vestlused

Lakhani ja Wolfi sõnade järgi on kasutaja ja ettevõtte omavaheline suhtlemine väga tähtis. Ettevõtte saab areneda korraldades vestlusi oma klientidele ja potentsiaalsetele klientidele. Sellised vestlused tugevdavad suhteid kasutajate ja ettevõtete vahel ja loovad võimalusi ettevõtetele sidemete tekkimiseks teiste ettevõtetega.

2.2.1 Suhtluskanalid

Goldman ja Gabriel on oma raamatus kirjeldanud järgnevaid suhtluskanaleid:

- **Listid**

Listid on vanim ja levinuim avatud lähtekoodi arendajate suhtlemise moodus, mida kasutatakse siiani väga paljudes avatud lähtekoodi projektides.

- **Foorumid**

Tänapäevane alternatiiv listidele. Kuulates hoolikalt inimeste vestlusi avalikus foorumis avatud lähtekoodiga projekti üle, võib ettevõtte saada teada üsnagi palju uut ja huvitavat. See teave võib sageli mõjutada uusi tooteid ja toote suundi.

- **Üritused**

Jutuajamistest on teinekord rohkem kasu kui nad toimuvad näost näkku. Sponsoreeriv ettevõtte võib korraldada kogukonna kokkutulekuid, kus kogukonna liikmed saavad võimaluse rääkida oma projektidest ja tööst. Sedasi kuulates kõike probleemide ja huvitavate võimaluste kohta millega kogukonna liikmed seisavad silmitsi, on võimalus saada väärtuslikke ideid uute projektide jaoks.

2.2.2 Täiendavad suhtluskanalid

Avatud lähtekoodiga projekti iseloom on nähtav kõigile. Olles nähtav strateegiliselt tähtsale ettevõttele, olenemata kas see ettevõtte võtab osa projektist või mitte, saab sellele ettevõttele paremini esitleda plaane, eesmärke, funktsioone ja avaldusi arengust. Avatud lähtekood teeb kättesaadavaks täiendavaid suhtluskanaleid ettevõtete vahel. [Goldman&Gabriel]

- **NetBeansi** projekt võimaldab Suni arendajatel oma projektiga tegeleda kasutades mitmesuguseid Java uudistegruppe ja erinevaid veebilogisi otseseks suhtlemiseks

teiste arendajatega. See on hea võimalus saada tagasisidet kogukonnalt tehniliste küsimuste kohta.

2.3 Levimine

2.3.1 Projekti levitamine

Goldman ja Gabriel on kirjeldanud projekti levitamist läbi kogukondade. Ettevõtte alustab pakkumise tegemisest kõigepealt projekti otsestele kasutajatele ja arendajatele. Seejärel saab pakkumine levida juba teistesse kogukondadesse ja kogukonna liikmete tuttavatele. Taoline levitamine on kuludeta ja avatud iseloomu näitav.

- **NetBeansi** selge eesmärk oli projekti levitamist silmas pidades luua kogukond oma integreeritud arenduskeskonna platvormile, mis oleks ühest kolmest kõige populaarsemast Java arendamise keskkonnast.
- **OpenOffice'i** kõikjalelevimine sisaldab suurt ja mitmekesist kasutajate baasi. Isik võib minna aadressile OpenOffice.org ja alla laadida kõige uuema versiooni ning seejärel osaleda maililistide arutluses kuidas saada kõige paremini ära kasutada OpenOffice'i rakendusi ja mida annaks parandada. Kuna sellele heale tundele mis tekib osaledes kogukonna sisestes vestlustes, võib inimene tõenäoliselt soovitada OpenOffice tarkvara teistele, aidates kaasa levikule.

2.3.2 Uue tehnoloogia levitamine

Avatud lähtekood on hea viis uue tehnoloogia levitamiseks. Projekt on atraktiivne inimestele ja nad tulevad õppima uut tehnoloogiat, tehes sellega asju ja vaadates mida teised sellega teevad. Inimesed, kes löid tehnoloogia on kättesaadavad küsimustele ja suunavad tehnoloogia arengut, põhinedes kogukonna tagasisidel. [Goldman&Gabriel]

- **Jini** projekt on näide sellest, kuidas kogukonna jõugpingutused võivad aidata tutvustamaks paljudele teistele arendajatele uut tehnoloogiat. Selleks lõi Jini meeskond uue elujõulise Jini kogukonna paljudele väljaspoolsetele arendajatele, kes üritasid õppida Jini tehnoloogiat.

2.4 Turg

Levitades head ja suure kasutajaskonnaga avatud lähtekoodiga platvormi, tekib võimalus luua turg lisamoodulite, toe ja teiste avatud lähtekoodiga seotud toodete ja teenuste jaoks.

2.4.1 Turu ehitamine

Turu ehitamine võib nõuda mitmeid protsesse, kasutades Goldmani ja Gabrieli järgi vähemalt kahte erinevat lähenemist:

- **Luua avatud lähtekoodi kogukond ümber tarkvara projekti.** Selline projekt võib olla seotud näiteks kolmandate isikute poolt renditud müügiks oleva tarkvaraga. NetBeans on integreeritud arenduskeskond, mis võimaldab programmiliidestele vastavaid mooduleid neisse sisse ühendada. Sellised moodulid on heaks kaubaks turul.
- **Ehitada üles infrastruktuur avatud lähtekoodiga projektidele, mis asuvad ümber ettevõtte poolt turule paisatava tehnoloogia.** Sun on ehitanud taolise infrasturkuuri ja kogukonna nimega Java.net. Java.net on nii rühm kogukondasi, mis loovad Java tarkvara, kui ka veebileht mis sisaldab infrasturktuuri võõrustamiseks neid kogukondasi.

2.4.2 Turg varalistele toodetele

Lisaks tasuta versioonile ehitatud tasulistele lisateenustele, saab ettevõtte avaldada tarkvara avatud lähtekoodiga tarkvarana - kasutades litsentsi, nagu GPL, mis nõuab, et iga suurem avatud lähtekoodil põhinev töö oleks samuti avatud lähtekood - ja siis litsentseerida tarkvara neile ettevõtetele, kes soovivad kasutada seda tarkvara oma varalises tootes. [Goldman&Gabriel]

- Sedasi talitab näiteks ettevõtte Trolltech AS oma Qt mitmeplatvormiliste rakenduste arendamise raamistikuga KDE - graafilisel kasutajaliidesel põhinev paljude lisaprogrammidega töölauakeskond - mis on aluseks Linuxile.
- Sama strateegiat kasutab ka MySQL AB oma andmebaasi toodetele.

2.5 Kasutajatepoolne abi

Antud seminaritöös välja toodud peatükkidest võib abi lugeda kõige olulisemaks. Paljud tarkvaratooted ebaõnnestuvad, kuna nad ei suuda täita kasutajate nõudmisi. Tervelt kolm neljandikku kõigist suurtest süsteemidest on äpardumised, mis ei toimi nii nagu ette nähtud või mis ei ole üldse kasutuses. [Gibbs] Üle 30% tarkvara projektidest tühistatakse, suurima põhjusena vähene kasutajapoolne sekkumine. [The Standish Group]

2.5.1 Projekteerimise abi

Tehes projekteerimise protsess kogukonnale nähtavaks, ei saa projekt mitte ainult nõu projekteerimise suundade kohta, vaid ka abi projekti teostamisel selles suunas. [Goldman&Gabriel]

- Kogukonna tagasiside NetBeansi kasutamise suundade kohta tõi tulemuseks uute suundade lisamise NetBeansi projekti. Nüüd NetBeans toetab ka arendajaid, kes soovivad lisada rakendusi NetBeansi alusraamistikule. See uus lähenemine on aidanud täiendada NetBeansi integreeritud arenduskeskonda.

2.5.2 Arendamise abi

Projekt võib loota abile, kui projekti kood on suhteliselt modulaarne ja ei nõua nädalaid õppimist selle muutmiseks. Kasutaja võib osutada projektile mitmesugust abi:

- vigade leidmine
- vigade parandamine
- tarkvara liigutamine ühelt platvormilt teisele
- uute rakenduste loomine
- uute moodulite loomine

Goldmani ja Gabrieli järgi tuleb tähele panna, et sellised muudatused ei pruugi olla need rakendused, mis olid planeeritud järgmise väljalaske jaoks, vaid on hoopis midagi väljaspoolsete arendajate individuaalsetele vajadustele vastavat. Selliseid panused võivad aga sellegipoolest olla kasuks tulevikus laiendamaks projekti ulatust.

- **NetBeans** on struktureeritud hulga moodulitena mis kommunikeerivad läbi programmiliideste. Üks eesmärk oli luua kogukond moodulite arendajatele, ja see on ka teoks saanud. Väljaspoolsed arendajad on tõepoolest loonud uusi mooduleid ja jätkanud nendega töötamist.
- **OpenOffice'i** kogukond on olnud tõhus arendamaks selle tohutud koodibaasi - lisades uusi mooduleid koodi kirja kontrolliks ja kompileerimiseks. Sellest on olnud suur abi OpenOffice'le paremal sissesulandumisel Linuxisse.

2.5.3 Kasutajatugi

Kogukonna liikmed koostavad nimekirja korduma kippuvatest küsimustest - FAQ - Frequently Asked Questions - ja lisavad selle projekti veebilehele, tänu millele saavad algajad kasutajad otsida vastuseid lihtsamatele küsimustele. Selline struktuur aitab inimestel üksteist aidata, lihtsamad küsimused saavad vastatud ja projekti liidrid saavad vajadusel tegeleda keerulisemate küsimustega. [Woods&Guliani, 2005] Tänu kogu sellele kasutajate endi poolt ära tehtud tööle ei pea ettevõtte sellele oma aega kulutama.

- **OpenOffice'i** kogukond on kirjutanud "kuidas-teha" juhiseid, andmaks ülevaadet kuidas teha mingit ülesannet OpenOffice'ga. Kogukond on kokku pannud nimekirja FAQ'si ja on loonud näidis ja mallifaile, mis demonstreerivad OpenOffice'i võimekust ja kuidas täita teatud ülesandeid. Selline abi on tehtud kättesaadavaks mitmetes keeltes.

2.6 Standardite areng

Mõned projektid on seadnud eesmärgiks arendada standardeid - programmiliideste definitsioone, keele laiendeid või vahendite laiendeid. Protsessi avalikustamine igapäevale muudab palju tõenäolisemaks parima nõuande leidmise ja suurema omaksvõtu saamise. [Goldman&Gabriel]

- **NetBeansi** integreeritud arenduskeskonna programmiliidesed on disainitud avatud viisil alustamiseks, seega avatud lähtekoodi kasutamine lihtsalt säilitab seda tava.
- **OpenOffice'i** eesmärk oli luua XML standard failivormingutele.
- **Jini** kogukond on osaliselt organiseeritud ümber standardeid andva keha. Kuna kogukond koosneb üksikisikustest ja ettevõtetest, on organisatsioonil kaks koda.

Ühes on igal inimesel üks hääl ja teises on igal ettevõttel üks hääl. Iga ettepanekule kuuluv standard peab saama mõlema koja heakskiidu, enne kui ta kogukonnas vastu võetakse. Arutelud selles firmas kavandatavate standardite üle on aidanud kogukonnal edukalt panna ettevõtteid koos töötama.

2.7 Piiratud ressursid

Mõnikord ei pruugi olla piisavalt vahendeid, et teha ära kogu vajaminevat tööd omapäi.

Goldmani ja Gabrieli on toonud välja avatud lähtekoodiga seotud võimalusi:

- **Ehitada projekt juba olemasoleva avatud lähtekoodiga projekti peale.** Kui teha tahetav töö lisab väärtusi juba tehtud tööle ja sobib selle nägemusega, siis selle projekti kogukond on kindlasti rahul antava panusega.
- **Luuu uus avatud lähtekoodiga projekt või avatud kogukond** - tingimusel, et saadakse tööle teisi ettevõtteid ja individuaale kes jagavad visiooni ja on nõus tegema tööd, et visioon ellu viia. Viis, kuidas Java on välja töötatud, on sellele strateegiale heaks näiteks. Läbi Java Community Process'i, uued lisandused Javale on ühiselt arendatud teiste projektis osalevate äriettevõtete poolt, kes kirjutavad vajalikke kirjeldusi, viitavad rakendusi ja tehnoloogia ühilduvus komplekte.

3 Avatud lähtekoodiga projekti algatamine

Selles peatükis heidame pilgu peamistele tegevustele, mida läheb tarvis avatud lähtekoodiga projekti loomiseks. Tutvume nii vajamineva infrastruktuuriga, kui ka avatud lähtekoodiga tarkvara loomise protsessi etappidega.

3.1 Vajalik infrastruktuur avatud lähtekoodi jaoks

Iga avatud lähtekoodiga projekt näeb ette avaliku koodi arhiivi olemasolu juurdepääsuks lähtekoodile. Samuti ka dokumentatsiooni selle kohta, kuidas kasutada ja kuidas muuta koodi, meililiste ja uudistegruppe küsimuste arutamiseks, vigade andmebaasi ja veebilehte juurdepääsu tagamiseks kõigile eelnimetatud teenustele. Need nimetatud elemendid on skeletiks heale ning toimivale avatud lähtekoodiga projektile.

3.1.1 Avaliku koodi arhiiv

Avaliku koodi arhiiv on projekti kõige tähtsam osa. Goldman ja Gabriel on rääkinud järgmistest nõuetest avaliku koodi arhiivile:

- Peamine nõue avatud lähtekoodiga projektile on, et kood oleks avalik.
- Igale arendajale peab olema võimalik kätte saada koodi viimane versioon igal ajal. Olenemata sellest, kas arendaja on ettevõttesisene või väljaspoolne.
- Mooduli eest vastutaval arendajal peab olema võimalik teha muutusi otse lähtekoodile.
- Toetused ja veaparandused arendajatelt, kellele ei ole veel antud kirjutamisjuurdepääsu lähtekoodile, tuleb integreerida lähtekoodi arhiivi õigeaegselt.
- Avatud lähtekoodi järke tuleb teha sageli, kui võimalik igapäevaselt. Need peavad olema kättesaadavad allalaadimiseks arendajatele ja kasutajatele. Kasutajatel peab alati olema võimalik allalaadida koodi töötavat versiooni.

CVS - Concurrent Versioning System – on enimkasutatud versioonihaldussüsteem avatud lähtekoodiga projektide seas. [Robbins] Seda kasutades saab kasutaja salvestada koodi ja dokumentide ajalugu. [CVS] CVS ise on samuti avatud lähtekoodiga projekt ja on seega saadaval tasuta aadressil <http://www.nongnu.org/cvs/>.

3.1.2 Projekti dokumentatsioon

Lohakas dokumentatsioon näitab üles lugupidamatust koodi järgmise kasutaja suhtes. [Woods&Guliani] Avatud lähtekoodiga projekt nõuab head sisemist dokumentatsiooni arendajate jaoks, et arendajatel oleks võimalikult kerge õppida tundma lähtekoodi. Järgnevalt toome ära olulisemad dokumentatsioonide liigid.

- **Meililisti arhiivid** on heaks võimaluseks arendajatele õppida mitmete varemsooritatud otsuste kohta.
- **Projekti tegevuskava** (*roadmap*) on tähtis dokument, mis kirjeldab praeguseid arenguplaane üldise projekti ja üksikute moodulite kohta. Paljud arendajad otsustavad projekti tegevuskava järgi, mille kallal nad hakkavad tööd tegema.
- **Soovidenimekiri** (*wish list*) on koht kus kasutajad saavad rääkida oma soovidest projekti kohta, osaledes sedasi projekti kujundamises.
- **Korduma Kippuvad Küsimused** (*FAQ*) loomine oleks esimeseks sammuks kasutajate dokumentatsiooni loomisele. [Goldman&Gabriel]
- **Viki** (*Wiki*) on veebilehestik, kus kasutajad saavad lehe sisu muuta läbi veebibrauseri. Viki mootorid on kasutuses paljudes avatud tarkvara projektides, olles kõigile kättesaadav üle veebi. Muudatuste tegemiseks ei lähe vabatahtlikel vaja kliendipoolset tarkvara janad võivad teha muudatusi mõnikord isegi selleks otsest luba küsimata. Viki on hea näide vabama dokumentatsiooni korraldamiseks. [Robbins]

3.1.3 Vigade andmebaas

On väga oluline, et oleks võimalik jälgida lahendamata vigu. Selleks on loodud veahaldussüsteemid. Osad arendajad eelistavad läbi meili kasutatavat tööriista - veateadeanded tulevad arendajale meilile ja arendaja saab omakorda vastata meilikaudu. Teised arendajad eelistavad veebipõhiseid veahaldussüsteeme. [Goldman&Gabriel]

Üks tuntumaid veahaldussüsteeme avatud lähtekoodiga projektide seas on Mozilla projekti jaoks loodud Bugzilla. [Robbins] Täpsemat informatsiooni ja nõuandeid võib leida aadressilt https://developer.mozilla.org/en/Bug_writing_guidelines.

3.1.4 Avatud meililist

Nagu juba mainitud, on avatud lähtekoodiga projekti juures tähtis, et kogu arutelu projekti kohta oleks avalik. Kõik kasutajad ja arendajad peaksid kasutama avalikku meililisti või uudistegruppi oma arutelu jaoks. Arutelud meililistides hõlmavad endas küsimuste ja vastuste sessiooni kasutajate ja arendajate vahel, ettepanekuid muudatuste ja edasiarendamise kohta, teateid uute väljalasete kohta ja ka hääletamist tähtsate otsuste üle. [Robbins]

3.1.5 Projekti veebileht

Iga projekt vajab veebilehte, kust potentsiaalsed kasutajad saaksid leida kogu info ja uudised projekti kohta. Goldmani ja Gabrieli sõnul peaksid veebilehel olema olemas kasutajajuhised, õpetused, meililistide arhiivid ja teised dokumentatsioonid. Ja loomulikult ei saa unustada allalaadimislehte, kus külastajad saaksid kätte viimase versiooni lähtekoodist.

3.2 Tarkvara protsess

Oleme heitnud pilgu infrastruktuurile, mis toetab avatud lähtekoodiga projekti. Nüüd uurime tegevusi, mida tuleb projekti edenedes ette võtta. Goldman ja Gabriel on ära märkinud olulist rolli mängivad tööotsad avatud lähtekoodiga projekti juures:

- **Visionäär/kogukonna kordinaator** soodustamaks arendajate kordineerimist, kogukonna suuremat kaasamist projekti, meililistide võõrustamist ning üldiselt projekti liikuvana hoidmist.
- **Mooduli omanikud**, kes vastutavad koodi arengu, teiste arendajate panuste ja veaparanduste integreerimise eest. Lisaks peavad nad osalema ka projekti meililistis.
- **Infrastruktuuri toetus** arhiivile, meililistidele, vigade andmebaasile ja veebilehele.
- **Veebilehe toimetaja**, kes hoiab veebilehte elus uue sisuga.
- **Dokumenteerijad** süsteemi ülesehituse ja projekteerimis põhjuste kirjapanemiseks.
- **Järgumeister** kontrollimaks järgu protsessi ja parandamaks ettejuhtuvaid probleeme.
- **Väljalaskehaldur** kordineerimaks väljalasketegevust.

3.2.1 Planeerimine

Goldmani ja Gabrieli sõnul peaks planeerimine ja otsustetegemine aset leidma kas projekti avalikus meililistis või avalikul kogukonnakoosolekul. Avalikude aruteludega võib otsuste tegemine võtta tavapärasest rohkem aega, kuid kuna seisukohad on mitmekesisemad, on lõplikud otsused tihti väga hea kvaliteediga. See võib viia lühema üldise arengukäiguni. Planeerimise alla kuuluvad:

- **Kokkuvõtvad sõnumid**, mis võtaks kokku põhiküsimused ja mida on otsustatud teha. Kokkuvõtvasse sõnumisse kuulub ka nimekiri vabatahtlikest, kes otsustasid projektis kaasa aidata.
- **Projekteerimise otsused** ja põhjused otsuste vastuvõtmise taga, mis tuleb kirja panna projekti projekteerimis dokumendi.
- **Ajagraafiku otsused**, mis tuleks kirja panna projekti tegevuskavasse.

3.2.2 Arendajate panuste ühendamine lähtekoodiga

Ainult väikesel rühmal arendajatel on luba lähtekoodi otseselt muuta. Neil, kellel luba puudub, peavad oma töö edastama mooduli omanikule, kes selle üle vaatab ja siis otsuse sobivuses vastu võtab.

Mõned uute funktsioonide ettepanekud on küsitavad või hõlmavad rohkem probleeme kui nad väärt on. Moodulite omanikud peavad tagama koodi kvaliteedi ja seega on sunnitud mõnikord koodile ära ütleva. Kogukonna seisukohalt kõige viisakam oleks neil paluda teha mõningaid muutusi oma töös, et kood saaks hiljem sobivana käiku minna. [Goldman&Gabriel]

3.2.3 Järgud

Oluline on, et kõige viimane stabiilne järk (*build*) oleks kättesaadav allalaadimiseks. Sedasi on võimalik saada kiiremini tagasisidet kasutajate poolt.

Kui järk on vigane, siis sellega tegeleb järgumeister. Järgumeistri ülesanne on probleemi tekkides leida põhjus ja selle parandamine niipea kui võimalik, tavaliselt vigase faili algsemat versiooni taastades. [Goldman&Gabriel]

Mozilla kasutab spetsiaalset tööriista nimega Tinderbox, mis automaatselt kompileerib projekti lähtekoodi ja annab teada kui leidub vigu. Selline tööriist leiab kiirelt vead üles, mis muidu oleksid võinud arendajal märkamatuks jääda. [Robbins] Tinderbox asub aadressil <https://developer.mozilla.org/en/Tinderbox>

3.2.4 Testimine

Igaüks, kes kasutab avatud lähtekoodiga rakendust, võtab osa testimisest. Mida kergemaks on kasutajatele tehtud vigadest teadaandmine, seda tõenäolisemalt nad ka seda teevad. Lisaks veast teadaandmisele võib kasutaja oskuste puhul ka vea ise ära parandada.

Osadel avatud lähtekoodiga projektidel puudub ametlik testimine - kõik vead on teatatud tegelikust kasutamisest. Mõnes mõttes - inimesed on aitavad testida koodi vabatahtlikult juba siis, kui nad laevad alla viimase järgu või viimase stabiilse järgu. [Goldman&Gabriel]

3.2.5 Väljalasked

Avatud lähtekood järgib põhimõtet, et uus lähtekood oleks võimalikult tihti ja varakult avalikustatud. Robbins loetleb oma uuringus kolme erinevat väljalasete tüüpi:

- **Stabiilsed** (*stable*) väljalasked toimuvad umbes samas tempos kui traditsioonilistel toodetel.
- **Õised** (*nightly*) väljalasked toimuvad tihedamini.
- **Avalikud arendamiseks** (*public "development"*) mõeldud väljalasked võivad toimuda juba kiiresti peale projekti alustamist.

Suure väljalaske protsess on sarnane varalise tarkvara omaga. Põhiline erinevus on, et vaba tarkvara puhul on see protsess vabam. Näiteks, kui kood uuele moodulile on stabiilne ja teeb kasulikke asju, siis see võidakse hõlmata väljalaskesse isegi kui dokumentatsioon selle kohta on õhuke.

Väljalaskehaldur kordineerib kogu protsessi. See hõlmab testijate värbamist, testimisprotsessi kordineerimist ja et testijad oleksid kogu toimuvaga kursis. Haldur aitab otsustada, mis läheb väljalaskesse ja mis jäetakse välja. Isegi osad veaparandused võidakse välja jätta, kui veaparandus võib kaasa tuua uusi vigu. Väljalaskmisperioodi ajal ei tehta enam suuri muudatusi koodi.

Kui enamik teadaolevad vead on parandatud ja väljalase on stabiilne, siis lastakse välja beta versioon. Paljud inimesed on nõus proovima beta versiooni, aidates leida järelejäänud vigu.

Lõpuks, kui viimased suured vead on parandatud, on väljalase valmis levitamiseks. Väljalaskele antakse unikaalne väljalaske number, et oleks teada milline on kõige uuem versioon. [Goldman&Gabriel]

3.2.6 Uue mooduli või alamprojekti lisamine

Goldmani ja Gabrieli sõnul on igal avatud lähtekoodiga projektil on oma eksperimentaalne valdkond, kus igaüks saab üles seada uue alamprojekti uue mooduli arendamiseks. Arendajad saavad testida uusi ideid ja teha neid ideid kogukonnale avalikuks kasutamiseks. Tavaliselt selline uus moodul ei ole kaasatud projekti ametlikku väljalaskesse.

Kui kogukond otsustab, et uus moodul annab projektile juurde olulist funktsionaalsust, siis tehakse see ametlikuks projektiosaks. See võib hõlmata suurt tööd, sest eksperimentaalse mooduli standardid on üsna erinevad ametliku omadest.

4. Soovitused

Eespool kirjutatust tulenevalt koostan nimekirja soovitustest, mille järgimine oleks aluseks edukale avatud lähtekoodiga tarkvara projektile.

1. **Hoia dokumentatsioon korras ja uuendatud.** Et kasutajatel oleks võimalikult kerge tundma õppida lähtekoodi ja projekti suundi.
2. **Kasuta CVS versioonihaldussüsteemi.** CVS on enimkasutatud versioonihaldussüsteem avatud lähtekoodiga projektide seas. Seda kasutades saab kasutaja salvestada koodi ja dokumentide ajalugu.
3. **Kasuta Bugzilla veahaldussüsteemi.** Bugzilla on üks tuntumaid veahaldussüsteeme avatud lähtekoodiga projektide seas. Seda kasutades on võimalik jälgida veel seni lahendama vigu ning sisestada uusi vigu.
4. **Loo projektile veebileht.** Et potentsiaalsed kasutajad saaksid leida kogu info ja uudised projekti kohta. Veebilehel peaksid olema olemas kasutajajuhised, õpetused, meililistide arhiivid ja teised dokumentatsioonid.
5. **Kasuta Tinderbox tööriista.** Kiirema tagasiside seisukohalt on oluline, et järk oleks vigadeta ja kättesaadav allalaadimiseks. Tinderbox automaatselt kompileerib projekti lähtekoodi ja annab teada kui leidub vigu.
6. **Tee väljalaskeid võimalikult varakult ja tihti.** Arendajatele on tähtis, et nad töötaksid kõige uuema koodi kallal ja ei kulutaks aega tööle, mida keegi teine on juba ära teinud. Teised inimesed saavad hakkata kasutama viimast väljalaset ja andma teada leitud vigadest.
7. **Loo projektile suhtluskanaleid nagu listid ja foorumid.** Et inimesed saaksid rääkida projektist ja ettevõtte saaks teada palju uut ja huvitavat projekti kohta.
8. **Korralda aeg ajalt ka üritusi.** Näost näkku kohtumistel kuulates probleemide ja huvitavate võimaluste kohta, millega kogukonna liikmed silmitsi seisavad, on võimalus saada väärtuslikke ideid uute projektide jaoks ja luua uusi sidemeid inimeste ja ettevõtete vahel.

Kokkuvõte

Antud teemat oma seminaritöös valides polnud mul erilist aimu, milline näeb välja avatud lähtekoodiga tarkvara arendusmeetod. Arvasin, et see on sarnane kommertstarkvara arendusmeetodile selle erinevusega, et lähtekood on avalik. Nii see siiski päris ei olnud. Avatud lähtekoodiga projekti arendamisest võtavad osa erinevate oskuste ja nägemustega inimesed ümber maailma, seega iga projekti vajadused on erinevad ja mingist kindlast alusreeglistikust ei ole kohustuslik kinni pidada. Avatud lähtekood annab projekti arendamisele vabamad käed.

Käesolevat seminaritööd kirjutades tutvusin lähemalt võimalustega, mida pakub avatud lähtekoodiga projekti arendamine. Neist peamised on suhtlemine kogukonna vahel ja abi, mida kogukond arendamisel osutab. Leidsin, et avatud lähtekoodiga projekt on väga kasutajasõbralik, kus kogukonna liikmed osutavad üksteisele abi vastates küsimustele ning koostades õppematerjale. Abi, mida kogukond projekti arendamisel osutab, olgu see siis koodi arendamine või testimine, võibki pidada avatud lähtekoodiga projekti suurimaks eripäraks ning eeliseks.

Oma seminaritöös käsitletud teemadest tulenevalt koostas ka nimekirja soovitustest, mille järgimine oleks aluseks edukale avatud lähtekoodiga tarkvara projektile. Soovitustes märkisin ära nii vajamineva infrastruktuuri, väljalasete kiiruse ja sageduse olulisuse ning suhtlemise vajalikkuse.

Allikate loend

- [CVS] CVS - Concurrent Versions System; URL:
<http://www.nongnu.org/cvs/>
- [Gibbs] Gibbs, W. Scientific American. "Software's Chronic Crisis"; URL:
<http://www.cis.gsu.edu/~mmoore/CIS3300/handouts/SciAmSept1994.html>
- [Goldman&Gabriel] Goldman, R., Gabriel, R. P. (2005); Innovation Happens Elsewhere: Open Source as Business Strategy; URL:
<http://dreamsongs.com/IHE/IHE.html>
- [Kikkas] Kikkas, K. (2006); Vabad litsentsid – mis see on?; URL:
<http://www.kakupesa.net/kakk/docs/vabadlitsentsid.pdf>
- [OSI] Open Source Licenses; Open Source Initiative; URL:
<http://www.opensource.org/licenses>
- [Robbins] Robbins, J. (2003); Adopting Open Source Software Engineering (OSSE) Practices by Adopting OSSE Tools
- [Stallman] Stallman, Richard Matthew (2002);. Free Software, Free Society. Ed. Joshua Gay. GNU Press; URL: <http://www.gnu.org/doc/TOC-FSFS.html>
- [Wheeler] Wheeler, David A. (2007); Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!; URL:
http://www.dwheeler.com/oss_fs_why.html
- [Woods&Guliani] Woods, D., Guliani, G. (2005); Open Source for the Enterprise
- [The StandishGroup] The Standish Group (1994); 1994 CHAOS Report; URL:
<http://www.liu.se/content/1/c6/04/12/28/The%20CHAOS%20Report.pdf>