

Tallinna Ülikool  
Informaatika Instituut

Indrek Saar

SOOVITUSED INFOTEHNOLOOGILISTE VAHENDITE JA MEETODITE  
VALIKUL TOOTLIKKUSE JUHTIMISEKS ALUSTAVAS  
TARKVARAARENDUSE ETTEVÕTTES

Magistritöö

Juhendaja: Ants Sild

Autor: \_\_\_\_\_ ” \_\_\_\_\_ ” 2010.a.  
Juhendaja: \_\_\_\_\_ ” \_\_\_\_\_ ” 2010.a.  
Instituudi direktor: \_\_\_\_\_ ” \_\_\_\_\_ ” 2010.a.

Tallinn 2010

## **Autorideklaratsioon**

Deklareerin, et käesolev magistritöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

(kuupäev)

.....

(autor)

# Sisukord

Sissejuhatus.....	4
1. Probleemi olemus ja teema aktuaalsus .....	6
1.1 Madal tootlikkus.....	6
1.2 Madal kvaliteet.....	7
1.3 Strateegia puudumine.....	8
1.4 Piiratud ressursid.....	9
2. Juhtimismudeli kavandamine.....	9
2.1 Mudel 5P-d.....	10
2.2 Mudel 7S-i .....	12
2.3 Kokkuvõte ja soovitused .....	14
3. Strateegia koostamise põhimõtted alustavale ettevõttele.....	15
4. Protsessid tarkvaraarendusettevõttes .....	18
4.1 Protsess kui neljas dimensioon .....	18
4.2 Agiilne tarkvaraarendus .....	21
4.2.1 Scrumi põhimõtted.....	22
4.2.2 Ekstreemprogrammeerimine (XP).....	23
4.2.3 Lepingud agiilses tarkvaraarenduses .....	25
4.2.4 Kokkuvõte ja soovitused .....	27
5. Infotehnoloogiliste vahendite valik.....	29
5.1 Tarkvara kui teenus.....	29
5.1.1 Teenuspõhised versioonihalduse ja meeskonnatöö vahendid .....	32
5.1.1.1 Unfuddle.....	33
5.1.1.2 GitHub .....	35
5.1.1.3 SourceGear Vault .....	38
5.1.1.4 Beanstalk.....	38
5.1.1.5 ProjectLocker .....	40
5.1.1.6 Kokkuvõte ja soovitused .....	41
Kokkuvõte.....	42
Summary .....	44
Kasutatud kirjandus.....	45
Kasutatud lühendid ja võõrkeelsed terminid .....	47

## Sissejuhatus

Olen viimased kuus aastat töötanud tarkvaraarenduse valdkonnas ja seda nii ettevõtja kui palgatöölisena. Selle aja jooksul on nii ettevõtluse kui IT valdkonnas toimunud mitmeid uuendusi ja õnneks mitte halvemuse vaid paremuse suunas. Soodsad ja kvaliteetsed äritarkvaralahendused ning IT ettevõtete suhteliselt hea käekäik, on pannud paljud meie seast mõtlema, et kas poleks mitte aeg ise midagi ära teha?

Ettevõtte loomine on tänasel päeval tehtud imelihtsaks, kuid ettevõtte edasine käekäik ning äriiline edukus sõltub suuresti valitud juhtimismetoodikast ning infotehnoloogilistest vahenditest. Alati pole võimalik ja ka mõistlik kulutada suuri summasid, et luua ettevõtte äritegevust toetav töökeskkond. Kui veel kümnekond aastat tagasi oli IT ettevõtte korralikuks toimimise panekuks vaja tarkvarasse investeerida märkimisväärselt suuri summasid, siis tänapäeval on võimalik läbi ajada märksa odavamalt ja ka kvaliteetsemalt.

Oma uurimistöös keskendun ma strateegilistele juhtimisvõtetele, meetoditele ja tarkvaralistele lahendustele ning teenustele, mis aitavad kaasa ning hõlbustavad IT ettevõtluse tootlikku alustamist. Kuna suuri rahalisi väljaminekuid enamik alustavatest ettevõtetest endale lubada ei saa, siis hakkabki peamist rolli mängima IT vahendite valik ja strateegia.

Olgu see siis kas teadmatusest või oskamatusest, kuid väga suur osa täna alustavatest ettevõtetest ei oma vähimatki strateegiat. Ühelt poolt on selles kindlasti süüdi IT tööstuse sh. tarkvaraarendusettevõtete suhteliselt hea käekäik ja teiselt poolt imelihtsaks tehtud ettevõtte alustamise protsess, kuid tuginedes isiklikele kogemustele ning uuringutele, siis võib öelda, et päris ilma strateegiata edu siiski ei saavuta.

Milline on jätkusuutlik juhtimismudel tänases majanduskliimas, pidades eelkõige silmas tarkvaraarendusega alustavaid ettevõtteid? Kuidas planeerida ettevõtte strateegia, kuidas valida põhi ja tugiprotsessid ja mida on tarvis, et tagada edu ka pikemas perspektiivis?

Uurimistöö eesmärgiks on välja selgitada peamised probleemid, mis pärsvad alustavate IT mikroettevõtete edu ja sellest lähtuvalt tulla välja soovitusetega, mis aitaksid neil ettevõtetel tootlikkust parandada ja pikemas perspektiivis ellu jääda. Soovitusi jagan eelkõige tarkvaraarenduse ettevõtete vaatenurgast lähtuvalt, kuid valdav osa nendest soovitusetel on laiendatavad ka teistele alustavatele mikroettevõtetele. Soovitused põhinevad erinevate kirjandusallikate (raamatud, teadusartiklid, internetiviited) kontentanalüüsil ja isiklikel kogemustel. Peamist tähelepanu saavad juhtimismudeli valik, strateegilise plaani koostamine

ja produktiivsust tõstavad IT lahendused. Kindlasti ei ole antud uurimistöö näol tegemist valdkonna kõigi probleemide ja lahenduste analüüsiga, vaid pigem on see autori nägemus mõningatest aktuaalsetest probleemidest ja tootlikkuse juhtimise põhitõdedest, mis läbi isikliku vaatenurga formuleerub kokkuvõtlikuks ja suunavaks materjaliks.

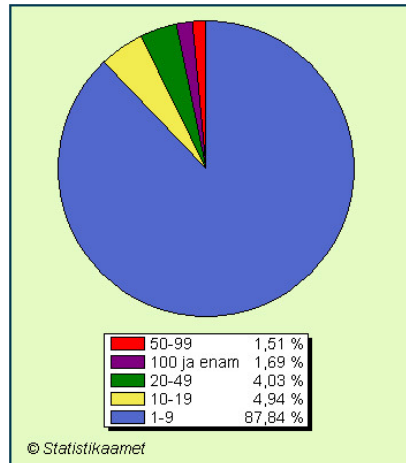
Võttes kokku ja analüüsides erinevaid meetodeid ning põhiprotsesse toetavaid infotehnoloogilisi lahendusi, tulen välja soovitustega, mille järgimine aitab alustavat mikroettevõtet tootlikumalt juhtida. Soovituste jagamisel lähtun isiklikest seisukohtadest ja maailmas levinud parimatest praktikatest.

Esimese peatükis keskendume Eesti IKT sektori hetke olukorrale ja aktuaalsetele probleemidele. Mis on ettevõtluse edu alus ja miks nii paljud nende reeglite vastu eksivad? Selgitades välja peamised probleemid antud maastikul, saab võtta kasutusele sobivad meetmed, et samu vigu tulevikus vältida ning tõsta oma ettevõtte tootlikkust.

Valdkonna kiire uuenemise tõttu pole teemat piisava uuritud. Puudub ühtne strateegiline mudel ning tarkvaraline platvorm, mille rakendamine aitaks oluliselt kiirendada ja tõhustada tarkvaraarendusega tegelevate ettevõtete tööle rakendamist. Tähelepanu keskpunktis on suured rikkad ettevõtted ning neile pakutavad kallid äri lahendused, kuid vähem on uuritud, kuidas tõhustada tootlikkust alustavas ettevõttes, kulutades selleks väga vähe või üldse mitte midagi. Praegusel internetiajastul on see kindlasti võimalik.

## 1. Probleemi olemus ja teema aktuaalsus

Valdav osa Eesti IKT sektori ettevõtetest on mikroettevõtted (kuni 9 töötajat). Statistikaameti andmebaasi kohaselt oli mikroettevõtete osakaal 2009. a. neljandas kvartalis 87,84% (vt joonis 1), mis oli 1459 ettevõtet kõigest 1661-st IKT sektori ettevõttest.



**Joonis 1.** IKT sektori ettevõtted töötajate arvu järgi protsentuaalselt. IV kvartal, 2009.

(Statistikaamet, 2009).

Probleemiks ei ole mitte mikroettevõtete suur osakaal, vaid nende vähene või puudulik toimimine. IKT uuringu 2009. aasta lõpparuandes on kirjas, et 2007. aasta lõpu seisuga oli Eestis 1969 IKT ettevõtet. Neist ligikaudu kolmandik olid pisikesed, ilma töötajateta firmad ja 41% kuni 9 töötajaga mikroettevõtted (IKT sektori lõpparuanne, 2009). Jämedalt võttes võib öelda, et kolmandik Eesti mikroettevõtetest ei anna toodangut, neil puuduvad eesmärgid ning on omanikele pigem kahjuks.

### 1.1 Madal tootlikkus

*„Arvutiajastu on tänapäeval nähtav kõikjal peale tootlikkuse statistika” (Robert Solow, Nobeli majanduspreemia 1987.a. laureaat).*

Info- ja kommunikatsioonitehnoloogia (IKT) on olnud viimastel aastakümnetel üks kiireminiarenevaid teadus-, tehnoloogia- ja ärivaldkondi maailmas. Möödunud kümnend on seejuures olnud eriti kiire kasvu periood. Seda perioodi on iseloomustanud järjest uute IKT-l põhinevate toodete ja teenuste masskasutusse jõudmine ning interneti kujunemine universaalseks side-infrastruktuuriks. Tavatarbijad on kogenud nii arvutustehnika kui ka

sidesüsteemide peadpöörivalt kiiret arengut. IKT tarbekaubastumine, toodete järjest kiirem arendustsükkel ja kasutuselevõtt, nende lühem tööiga ja vähenenud kasumimarginaalid on aga muutnud väiksemate, finantsiliselt nõrgemate ja seetõttu ka lühemaajalise tulevikuvaatega IKT ettevõtete elu järjest keerulisemaks (Eesti infotehnoloogia tulevikuvaated, 2009).

Statistikaameti uurimus näitab, et IKT sektori ettevõtete tööviljakus lisandväärtuse alusel oli 2009. aastal kaks korda kõrgem kui kogu ettevõtluses keskmiselt. Seega ei saa madalat tootlikkust IKT sektori kohta üldistada, kuid kui vaadata täpsemat statistikat, siis on näha, et kõige väiksema panuse annavad just nimelt mikroettevõtted. Võrreldes 2009. aasta viimase kvartali müügitulu ja inimeste koguarvu mikroettevõtetes, on näha, et antud suhtarv on teistega võrreldes kõige tagasihoidlikum (vt tabel 1). Antud tulemuses on just nimelt süüdi suur hulk mikroettevõtteid, kes ei anna toodangut ja viivad statistilised näitajad alla. Samas eksisteerib suur hulk mikroettevõtteid, kes teenivad märkimisväärset kasumit ja kompenseerivad suuresti selle, et ligi kolmandik mikroettevõtetest märgatavat müügitulu ei teeni, seega on lõhe mikroettevõtluses väga suur.

**Tabel 1.** IKT ettevõtete majandusnäitajad jooksevhindades, 2009, 4. kvartal (Statistikaamet, 2009).

	Keskmine tööga hõivatud isikute arv	Müügitulu, tuhat krooni	Inimese kohta
<b>Ettevõtte suurus (inimesi)</b>	15980	6611769	414
<b>1-9</b>	<b>3197</b>	<b>737165</b>	<b>230</b>
10-19	1055	400606	380
20-49	1697	753831	444
50-99	1680	415195	247
100 ja enam	8351	4304972	516

## 1.2 Madal kvaliteet

Kuigi Eesti infotehnoloogiaettevõtete tase on üldises plaanis kõrgelt hinnatud, siis tööde kvaliteedinäitajad on pigem madalad ja seda just nimelt mikroettevõtetes. Üha enam tuleb turule selliseid tegijaid, kes pööravad kvaliteedile liiga vähe tähelepanu. Riigikontrolli hinnangul pärsib töökindlate ja jätkusuutlike infosüsteemide loomist kõige enam puudulik eelanalüüs, alternatiivsete lahenduste kaalumata jätmine ning projektide läbiviimise põhitõdede eiramine (Riigikontroll, 2010).

Olen mõnelt tuttavalt kuulnud ka seda, et nende ettevõtte on loobunud IT lahenduste tellimisest Eesti tootjatelt, kuna saadud lahendused on olnud mitmel korral väga madala kvaliteediga ja vigade parandamine on kujunenud liiga aeganõudvaks ja kulukaks.

Tundub, et alustava tarkvaraettevõtte üheks suureks konkurentsieeliseks oleks üle keskmise kõrge kvaliteediga tooted. See, kuidas integreerida kvaliteedimõõdikud ettevõtte üldisesse juhtimismudelisse ning läbi pideva tagasiside oma põhiprotsesside tulemit kvaliteetsemaks muuta, tuleb uurimise alla edaspidi.

### **1.3 Strateegia puudumine**

Uuringud on näidanud, et strateegilise juhtimise alaste teadmiste puudulikkus on üks väikeettevõtete läbikukkumiste peapõhjusi. Väikeettevõtted, kelle juhte iseloomustab strateegiline mõtlemine, on samas teistega võrreldes edukamad. Muidugi on väikeettevõtte strateegia kujundamisel oma eripära: protsess on vähem formaalne, tulemused ei pruugi olla paberil fikseeritud ega ametlikult heaks kiidetud. Mõnigi kord on kogu strateegia vaid ühe isiku mõtetes, kuid see ei tähenda, et teda ei ole. (Leimann, Skärvad, Teder. 2001). Sama kehtib ka alustavale ettevõttele. Selleks, et tagada ellujäämine ja kasv, peavad asutajad vähemalt oma peades omama nägemust, mis nende arvates tagab neile piisava edu.

Strateegilised valikud alustavas väikeettevõttes erinevad suurtest ja juba toimivatest ettevõtetest mitmele viisil, seega meeletehnik suurte ettevõtete jäljendamine ei pruugi anda soovitud tulemust. Alustaval ettevõttel on eksimisruum vägagi piiratud ning kehvalt valitud strateegilised otsused saavad sageli saatuslikuks.

Edaspidi tuleb juttu alustava tarkvaraarendusettevõtte jaoks lihtsustatud strateegilise planeerimise mudelist, mis hoiab ettevõtte asutajate mõtted fookuses ning suunab tegevust õiges suunas. Eesmärgiks pole mitte strateegilise planeerimise elementide ülevõtmine, vaid nende selekteerimine ja kohandamine vastavalt alustava mikroettevõtte vajadustele.

## 1.4 Piiratud ressursid

*„IT on kulukas valdkond. Ilma selge sihita on tõsine oht teha investeeringuid juhuslikult ja mõttetult.“  
(Hansson, Leego & Partner).*

Sageli mõeldakse, et kvaliteetse töökeskkonna sisse seadmine on väga kulukas ettevõtmine ning alustava ettevõttega ei saa me seda endale lubada. Kuigi tänapäeval on saadaval väga palju väga korralikku tasuta tarkvara ei osata seda tihtipeale ära kasutada või kasutatakse saada olevaid ressursse väga piiratud. Lähtuvalt tarkvaraarendusettevõtete vajadustest, uurin töö teises osas erinevaid tarkvaralisi platvorme, mis ühe alustava tarkvaraettevõtja kasutuses kindlasti peaks olema.

Tarvaaraarenduse ettevõttes on mõistlik esmajoones plaan ja protsessid paika panna ning vastavalt ajalistele ja rahalistele võimalustele rajada sellest lähtuv infrastruktuur. Selleks, et algsed investeeringud end ära tasuksid, tuleb hoolikalt läbi mõelda, mida me vajame ning kui palju meil on võimalik nende toodeta ja teenuste eest maksta.

Tarkvaraarendusettevõtte puhul on projekti kood ettevõtte pühaks reliikviaks, mille turvalisus peab olema tagatud. Siit tulenevalt on üheks uurimisobjektiks koodi turvaline haldamine (versioonihaldustarkvarad), aga samuti projektijuhtimise ja meekonnatöö vahendid, mis aitavad ettevõtet tootlikult juhtida.

Valdav osa olulisemaid IKT alaseid uuendusi sünnib väljaspool Eestit, arenenud tööstusriikides. Eestile on seetõttu ennekõike oluline osata mujal sündivat tehnoloogiat ära kasutada. (Eesti infotehnoloogia tulevikuvaated, 2009).

## 2. Juhtimismudeli kavandamine

Selleks, et ettevõtte tegevusi mingilgi määral planeerida ja teadlikumalt suunata, on tarvis järgida teatud tegevusmustrit. Uurin siinkohal juhtimismudeli olemust ja toon välja lihtsustatud mudelid, mida alustavas ettevõttes võimalik kasutusele võtta.

Lähtuvalt ettevõtte strateegilistest eesmärkidest valitakse põhi ja tugiprotsessid, mille täitmise järel saadakse planeeritud tulemused. Eesmärk on uurida paari alustava ettevõtte jaoks sobilikku juhtimismudelit ning saadud lisateadmiste baasil pakkuda välja lihtne skeem, mis

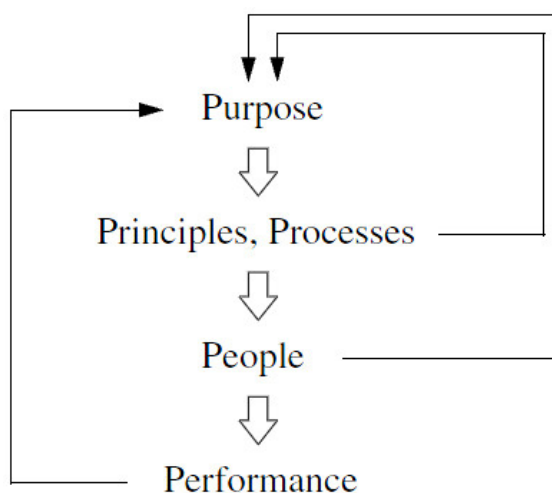
igal alustaval mikroettevõtjal meeles peaks mõlkuma. Oluline on eraldi uurida ka kvaliteedi kontrollmehhanisme ja need loodavasse mudelisse sisse integreerida.

Kuigi erinevaid juhtimismudeleid/raamistikke on palju, siis toon siinkohal välja need, mis tänu oma lihtsusele ja praktilisele poolele võiksid sobida alustavale tarkvaraarenduse ettevõttele. Meeles tuleb pidada, et juhtimismudel ei ole vahend, mis garanteerib ettevõtte edukuse, vaid tegu on vahendiga, mis aitab hoida mõtted õigel kursil ning keskenduda olulistele tegevustele.

## 2.1 Mudel 5P-d

Nimetus 5P-d tuleneb inglise keelsest nimest *5P-s Model* (Pryor, White, Toombs. 1998). Antud mudeli sai uurimise alla võetud, kuna mitmed praktikad näitavad, et antud mudelis järgitavad meetodid on eriti hästi rakendatavad just alustavas ja väikeses ettevõttes. 5P mudelis on lisaks strateegiale pööratud tähelepanu ka kvaliteedi juhtimisele, mis on tarkvaraarenduse ettevõtte puhul äärmiselt oluline.

Nimetus 5P-d tuleneb sellest, et antud mudel koosneb viiest suuremast elemendist – eesmärgid, põhimõtted, protsessid, inimesed ja saavutused, mis inglise keeles algavad kõik P tähega (*Purposes, Principles, Processes, People, Performance*). Mudeli visualiseering on toodud joonisel 2.



**Joonis 2.** *The 5P's model* (Pryor, White, Toombs. 1998).

Antud joonisel (vt joonis 2) välja toodud võtmetegurid on detailsemalt tõlgendatavad järgmiselt:

- *Strateegia (Purpose)*
  - SWOT analüüs
  - Missioon
  - Visioon
  - Eesmärgid
- *Põhimõtted (Principles)*
  - Ettevõtte filosoofiad ja eetika, töökorraldus
- *Protsessid (Processes)*
  - Reasta kõik protsessid
  - Dokumenteerige kõik protsessid
  - Määra protsessile vastutav omanik
- *Inimesed (People)*
  - Inimressursi ja koolituste planeerimine
- *Tulemused (Performance)*
  - Loo mõõdikud saavutatud tulemuste hindamiseks ning vajadusel tee korrekture

Nooled antud joonisel kujutavad strateegia (*Purpose*) ja struktuuri (põhimõtted kui sisene ja protsessid kui väline struktuur) vahelist seost ning struktuuri mõju inimeste käitumisele (*People*) ja sellest saavutatavaid tulemusi (*Performance*). Antud mudeli kohaselt on strateegia juhtrollis, mis suunab kogu ülejäänud tegevust, otseselt aga struktuuri (põhimõtted ja protsessid). Struktuur juhib käitumist (inimesi) ja käitumine mõjutab otseselt tulemust. Nool tulemustelt eesmärgile kajastab tagasiside mehhanismi, mis juhib ettevõtte tegevust eesmärkide suunas. Järjepidev tagasiside, et võrrelda tulemusi ja strateegias seatud eesmärke on hädavajalik, et tagada õige suund ja eesmärkide järjepidev saavutamine. Antud

juhtimisraamistiku eesmärk on aidata kaasa ettevõtte kvaliteetsele tulemuslikkusele, heale staatusele ja pikale elueale.

## 2.2 Mudel 7S-i

Tegu on 1980-ndatel ettevõtte kirjeldamiseks loodud diagnostilise tööriistaga, mille autoriteks on Tom Peters, Robert Waterman ning nende kolleegid ettevõttest *McKinsey & Company*. Seetõttu on mudel tuntud ka kui McKinsey mudel. Antud mudeli 7 (s-ga algavat) elementi on strateegia (*strategy*), struktuur (*structure*), süsteemid (*systems*), stiil (*style*), oskused (*skills*), personal (*staff*), jagatud väärtused (*shared values*). Kui S-id on kokku sobivad ja üksteist toetavad, siis liigub organisatsioon eesmärgipäraselt. Kui S-id hakkavad omavahel konflikte tekitama, siis on see märk sellest, et organisatsiooni toimimises on midagi valesti. Samas, organisatsiooni, kus need seitse S-i on kooskõlas, on tunduvalt raskem muuta kui organisatsiooni, kus need pole kooskõlas. Mikroettevõtte puhul antud raamistik väga hästi ei sobi, sest väike ettevõtte nõuab rohkem agiilsust ning protsesside vahelised seosed ei saa nii jäigalt defineeritud olla. Küll aga leiab siit sarnasusi eelnevalt kirjeldatud 5P mudeliga ning sobib niiöelda teiseks strateegiliseks platvormiks, kui ettevõtte on juba äritegevuse käima saanud ning soovib laieneda ja tootmist efektiivsemaks muuta. Seega, antud mudel on alustava mikroettevõtja esmavajaduste tarvis liialt detailne ning nõuab pikemat planeerimist ja keskendumist, kuid ettevõtte kasvades ning esmaste protsesside juurdumise järel, saab seda edukalt samm sammu haaval sisse tooma hakata.

7S mudeli järgimine peaks tagama ettevõtte kõigi osade toimimise. Antud mudelit saab kasutada abivahendina mitmetel eesmärkidel, nagu näiteks:

- Ettevõtte tootlikkuse parandamiseks
- Tulevikumõjude hindamiseks
- Ettevõtte erinevate osapoolte vaheliste seoste hindamiseks ja analüüsiks
- Strateegia planeerimiseks ja hindamiseks

Elementid McKinsey mudelis jagunevad „pehmeteks“ ja „kõvadeks“ elementideks (vt tabel 2).

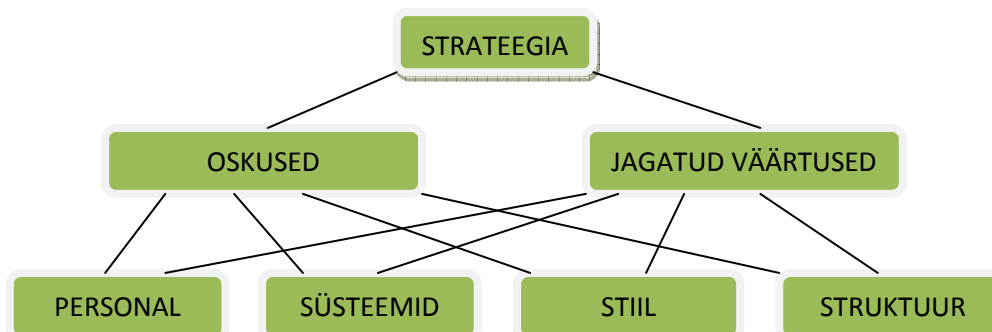
**Tabel 2.** 7S mudeli elemendid.

Kõvad elemendid	Pehmed elemendid
Strateegia	Jagatud väärtused
Struktuur	Oskused
Süsteemid	Stiil
	Personal

„Kõvad“ elemendid on kergemini tajutavad ning juhtkond saab neid otseselt mõjutada. Need on strateegilised eesmärgid, organisatsiooni kaardistus ning formaalsed protsessid ja IT süsteemid.

„Pehmed“ elemendid seevastu on raskemini kirjeldatavad ja juhitavad ning on pigem mõjutatud kultuurist kui juhtkonnast. Samas on ka „pehmed“ elemendid ettevõtte kui terviku toimimiseks äärmiselt vajalikud.

Kõik elemendid 7S mudelil on omavahel tihedalt seotud, mistõttu muutus ettevõtte ühes osas mõjutab ka ettevõtte teisi osasid.



**Joonis 3.** 7S mudelis on elemendid omavahel põimunud (Leimann, Skärvad, Teder. 2001).

- Strateegia – plaan, mil moel tagada ja kasvatada ettevõtte konkurentsieelist.
- Struktuur – Mil moel ettevõtte on struktureeritud ja kes raporteerib kellele.
- Süsteemid – Igapäevased tegevused ja protseduurid, mida vaja, et töö tehtud saaks.
- Jagatud väärtused – Üldine tööeetika ja ettevõtte põhiväärtused, mida personal järgib.

- Stiil – Juhtorganite toimimise stiil.
- Personal – Töötajad, kes oma võimekusega tagavad tulemused.
- Oskused – Töötajate tegelikud kompetentsid ja oskused.

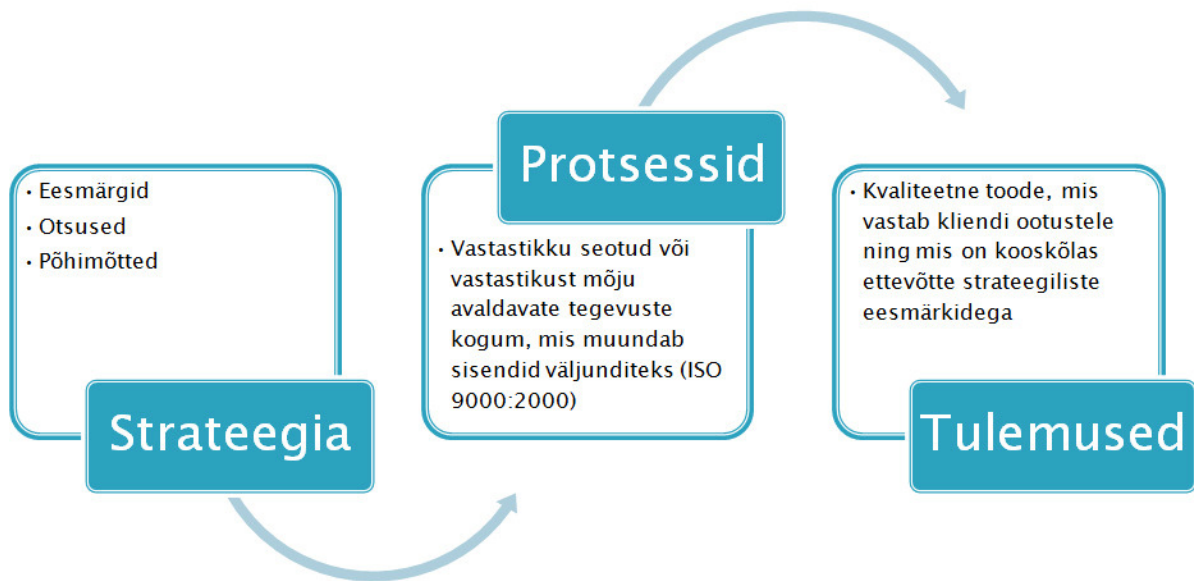
(The McKinsey 7S Framework)

## 2.3 Kokkuvõte ja soovitused

Juhtimismudel on kui eeskiri, mida järgides on lihtsam õigel rajal püsida. See, kas lähtuda ühest konkreetsest mudelist ja katsuda kõiki seal kirjeldatud protsesse täpselt järgida, on juba iseasi. Pigem tundub, et oluline on teada, et sellised abivahendid eksisteerivad ja vajadusel neile tugineda. Seega, analüüsides erinevaid juhtimismudeleid ja tuues välja kaks sellist, mis sobivad kõige paremini alustavale ja väikestele ettevõtetele, võib välja tuua järgmised soovitused:

- **Strateegia** – kõik toetub strateegiale. Loo lihtne ja ülevaatlik strateegiline plaan. Mis on meie strateegia, mis aitab meil eesmärke saavutada? Kuidas eristume teistest? Strateegia koostamisest tuleb juttu ka edaspidi.
- **Strateegiast lähtuvalt** valitakse äritegevuseks vajalikud protsessid ning põhitegevust toetavad tugiprotsessid. Kuidas me tööd teostame? Millised arendusmetoodikad ja parimad praktikad juurutame? Milline on meie äritegevust toetav tarkvaraline platvorm?
- **Protsesside ellu viimiseks** on tarvis inimesi. Millised positsioonid peavad olema töötajatega täidetud? Kui palju inimesi me hetkel vajame?
- **Inimesed annavad tulemusi.** Kas tulemus on ootuspärane? Mida me saame järgmine kord paremini teha? Vastates pidevalt nendele küsimustele, parandame nii toote kvaliteeti, kui tootmisprotsessi tervikult.

Joonisel 4 on toodud ülevaatlik juhtimismudel, kus peal suund ja olulisemad etapid jõudmaks ideest jätkusuutliku tootmiseni.



**Joonis 4.** Ülevaatlisk juhtimismudel

Enamike juhtimismudelite vundamendiks on ettevõtte strateegiline plaan. Järgmiseks tuleb uurimise alla ettevõtte strateegia, mida see endast kujutab ja kuidas seda valutult muude tegemiste käigud planeerida.

### 3. Strateegia koostamise põhimõtted alustavale ettevõttele

Ettevõtte asutamine on tänaseks tehtud imelihtsaks, kuid ainult ettevõtte asutamisega efektiivset äritegevust ja püsijäämist ei taga. Selleks, et esimese tegevusaasta järel mitte pankrotti minna, on tarvis oskuslikult valitud ärimudelit ja strateegiat. Võtmesõnadeks tänases ärikliimas on kvaliteet ja paindlikkus – ole oma ala professionaal ning arvesta klientide muutuvate nõudmistega. Selleks, et ettevõtet oskuslikult juhtida, on loodud mitmeid juhtimismudeleid, mis peaksid oluliselt hõlbustama valikute tegemist ja suunama ettevõtja tootlikkust positiivses suunas. Kuna juhtimismudeli vundamendiks on ettevõtte strateegia, siis toon siinkohal välja strateegilise planeerimise olulisemad komponendid, millele tuleks tähelepanu pöörata juba ettevõtte väga varases algusfaasis.

Henry Mintzberg oma raamatus „*The Rise and Fall of Strategic Planning*“ räägib sellest, et strateegiat võib mõista mitut moodi, kuid enim levinud on järgmised arusaamad:

- Strateegia on plaan, kuidas saada punktist A punkti B ehk praegusest olukorrast soovitud olukorda.

- Strateegia on ajapõhine tegevusmuster.
- Strateegia on positsioon. See tähendab, et mil moel ettevõtte ennast turul positsioneerib – mida müüb ja kellele see on suunatud.
- Strateegia on perspektiiv, see on visioon ja suund.

(Mintzberg, 1994).

Väikeettevõtte jaoks on strateegia sarnane väikese lahingurühma tegevusplaaniga – kes on meie vaenlane ning kuidas talle ära teha. Lahingurühmaga võrreldes on ettevõttes strateegiline juhtimine siiski mõnevõrra keerulisem, sest vaenlase asemel on klient. Ehk siis tuleb mõelda, mida klient tahab ning kuidas me saame talle seda kõige paremini pakkuda, et temas tekiks ja säiliks soov meile võimalikult palju teenuse või kauba eest maksta. (Valmra, 2007).

Järgnevalt toon strateegilise planeerimise elemendid (vt tabel 3), mis on sobilikud alustava ettevõtte strateegia kirjeldamiseks.

*Tabel 3. Strateegilise planeerimise komponendid (Pryor, Toombs, Anderson, White).*

Strateegia komponent	Selgitus
<b>SWOT analüüs</b>	Ettevõtte sisemiste tugevuste ja nõrkuste ning väliste ohtude ja võimaluste analüüs.
<b>Missioon</b>	Miks ettevõtte eksisteerib.
<b>Visioon</b>	Kus soovitakse olla tulevikus.
<b>Põhiväärtused</b>	Põhimõtted, millest inimesed hoolivad ja mis mõjutab nende käitumist.
<b>Eesmärgid</b>	Seatud eesmärgid, mille nimel tööd tehakse.
<b>Mõõdikud ja kontroll</b>	Monitooringu ja tagasiside element, mis vastab küsimustele „Kuidas ettevõttel läheb?“ ja „Kas on tarvis midagi muuta?“
<b>Kriitilised edufaktorid</b>	Mida on tarvis edu saavutamiseks teha.
<b>Konkurentsieelis</b>	Mis teeb meid konkurentidest paremaks.

Selleks, et ettevõtte eesmärged ja eripära paremini hinnata ning strateegia loomine võimalikult sirgjooneliseks teha, toon siinkohal välja hulga küsimusi, millele vastamine aitab alustaval tarkvaraarenduse ettevõttel end positsioneerida ja oma tugevusi ja nõrkusi hinnata. Bob Walsh, oma raamatus „*The Web Startup Success Guide*“ (Walsh, 2009) soovib alustaval

tarkvaraarenduse ettevõtte strateegia koostamiseks endalt küsida järgmisi suunavaid küsimusi:

- Miks me selle ettevõtte teeme?
- Mida täpselt me müüme?
- Kes meie toodangut ostab ja miks?
- Kuidas me oma toodangu ja arvatava ostja kokku viime ehk kuidas kliendid meid leiavad?
- Kuidas me edu mõõdame?
- Millised tegevused on meie põhitegevused, millele me keskendume ise ning mis on toetavad tegevused, mida on mõistlik sisse osta?
- Milline on meie põhiklientuuri iseloom ja ostuharjumused, kellele me oma toodet või teenust müüme?
- Mis teeb meid võrreldes teiste tarkvaraarenduse ettevõtetega eriliseks?
- Millised on minu ja mu partnerite isiklikud tugevused ja nõrkused?
- Kes on meie peamised konkurendid?

Alustava tootmisettevõtte strateegia peaks eelkõige keskenduma hinnale, kvaliteedile ja paindlikkusele. Tabelis 4 on ära toodud tootmisettevõtte peamised strateegilised eesmärgid, mis aitavad ärilisele edukusele kaasa.

*Tabel 4. Tootmisettevõtte strateegilised eesmärgid (Spring, Boaden).*

Eesmärk	Selgitus
<b>Hind</b>	Tekitada konkurentsieelis läbi madala hinna.
<b>Kvaliteet</b>	Toota kõrge kvaliteediga ning vastavalt standarditele.
<b>Tähtajad</b>	Tähtaegadest kinnipidamine aitab võita usaldust ning näitab professionaalsust.
<b>Kiirus</b>	Kliendi soov tuleb täita võimalikult kiiresti.
<b>Paindlikkus</b>	Võime muutuvate vajadustega ja ebaselgusega toime tulla.
<b>Muu</b>	Hilisem klienditugi, reklaam, lai tootevalik.

Kokkuvõtteks võib öelda, et üldjuhul jätvad alustavad ettevõtted oma strateegilised plaanid kirja panemata, kuna kogu see protsess on liiga keeruline. Erinevad juhtimise käsiraamatud kulutavad sadu lehekülgi strateegi olemuse ja vajalikkuse kirjeldamisele, kuid alustava ettevõtja jaoks on tähtis, et esmane strateegia koostamine oleks lihtne ja selle planeerimine võimalikult mugav. Vastates eelmainitud küsimustele ning pannes ettevõtte tegevuse planeerimisel ja täideviimisel rõhu kvaliteedile, kiirusele ja hinnale, saab esmase strateegia sõnastada kümnekonna lausega ning iga kord, kui siht silme ees kokku hakkab jooksuma, saab need küsimused-vastused välja võtta ja oma eesmärgid ja põhimõtted taaskord üle vaadata.

#### **4. Protsessid tarkvaraarendusettevõttes**

*„Protsessid on ettevõtte juhtimismudelites kesksel kohal. Protsesside kaudu toimub uue väärtuse loomine. Protsesside määramine peaks algama organisatsiooni missioonis, visioonist ja strateegilistest eesmärkidest.“ (Tiia Tammaru).*

Ettevõtte põhiprotsessideks on põhitegevusega seotud ning kliendile väärtust loovad tegevused. Väga üldiselt öeldes on tarkvaraarenduse ettevõttes kliendile lisaväärtust loovaks põhiprotsessiks tarkvara tootmine, kuid selleks, et tarkvara toota, on vaja paika panna ridamisi toetavaid tugiprotsesse.

Antud peatükis tulevad uurimise alla erinevad protsessid, mis aitavad tarkvara tootmist võimalikult efektiivselt juhtida ja ellu viia. Protsesside valikul ja kirjeldamisel lähtun agiilsete arendusmetoodikate põhitõdedest ning seda ekstreemprogrammeerimise (XP) ja *scrumi* näitel. Agiilsed arendusmetoodikad on valitud seetõttu, kuna usun, et antud lähenemine annab alustavale ettevõttele vajaliku paindlikkuse ja õiged töövõtted, et tänases konkurentsitihedas keskkonnas edukalt toimida.

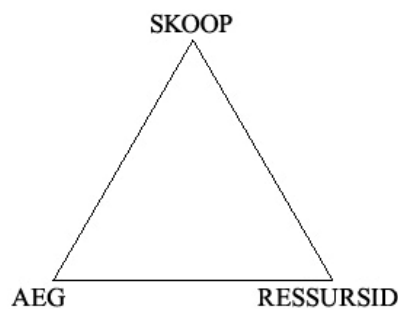
##### **4.1 Protsess kui neljas dimensioon**

Alistair Cockburn, kes on üks tarkvaraarenduse agiilse liikumise eestvedajatest ja kes 2001. aastal aitas kirjutada agiilse tarkvaraarenduse manifesti (*Manifesto for Agile Software Development*), kirjutab oma isiklikul kodulehel <http://alistair.cockburn.us> protsessist, kui tarkvaraarenduse neljandast dimensioonist.

Cockburn kirjutab, et kuigi agiilse tarkvaraarenduse manifest väidab, et inimesed ja inimeste vahelised suhted on protsessidest ja töövahenditest märksa tähtsamad, siis ikkagi kulutavad agiilsete meetodikate järgijad palju aega rääkimaks erinevatest protsessidest. Näitena võib tuua ekstreemprogrammeerimise (*XP*), mis koosneb peaaegu täielikult protsessidest. Kas järgid testimise, planeerimise, paarisprogrammeerimise jne. protsesse, millele kogu *XP* olemus on ehitatud? *XP*-st ja seal kirjeldatud protsessidest tuleb juttu edaspidi.

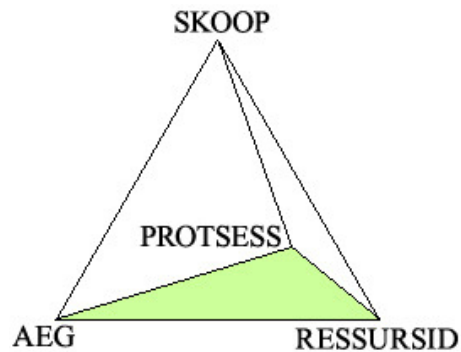
Protsess on kahtlemata oluline. Kuigi hea protsess ei taga toote valmimist ja hea meeskond suudab toote valmis teha ka ilma, et omaks vähimatki teadmist protsessidest, siis kehv protsess saab kindlasti olema suureks takistuseks.

Projektil on kolm dimensiooni – skoop, aeg ja ressursid. Jättes ühe dimensiooni vabaks, saab kahte ülejäänud dimensiooni vajadusel muuta. Kui lukustada kõik kolm dimensiooni, muutub projekti ellu viimine uutel tingimustel võimatuks.



**Joonis 4.** Projektijuhtimise „raudne kolmnurk“. (Cockburn, *Process: the 4th dimension*, 2003).

Selleks, et erand olukordades toimida ka siis, kui kõik kolm eelmainitud mõõdet on lukus ja rohkem aega, ressursse ega kitsamat skoopi pole võimalik saada, tuleb sisse tuua neljas mõõde – protsess. Cockburn väidab, et enamike organisatsioonide protsessid on niivõrd ebaefektiivsed, et nende ülevaatamise ja optimeerimise tulemusena võib olulisel määral nii aega kui ka ressursse kokku hoida. Üheks aega ja ressursse säästvaks tegevuseks tarkvaraarenduses on agiilsete arendusmeetodikate kasutuselevõtt.



**Joonis 5.** Neljas dimensioon – protsess. (Cockburn, Process: the 4th dimension, 2003).

Mida tuleks teha, et murda välja „raudsest kolmnurgast“ ja võita aega ja ressursse ning seeläbi oluliselt tõsta ettevõtte tootlikkust? Cockburn väidab järgmist: „Kolm aegunud ja ebaefektiivset arusaama, mis tuleks välja vahetada, on järgmised:“

- Enne tarkvara disaini peavad nõuded olema selgelt kirjeldatud ning programmeerimist saab alustada alles siis, kui tarkvara disain on paigas.
- Ülesannete detailne üleskirjutus on parem kui eskiiside loomine nende põhjal diskuteerimine.
- Inimesed töötavad privaatsetes tingimustes paremini.

Cockburn soovib need mõttelaadid asendada järgmiselt:

- Alusta tarkvara disainiga niipea, kui oled kogunud alustamiseks vajaliku hulga informatsiooni (vajalik hulk on projektiti erinev) ja alusta programmeerimist kohe varases staadiumis, et saada võimalikult kiirelt vajalikku tagasisidet.
- Dokumenteerige ainult üldised põhimõtted ja pidepunktid ning asenda dokumentatsiooni puudulikud osad hea vestlusega.
- Too inimesed ühte tööruumi kokku.

Selleks, et antud soovitusi järgida, on loodud mitmeid agiilseid arendusmetoodikaid, mis läbi tiheda suhtluse, pideva testimise ja lühikeste iteratsioonide, aitavad muuta tarkvaraarenduse palju paindlikumaks. Järgmiseks teen lühikese ülevaate veel mõningatest agiilse tarkvaraarenduse põhitõdedest ning konkreetsete näidetena tulevad vaatluse alla *XP* ja *scrum*.

## 4.2 Agiilne tarkvaraarendus

See millise tarkvaraarendusmetoodika kasuks üks või teine alustav ettevõtte otsustab, on iga ettevõtja enda asi. Kõige suurem viga on see, kui antud otsus üldse tegemata jäätakse ning opereeritakse ilma kindla plaani ja tegevusmustrita. Kuna tarkvaraarenduse metoodika valik võib suuresti määrata tarkvaraprojekti edu või läbikukkumise, siis soovitav oleks see osa enda jaoks korralikult selgeks teha ja mingi valik siiski teha. Antud uurimistöö autor soovib alustavatel tarkvaraarenduse ettevõtetel kasutada agiilseid arendusmetoodikaid.

Kaasaegses kiirelt muutuv ärikeskkonnas ei ole tarkvaraarendus traditsiooniliste metoodikate abil sageli võimalik ning seepärast tuleb kasutada uusi metoodikaid, mis on suutelised muutustega kiirelt kohanema. Agiilse tarkvaraarenduse selgitamisel lähtun *Agile Manifesto* põhilistest seiskohtadest.

2001. aasta veebruaris Utah's toimunud konverentsil loodi *Agile Alliance*, et ühendada jõupingutused agiilmetoodikate arendamisel. Konverentsil osalejad koostasid ka ühiseid väärtusi ja metoodikaid määratleva *Manifesto for Agile Software Development*. Me tahame luua uusi paremaid tarkvara arendusmetoodikaid. (allikas: <http://scrum.ee/manifesto>)

Me hindame:

- enam inimesi ja nendevahelisi suhteid kui protsesse ja arendusvahendeid
- enam töötavat tarkvara kui täielikku dokumentatsiooni
- enam kliendi osalust arenduses kui lepingute koostamist
- enam muudatustele reageerimist kui plaani järgimist.

Agiilmetoodikate põhiprintsiibid:

- väärtus kliendile - peamine on tulemus
- individuaalsed võimed - toetu isiku oskustele
- koostöö - innovatsioon läbi interaktsiooni meeskonnas
- adaptatsioon - tagasiside ja muudatuste haldamine
- minimalism - peamine on loodava tarkvara lihtsus.

Agiilsed meetodid julgustavad programmeerijaid end raskekaalulistest protsessiahelaist vabastama ja suunavad nad paindlikkusele, kus arendusetapid on jagatud lühikestesse tsüklitesse ning klient arendusse tihedalt kaasatud. Ennustamise asemele tegeletakse kiire kohanemisega vastavalt hetke olukorrale (Boehm, Turner. 2004).

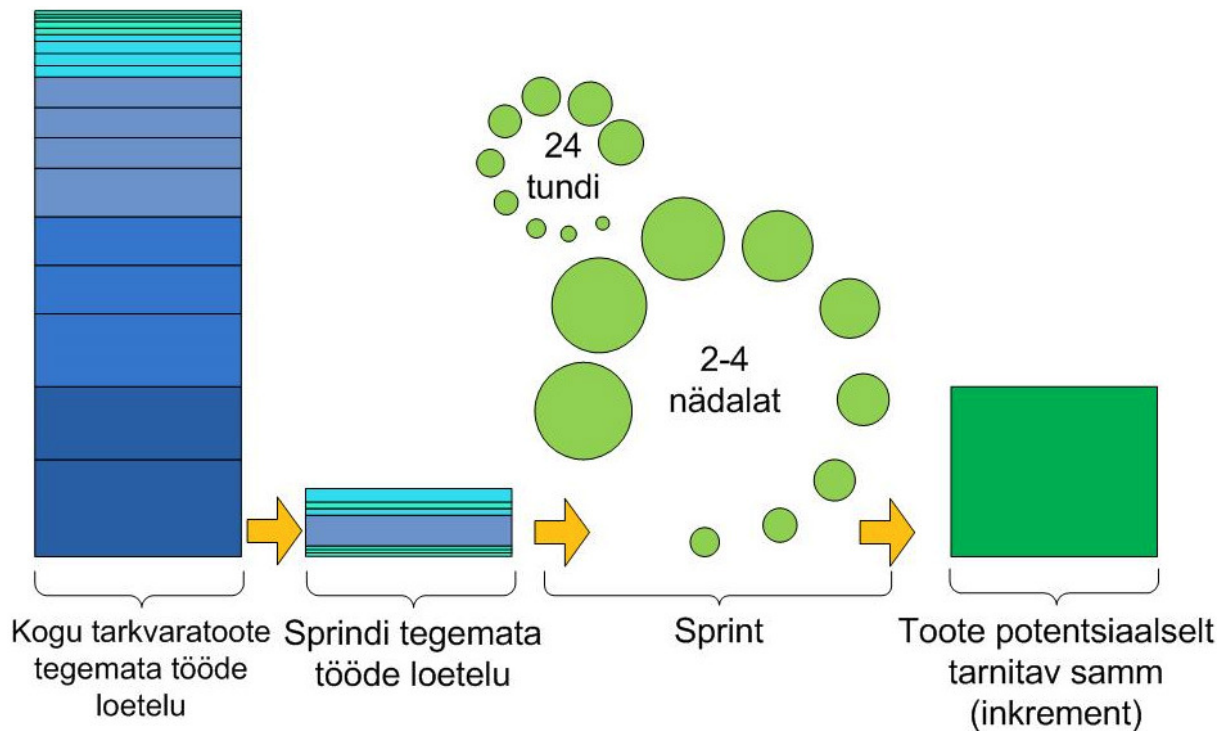
Kokkuvõtlikult võib öelda, et tänu kliendi muutlikule meelele ei ole mõttekas kogu arendusprojekti detailselt läbi analüüsida ja dokumenteerida. Selle asemel on mõistlik jagada arendus väikesteks etappideks ja anda kliendile pidevat tagasisidest saavutatud tulemustest. Sel moel on muutuste sisseviimine kergem ja klient rahulikum.

### **4.2.1 Scrumi põhimõtted**

*Scrumist* on saanud üks kõige edukam tarkvara agiilse arendamise meetoodika nii organisatsioonide, arendajate kui juhtide jaoks. *Scrumi* kasvavat populaarsust võib seletada tema suutlikkusega tõsta ja võimendada investeringult teenitavat tulusust ning võimega ühendada juhtkonda ja arendajaid firma ärieesmärkide saavutamise nimel. *Scrummi* on lihtne mõista ja rakendada, koostamisprogrammid on olemas ja töötavad. *Scrum* pakub paindlikku raamistikku, mis aitab suurtel meeskondadel keskenduda sihile, et ühiselt, kogu meeskonnaga, saavutada iga sprindi ülesanded. (Scrum, 2009).

Scrum on paindlik tarkvara arendamise raamistik. Töö põhineb tsüklilisusel, mille etappe nimetakse sprintideks. Sprindi kestus on tavaliselt kaks kuni neli nädalat. Igaks sprindiks võtavad meeskonnad töösse tähtsuse põhjal järjestatud ülesanded, lähtudes kliendi vajadustest. Ülesandeid nimetatakse *user story'deks* (kasutuslugudeks), nii et funktsioonid, mida arendatakse eelkõige, on kliendile kõige suurema väärtusega. Iga sprindi lõpus tarnitakse potentsiaalselt kasutatav tarkvaratoode kliendile. (Scrum, 2009).

Joonisel 5 on toodud *scrum* arendusmeetoodika raamistik.



*Joonis 5. Scrum-i põhine tarkvaraarendus. (Scrum, 2009).*

#### 4.2.2 Ekstreemprogrammeerimine (XP)

Ekstreemprogrammeerimine (*XP*) on kõige levinum agiilne meetodika (Boehm, Turner, 2004). Ekstreemprogrammeerimise teerajajaks on Kent Beck, kes kirjeldab seda kui kergekaalulist meetodikat väikestele ja keskmise suurusega meeskondadele, kes arendavad ebaselgelt määratletud või sageli muutuvate nõuetega tarkvara (Beck, 2000).

Ekstreemprogrammeerimise eesmärgiks on toota kvaliteetset ja nõuetekohast tarkvara ning vähendada arendusprotsessis ette tulevate muudatuste maksumust. Kui traditsioonilises tarkvaraarenduses fikseeritakse nõuded enne projekti realiseerima asumist, mistõttu on muudatuste sisse viimine üsna keeruline, siis ekstreemprogrammeerimise näol on tegemist paindliku meetodikaga, mis kohaneb vastavalt tekkinud vajadustele.

Tänu oma kliendikesksusele on ekstreemprogrammeerimine osutunud väga edukaks. Kui klassikalises tarkvaraarenduses on kõik protsessid täpselt analüüsitud ja kirjeldatud ning arenduse käigus enam muudatusi ei teha, siis ekstreemprogrammeerimises lähtutakse kliendi hetkevajadustest. Praktika näitab, et pikaajaline etteplaneerimine tarkvaraarenduses pole enamasti kuigi mõttekas, sest kliendi vajadused on ajas muutuvad ning nende muudatuste sisse viimine väga keeruline. Ekstreemprogrammeerimine seevastu suudab muudatusi ellu

viia ka arendustsükli hilistes faasides. Ekstreemprogrammeerimine lähtub meeskonnatöö põhimõttest, kus kõik osapooled on võrdselt tähtsad. Arendusse on kaasatud nii arendajad, projektijuhid ja kindlasti ka klient ise, mistõttu arendus liigub õiges suunas ja toodang vastab kliendi nõudmistele. Viis olulist väärtust, millest ekstreemprogrammeerimises lähtutakse, on järgmised:

- Suhtlus – Dokumentatsiooni asendab tihe vestlus. Kõik teavad, kes mida teeb.
- Lihtsus – Arendusprotsess on võimalikult läbipaistev.
- Tagasiside – Klient on arengute ja probleemidega kursis.
- Tunnustus – Meeskonna töö. Kõik osapooled tunnustavad ja julgustavad teineteist.
- Julgus – Soov midagi ära teha ning julgus vigu tunnistada.

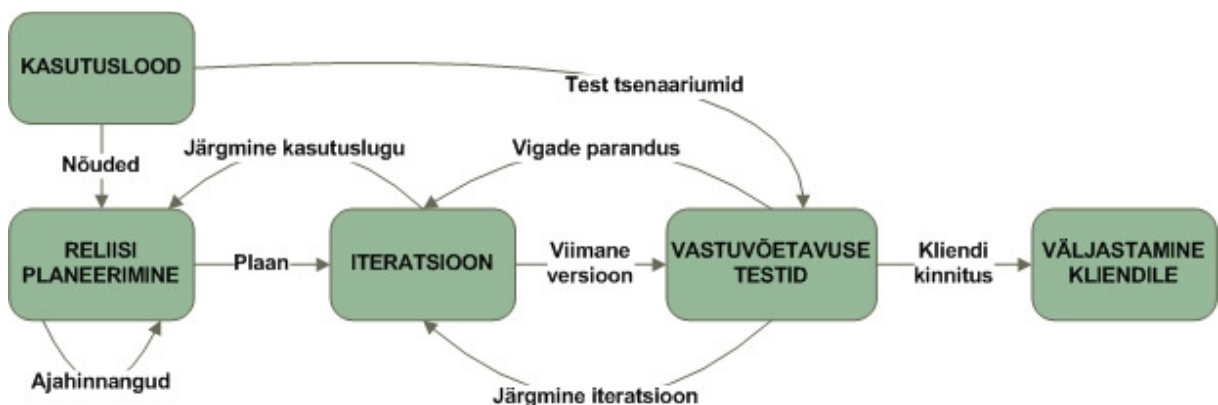
Meeskonnaliikmed suhtlevad kliendi ja teiste osapooltega pidevalt. Tarkvara disain hoitakse lihtne ja puhas. Läbi pideva testimise on tagatud igapäevane tagasiside ning pidev kvaliteedi kontroll. Valmis saadavad süsteemi osad vaadatakse koos kliendiga üle ning vajaduselt tehakse muudatusi (allikas: <http://www.extremeprogramming.org>)

Põhiväärtustest lähtuvalt on XP metoodikale kohandatud 12 parimat praktikad (vt tabel 5).

*Tabel 5. XP metoodika parimad praktikad.*

PRAKTIKA	SELGITUS
<b>Planeerimismäng</b>	Järgmise arendustsükli skoobi täpsustamine ja ajahinnangute andmine.
<b>Väikesed arendustsüklid</b>	Arenda kiirelt väike töötav süsteem, mida vajadusel hiljem täiustatakse.
<b>Metafoor</b>	Ühtne arhitektuuri kirjeldus, mis kõigile osapooltele arusaadav. Ühtne sõnavara.
<b>Lihtne disain</b>	Tarkvara disainimisel tuleb alati lähtuda põhimõttest „lihtne on parim“.
<b>Testimine</b>	Funktsionaalsed testid tuleb automatiseerida.
<b>Refaktoormine</b>	Tarkvara koodist korduste ja üleliigse osa eemaldamine. Eesmärgiks parandada koodi kvaliteeti ja loetavust.
<b>Paarisprogrammeerimine</b>	Koodi kirjutavad kaks inimest ühe arvuti taga, et keskenduda tööle ja vältida vigu.

<b>Kood kui ühine omand</b>	Kood on ühine. Iga arendaja võib suvalist koodirida muuta.
<b>Pidev integratsioon</b>	Iga töötava mooduli valmimise järel teha see teistele kättesaadavaks, et töökulgu oleks pidevalt jälgitav ning töötav prototüüp vastaks viimasele olukorrale arenduses.
<b>40-tunnine töönaeral</b>	Tööta maksimaalselt 40 tundi nädalas. Ületundidega töötamine ei ole pikas perspektiivis jätkusuutlik ning loob pigem kahju kui kasu.
<b>Kliendi kaasamine</b>	Kliendi esindaja töötab koos arendusmeeskonnaga, et tagada kiire tagasiside ja õige suund.
<b>Kodeerimis-standardid</b>	Kogu projekti kood vastab ühtsetele standarditele, mida kõik arendajad järgivad.



**Joonis 6.** Ülevaatluk ekstreemprogrammeerimise protsessiskeem. (Extreme Programming Project).

### 4.2.3 Lepingud agiilses tarkvaraarenduses

Kuigi agiilsete arendusmetoodikate näol on tegemist paindlike süsteemidega, mille põhimõtted on enamasti aktsepteeritavad nii tootjale kui tellijale, siis üheks segadust tekitavaks asjaoluks antud keskkonnas on lepingud. Kuidas agiilsete arendusvõtete korral vormindada lepingud sel moel, et riskid mõlema osapoole jaoks võimalikult madalad oleksid?

Variante on mitmeid, kuid lähtuda tuleks eelkõige kliendi soovidest. Erinevat tüüpi lepingute kirjeldamisel tuginen suuresti Alistair Cockburni blogile (Cockburn, 2006).

- **Aeg ja raha.** Selline kokkulepe sobib siis, kui tellija ja täitja teineteist usaldavad. Põhimõtteliselt on tegu lihtsamat sorti lepinguga, kus tellija maksab täitjale perioodiliselt vastavalt tehtud tööde eest. Tööde teostamiseks kulutatud aeg on hinnatud tundides või päevades ja tasustatakse vastavalt kokkulepitud ühiku hinnale. Selline lepingu vorm on hea siis, kui tööd on muutuva iseloomuga ning puudub konkreetne analüüs. Klient võib muuta oma soove vastavalt vajadusele ning täitja implementeerib soovitud lahenduse ja arveldab pärast vastavalt tööde teostamiseks kulunud ajale. Nagu juba öeldud, siis nõuab antud lepingu vorm osapoolte suurt usaldust ning rakendatakse üldisel siis, kui osapooled on juba korduvalt edukalt koostööd teinud.
- **Fikseeritud hind ja skoop.** Tegu on vana ja levinud põhimõttega. Lähtuvalt tööde iseloomust ja nõuetest ning tuginedes isiklikule intuitsioonile ja kogemusele, anna soovitud töödele hinnang – kui suure raha eest on see võimalik realiseerida? Mõningatel kordadel fikseeritakse tähtaeg, mis ajaks töö peab tehtud olema, kuid põhirõhk on siiski tööde skoobil ja maksumusel. Klient teab mida ta oma raha eest saab ja tööde teostaja on kohustatud fikseeritud tasu eest soovitud lahendused implementeerima. Antud lepingu vorm on sobilik siis, kui osapooled pole varem koostööd teinud ja ei saa teist poolt täielikult usaldada. Antud juhul on oluline ka, et nõuded oleksid üsna täpselt paigas ning neid arenduse käigus oluliselt ei muudetaks. Antud lepingu vormi puhul peaks tööde teostaja mõtlema sellele, et kuidas töö protsess võimalikult efektiivseks teha, et kokkulepitud tööd võimalikult lihtsalt tehtud saaksid. Siin tulevad mängu agiilsed töövõtted ja nn. neljas mõõde, millest eelnevalt juttu oli.
- **Osade kaupa tellimine.** Klient ja tööde teostaja jagavad tööd loogilisteks väikesteks osadeks, kus igale osale määratakse tööde teostamise hind. Sel moel toimub tööde vastuvõtmine ja tasumine väikeste osade kaupa. Tegu on väga mõistliku kokkuleppega. Sel moel on tööde teostaja huvitatud maksimaalsest efektiivsusest ning tööde tellijal on võimalik järgmistesse etappidesse sisse viia vajalikke muudatusi. Ajakriitiliste ja pikalt ette analüüsitud lahenduste teostamiseks pole see parim vairant, kuid agiilsete arendusmetoodikate praktiseerimiseks jällegi ideaalne.

- **Fikseeritud hind, skoop ja tähtaeg.** Juhul kui kliendi vajadused on kõik teada ja tööde teostamise tähtaeg kriitiline, siis tegutsetakse kindla plaani järgi, kus on selgelt teada tähtaeg ja tööde skoop. Tööde algusfaasis lepitakse kokku tööde maksumus ning valmimise tähtaeg. Antud lepingu vorm on väga levinud avalikus sektoris, kus eelarved ja ajalised piirangud on selgelt paigas. Selleks, et riske maandada ja tööd õigeaegselt ajaks valmis saada, tuleb sellistel juhtudel teha kliendiga tihedat koostööd ja muuta kogu arendusprotsess võimalikult läbipaistvaks, et vähimagi eesmärkidest kõrvalekaldumise või vigade ilmnemise korral sellele tähelepanu juhitaks ja puudused kõrvaldataks.

#### 4.2.4 Kokkuvõtte ja soovitused

Kaks tuntumat agiilset arendusmetoodikat on *XP* ja *scrum*. Kuna mõlemad metoodikad kuuluvad agiilsete metoodikate hulka, siis on üldised põhimõtted sarnased – lühikesed arendusetapid, mis jagatud mitmesse iteratsiooni, kommunikatsioon ja vähene dokumentatsioon.

*XP* on arendusmetoodika, mis koosneb suurest hulgast parimatest praktikatest. Antud metoodikas on kesksel kohal suhtlemine, tarkvara võimalikult lihtne disain, pidev tagasiside, tunnustus ja julgus. Selle asemel, et kõike dokumenteerida, eelistatakse verbaalset ja vahetut suhtlemist, et sel moel probleemid kiiremini lahendatud saaks. Kuna dokumentatsioon on teisejärguline, siis on oluline kliendi pidev kaasatus.

Sageli arvatakse ekslikult, et *scrum* on sama mis *XP*, kuid päris nii see pole. *Scrum* on keskendunud pigem tarkvaraprojekti juhtimise poolele ja tehnilistesse detailidesse ei lasku, seega on tegu pigem juhi mõttelaadi ja töökorraldusega, kui konkreetsete arendusreeglitega. *Scrum* on erinevate agiilsete võtete kogum, mis prioritseerib ülesanded vastavalt kliendi vajadustele ning koondab sarnase iseloomuga või muul moel kokku sobivad ülesanded ühte arendusetappi – sprinti. Seega tegeleb *scrum* rohkem tarkvaraprojekti töökorraldusliku ja tööde kontrollimise poolega.

Leidub vähe ettevõtteid, kes on täielikult juurutanud ühe või teise arendusmetoodika. Kogemused näitavad, et parima tulemuse saab erinevate arendusmetoodikate kombineerimisel vastavalt enda vajadustele ja katsetustele. Tuues näitena *XP*, kui ühe levinuma agiilse arendusmetoodika, siis on alustaval tarkvaraarenduse ettevõttel väga keeruline kõiki mainitud

parimaid praktikaid realiseerida ning see ei oleks ka kuigi mõistlik. Alustava ettevõtte puhul on esmatähtis äritegevuse kiire alustamine ning põhitõdede juurutamine. Selge on see, et esimese hooga ideaalset arendusprotsessi tööle saada on võimatud, kuid rakendades mõningaid järele proovitud lahendusi, on tulemus enamasti üsna ootuspärane.

Alustaval ettevõttel on tarvis võita kliendi usaldus, seetõttu peavad protsessid olema lihtsad ja läbipaistavad. *Scrum* on siinkohal suuresti abiks. Esiteks saab kliendile vajadusel öelda, et me kasutame *scrum* arendusmetoodikat, mis seisneb selles, et püüame olla teile võimalikult lähedal ning arvestada teie muutuvaid vajadusi ning pakkuda kiireid tulemusi, mis muudab ka kliendi rahulikumaks, sest isegi kui klient ei tea, mis asi see *scrum* on, siis vähemalt ta teab, et toimitakse mingi kindla plaani alusel. Lihtsustatud kujul võiks kogu protsess näha välja järgmine:

1. Leppige kokku, millised tööd on kõige tähtsamad ja tegelege esialgu ainult nendega.
2. Valige tööd selliselt, et suudaksite nende tegemiseks kuluvat aega adekvaatselt hinnata. Põhimõte on, et üks iteratsioon kestab maksimaalselt kuu aega ja pigem vähem.
3. Järgmiseks leppige kokku kuidas toimub tööde eest tasumine. Parim variant on see, kui arendus ja arveldamine toimub väikeste etappide (kuni 1 kuu) kaupa.
4. Analüüsige iteratsiooni valitud töid ja jagage tööd teostajate vahel ära.
5. Korraldage hommikusi koosolekuid, et jälgida tööde kulgu ja anda tagasisidet.
6. Takistuste ilmnmisel kontakteeruda kliendiga ja leida probleemile lahendus.
7. Testi ise enda tööd ja vajadusel paranda.
8. Tee kõik endast olenev, et töö tähtajaks valmis saaks, vajadusel kaasa täiendavat tööjõudu.

## 5. Infotehnoloogiliste vahendite valik

Kuigi eduka ettevõtluse aluseks on head ideed ja parimate praktikate järgimine, siis ainult sellest siiski ei piisa. Tänapäeval on iga ettevõtte efektiivseks igapäevaseks toimimiseks vaja hulgaliselt infotehnoloogilisi abivahendeid ja kui jutt käib tarkvaraarenduse ettevõttest, siis seda enam.

Tarkvaraarenduse ettevõttes on teatud hulk tegevusi, mille oskuslikuks ja efektiivseks toimimiseks on tarvis erinevaid IT lahendusi. Üks suur probleem paljudele alustavatele ettevõtetele on olnud see, et kuidas sageli vägagi piiratud rahaliste vahenditega soetada äritegevuseks vajalikku tarkvara. Alati on soov ja vajadus rohkema järele, kui alustav ettevõtte endale lubada saaks. Juhul kui äritegevusest midagi välja ei tule ja ettevõtte varases faasis oma tegevuse lõpetab, siis IT lahendustesse investeeritud raha enam tagasi ei saa. Seega tuleb osata riske maandada ja osta esialgu ainult hädavajalikud lahendused või võimaluse korral lausa rentida.

Järgmises peatükis „tarkvara kui teenus“, tuleb juttu kaasaegsetest töövahenditest, mis rahuldavad enamike alustavate tarkvaraarenduse ettevõtete vajadused. See millise arendusplatvormi ja spetsiaaltarkvara iga ettevõtte valib, sõltub ettevõtte eesmärkidest ja pädevusaladest, seega tarkvaraarendajate spetsiaalvahenditest siinkohal juttu ei tule, küll aga on terve hulk üldist töökorraldust ja meeskonnatööd soodustavaid IT lahendusi, mida iga tarkvaraarenduse ettevõtte kasutada saab, et tõhustada asjaajamist ja tõsta tootlikkust.

### 5.1 Tarkvara kui teenus

Tänaseks on turule tulnud mitmed edukad ettevõtted, kes pakuvad teenusepõhiseid terviklikke tarkvaralahendusi, inglise keeles *Software as a Service (SaaS)*. *Gartner Group* väidab, et aastaks 2011 on 25% uutest äritarkvaradest muutunud teenuspõhiseks. (Gartner, 2006). Teenuspõhise tarkvara kasutamisel on mitmeid eeliseid (Wikipedia):

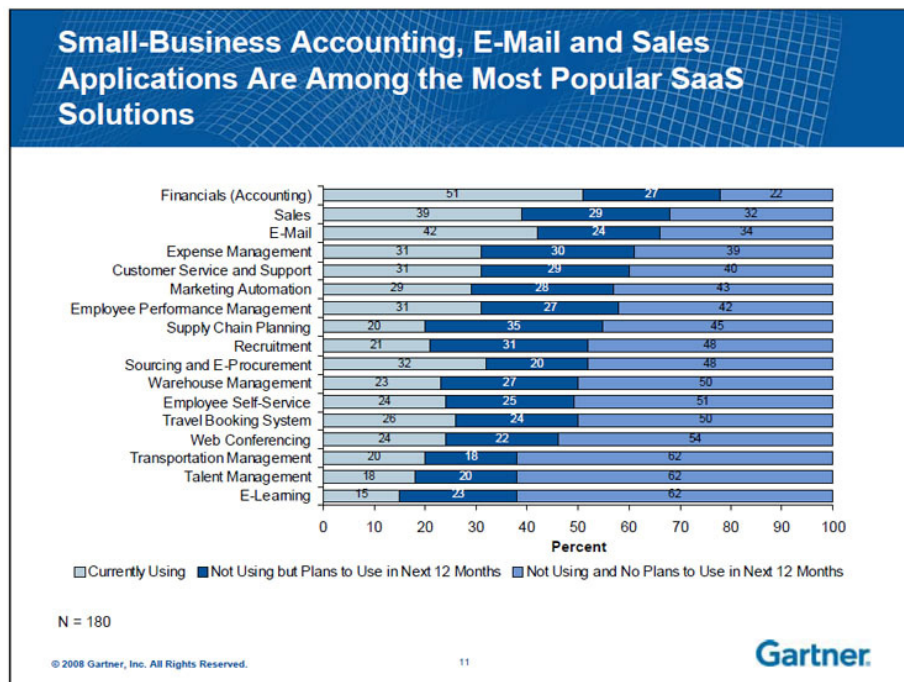
- Pole tarvis kulutada raha serveri ega muu toetava tarkvara peale, kogu töö käib läbi veebibrauseri.
- Selle asemel, et kulutada rahalised ressursid IT infrastruktuuri peale, saab selle raha kulutada konkurentsieelise loomiseks.
- Kallite litsentsitasude asemel igakuised kindlad rahalised kohustused vastavalt valitud teenustele.

- Suurenevate nõudmiste korral on jõudlust võimalik juurde osta, st. maksad ainult selle eest, kui palju ressursse reaalselt tarbid.
- Paindlikkus ja skaleeruvus.
- Kasutamise lihtsus.

Lisaks eelmainitule on olulisemad eelised veel:

- Madalamad kogukulud. Uuringud näitavad, et *SaaS* teenuste kasutamine tuleb keskmiselt 30% odavam võrreldes ettevõtte enda serveri kasutamisega.
- Pole vaja tegeleda serveri hooldamisega.
- Pole vaja tegeleda tarkvara uuendamisega.
- Ei pea muretsema varukoopiate pärast.
- Teenust saab kasutama hakata koheselt – kiire kasutuselevõtt.
- Sobib kõige paremini väikestele ja keskmise suurusega ettevõtetele.

Uuringud näitavad, et peamised põhjused, miks ettevõtted tänapäeval teenuspõhistele tarkvaraplatformidele liiguvad, on kiire kasutuselevõtt ja soodne hind. *Gartner Groupi* uuringute kohaselt on kõige populaarsemad teenuspõhised rakendused väikeettevõtete raamatupidamine ning müügi ja e-maili teenused (vt joonis 7).



**Joonis 7. Populaarsemad SaaS lahendused. (Gartner, 2008).**

Peamine risk on seotud teenust pakkuva ettevõtte kadumisega, mis tähendab, et tööprotsessid tuleb ümber organiseerida ja leida uued tarkvaralised võimalused. Seetõttu on oluline, et valitud teenusepakkuja oleks usaldusväärne. *Gartner Groupi* uuringute kohaselt kulutavad ettevõtted 75% oma IT eelarvest tarkvara ja süsteemide hooldusele, kuid antud olukord on päev-päevalt muutumas.

Hiljutised uuringud näitavad, et teenuspõhine tarkvarakasutus on ennast enamikel juhtudel igati õigustanud. Rohkem kui 95% ettevõtetest, kes teenuspõhist tarkvara kasutavad, plaanivad sama moodi jätkata või selle osakaalu veelgi suurendada (Computerworld, 2010).

Tänaseks päevaks on turule tulnud päris mitu tõsiselt võetavat ettevõtet, kes pakuvad enamlevinud kontoritarkvara ühtse veebipõhise teenusena. See tähendab, et tekstitöötlus, tabelarvutus, esitlusvahendid, märkmikud, koosolekute planeerija, klientide ja projektide haldus jms. tarkvara on kolinud internetti ja igal soovijal on seda koheselt võimalik kasutama hakata. See omakorda aga tähendab, et efektiivne paberimajandus ja koostöö saab hoo sisse kohe ettevõtte loomise esimestel tundidel. Enam pole vaja oodata ja planeerida ning soetada kallist kontoritarkvara. Üheks selliseks kontoritarkvara täislahendust pakkuvaks veebikeskkonnaks on näiteks [zoho.com](http://zoho.com), kuid valikut jagub veel küllaga.

Kuigi teenuspõhine tarkvara rent on mugav ja suhteliselt soodne viis, et kogu oma kontor virtuaalsel kujul korraga tööle rakendada, siis pole see absoluutselt hädavajalik. Kombineerides erinevaid vabavaralisi lahendusi, saab igapäevaseks kontoritööks vajaliku tarkvaraplatvormi kokku panna ka täiesti tasuta. Küll aga on kõigil alustavatel tarkvaraarenduse ettevõtetel üks suur mure – kuidas turvaliselt ja mugavalt majutada tarkvaraprojektide kood jm. materjal? Järgmises peatükis tuleb juttu teenuspõhistest versioonihaldustest – milleks vaja ja kuidas valida!

### 5.1.1 Teenuspõhised versioonihalduse ja meeskonnatöö vahendid

Ettevõtluse alustamine nõuab aega, raha ja planeerimist. Selleks, et algsed investeeringud end ära tasuksid, tuleb hoolikalt läbi mõelda, mida me vajame ning kui palju meil on võimalik nende toodeta ja teenuste eest maksta. Tarkvaraarendusettevõtte puhul on projekti kood ettevõtte pühaks reliikviaks, mille turvalisus peab olema tagatud.

Antud peatükis otsime küsimust järgmistele küsimustele:

- Millised on parimad teenuspõhised versioonihalduse lahendused?
- Kui palju peab alustav ettevõtte investeerima, et saada turvaline versioonihalduskeskkond?

Alustav tarkvaraarendusega tegelev ettevõtte peab klientide usalduse võitmiseks palju pingutama. Üks esimesi asju, millele tuleb rõhku pöörata, on turvalisus. Kui tahad, et sind ja sinu ettevõtet tõsiselt võetaks, kaitse oma koodi! Tarkvaraarendus on peen teadus ja nii nagu teaduses ikka, kipuvad asjad ka siin aegajalt untsu minema. Selleks, et andmed kogemata kaduma ei läheks või ülekirjutatud ei saaks, tuleb appi võtta abivahendid. Versioonihalduse kasutuselevõtt mitte ainult ei ennetata probleemide tekkimist, vaid laseb ka rohkem keskenduda sinu põhitegevusele, ilma, et peaksid pidevalt muretsema oma koodi turvalisuse pärast.

Kui suured ja rikkad ettevõtted saavad endale lubada eraldi serveriparki, kus muuhulgas on tööle seadistatud ka mõni tõhus versioonihalduse repositoorium (*SVN*, *TFS*, vms.), siis alustav ettevõtja endale enamasti sellist investeeringut lubada ei saa, aga see ei saa olla takistuseks. Õnneks on tänapäeval piisavalt palju teenusepakkujaid, kes kas päris tasuta või mõistliku summa eest on nõus sinu arendusprojekti majutama ning sind seeläbi oluliselt aitama. Küsimus on, keda võtta ja keda jätta. Selge on see, et kui me usaldame oma hinnalise arendusprojekti kellegi võõra kätte, siis tahame olla kindlad, et tegu on turvaliste kätega ning loomulikult mängib rolli ka teenuse hind.

Versioonihalduse võlu seisneb selles, et selle abil saavad mitu arendajat sama koodi samal ajal redigeerida, ilma et peaksid muretsema, et nende töö kaotsi läheb ning vajadusel saab faile alati hiljem taastada. Enam pole isikliku failiserveri olemasolu esmatähtis, sest antud teenust on võimalik saada kas päris tasuta või vähemasti mõistliku hinna eest.

Selleks, et selgitada välja, millised on kvaliteetsed ja Eesti ettevõtjale taskukohased internetipõhised versioonihaldusteenused, viisin läbi internetipõhise kontentanalüüsi, kus

võrdlesin ligi 30. internetipõhist versioonihaldusteenust pakkuvat keskkonda. Eesmärgiks oli välja uurida, kui soodsalt on mikroettevõttel võimalik vastav teenus enda käsutusse saada. Sõelale jäid esmajoones need teenusepakkujad, kes on turul juba mõnda aega olnud ning võitnud kasutajate usaldusväarsuse. Lõpuks valisin välja viis parimat keskkonda, mis kõik saavad versioonihaldusteenuse pakumisega suurepäraselt hakkama ning kelle teenust oli mugav kasutada. Nende viie teenusepakkuja lühianalüüsid toon välja järgnevates peatükkides.

Kuna teenuspõhine tarkvara on viimastel aastatel suurt populaarsust võitnud, siis enamik sellist teenust pakkuvatest keskkondadest ei piirdu vaid ühe teenusega. Lisaks versioonihaldusteenusele pakuvad enamik selliseid keskkondi ka muid arendaja ja meeskonna produktiivsuse seisukohalt olulisi tööriistu, nagu näiteks veahaldus, *wiki*, jms. meeskonnatöö ja projektijuhtimise vahendid. Kuigi uurimise peamine tähelepanu langes versioonihalduslahendustele, siis saavad mõningast tähelepanu ka muud produktiivsust tõstvad vahendid, mida analüüsivad keskkonnad pakuvad.

#### **5.1.1.1 Unfuddle**

*Unfuddle* (<http://unfuddle.com>) on turvaline veebipõhine projektijuhtimise ja tarkvaraarenduse keskkond. *Unfuddle* suureks eeliseks on asjaolu, et ta seob palju erinevat funktsionaalsust ühtseks tsentraalseks rakenduseks. Sisaldades nii vigade ja versioonihaldust, *wikit* kui ka ajahaldust, teeb see *unfuddle*st ühe populaarseima ja enimkiidetud teenuse omas vallas. Piiramatu hulk *SVN* ja *Git* repositooriumeid ning võimalus repositooriumis olevat koodi vaadata ja erinevaid versioone omavahel võrrelda.

Kui esialgu oli *Unfuddle* näol tegemist lihtsalt asjaliku *SVN* teenusega, mis võitis paljude arendajate poolehoidu, kuid veidi hiljem, pärast *Git* repositooriumi toe lisamist, asus teenus tugevalt konkureerima *GitHub*-ga, millest ka järgmisena juttu tuleb. Kuid *unfuddle* näol pole tegemist pelgalt välise repositooriumiga, see on midagi palju enamat. *Unfuddle* pakub samas pakettis lisaks ka soliidset kogust projektihalduse lahendusi, sisaldades nii arendajate vahelist sõnumisüsteemi, vigade haldust ja veel paljugi muud, tehes sellest asjaliku ja tervikliku töökeskkonna paljudele arendajatele ja arendusmeeskondadele.

#### **Olulisemad omadused**

- *SVN (Subversion)* tugi – vabavaraline tarkvaralahendus koodi säilitamiseks ning versioonide haldamiseks. Tegu on ühe populaarseima versioonihalduslahendusega,

mida kasutavad miljonid kasutajad igapäevaselt nii vabavaraliste kui ka kommerts lahenduste säilitamiseks ja haldamiseks. *Subversioni* projekt algatati 2000. a. ettevõtte *CollabNet* poolt. Täpsemat infot *Subversioni* kohta leiab kodulehelt <http://subversion.apache.org/>

- Piletid (*tickets*) – loo ja halda veateateid ning hoia projekt organiseeritud. Pileteid saab prioritseerida, kommenteerida ning loomulikult määrata sobilikule meeskonnakaaslasele. Samuti on piletite ajalugu (liikumised, seotud isikud, kommentaarid jms.) hõlpsasti jälgitavad, mis aitab meeskonnal asjade kulgemisel silma peal hoida.
- Ajahaldus (*time tracking*) – *unfuddle* ajahalduse funktsionaalsus võimaldab projekti kulgemisest ja hetke olukorrast saada parema ülevaate. Omades ülevaadet, kui palju töötunde projekti või vaheetapi realiseerimiseks on kulunud ja kui palju veel ette on nähtud, saab reaajas anda projekti kulgemisele adekvaatsemaid hinnanguid ning näha ja ennetada ohte nende varases staadiumis.
- Projektid (*projects*) – ühe konto alt on võimalik hallata kõiki oma erinevaid projekte. Igal projektil on omad piletid, sõnumid jm. varem mainitud funktsionaalsus. Samuti saab määrata projektis osalejad ning anda osalejatele kasutusõigused vastavalt vajadusele ja rollile projektis. Samuti on tasulistel kontodel võimalus oma projekte arhiveerida.
- Armatuurlaud (*dashboard*) – Annab projektis toimuvast tervikliku ülevaade. Kokkuvõtte viimastest sündmustest, projekti verstepostidest ning märkmetest. Üldine informatiivne leht.
- Sõnumid (*messages*) – Selleks, et meeskonnas info liiguks, on *unfuddle* oma rakendusse loonud ka meeskonnasisese suhtluskeskkonna, mis võimaldab teiste projektiosalistega ühenduses olla. Lisaks saab oma sõnumitele lisada ka faile.
- Verstepostid (*milestones*) – annab ülevaate projekti progressist ning võimaldab inimestele ülesandeid jagada.
- Märkmik (*notes*) – on kogum projektiga seotud faile, näiteks dokumentatsiooni failid. Igat märkmikku saab vaadelda kui iseseisvat *wikit*, mis on mõeldud asjakohase info hoidmiseks.
- Õiguste haldus (*Permissions*) – võimaldab igale projekti osatäitjale määrata sobilikud õigused.

## Turvalisus

*Unfuddle* serverite ja tulemüüride haldamisega tegeleb *Amazon*. Vaata lisaks <http://aws.amazon.com>. Andmete transportimine toimub üle 128-bitise SSL ühenduse.

## Maksumus

Kui kahemehe firma saab versioonihalduse kätte täiesti tasuta, siis suurema inimeste arvu korral tuleb pisut juba investeerida (vt joonis 8).

	Private FREE <small>Sign Up</small>	Micro \$9/mo. <small>Sign Up</small>	Compact \$24/mo. <small>Sign Up</small>	Corporate \$49/mo. <small>Sign Up</small>	Enterprise \$99/mo. <small>Sign Up</small>
Storage	200MB	512MB	2GB	4GB	10GB
Active Projects	1	4	10	20	50
Archived Projects	0	4	10	20	50
People	2	10	20	Unlimited	Unlimited
Notebook Pages	3	Unlimited	Unlimited	Unlimited	Unlimited
Repositories Subversion & Git	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Messages	✓	✓	✓	✓	✓
Milestones	✓	✓	✓	✓	✓
Bug Tracking	✓	✓	✓	✓	✓
RSS and iCal	✓	✓	✓	✓	✓
File Attachments		✓	✓	✓	✓
SSL			✓	✓	✓
Time Tracking				✓	✓
Demographic	Small, single developer projects	Small teams of developers	Small teams requiring a secure connection	Larger development teams	Very large development teams

Joonis 8. *Unfuddle* teenuste hinnakiri. (<http://unfuddle.com/about/tour/plans>, 2010)

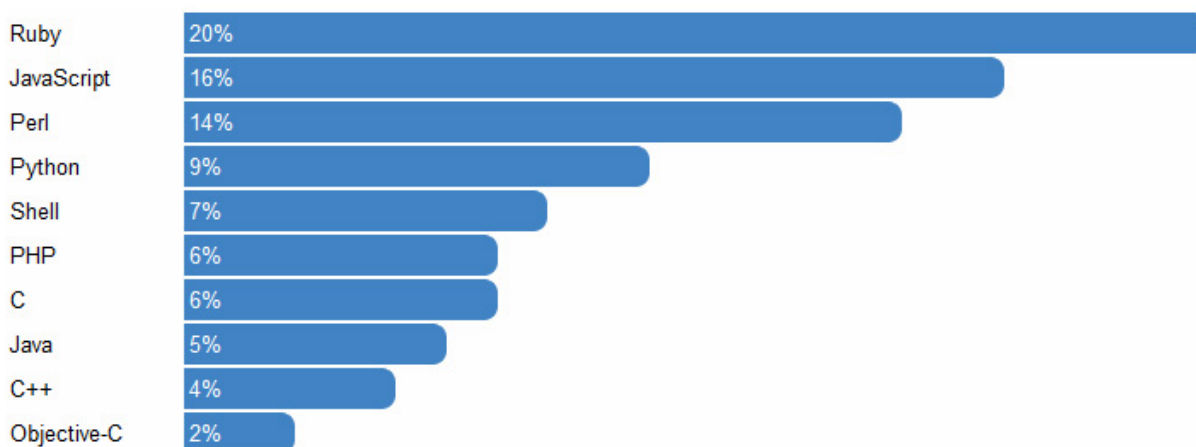
### 5.1.1.2 GitHub

GitHub (<http://github.com/>) on kiire ja efektiivne versioonihalduse süsteem, mis on eelkõige mõeldud tarkvaraarendusega tegelevatele meeskondadele. *GitHub* on veebipõhine platvorm, mis koondab endasse *Git* repositooriumi ja mõned teised meeskonnatööd abistavad vahendid.

Kui *Unfuddle* on mõeldud ainult privaatse koodi majutamiseks, siis *GitHub* võimaldab ka oma projektid teistele avalikuks teha. *GitHub*i näol ongi tegemist suure arendajate sotsiaalse võrgustikuga, kes oma vabavaraalist koodi teistega jagavad. Lisaks koodile jagatakse lahkelt ka ideid ja arvamusi. Seisuga mai 2010 on võrgustikuga liitunud 255000 arendajat ning ühtekokku leiab üle 815000 repositooriumi. Kui soovid, et sinu kood jääks teiste silmade eest nähtamatuks, siis tasuliste pakettidega on see täiesti võimalik.

*GitGub* on enim populaarsust võitnud *Ruby on Rails* arendajate kogukonna seas ning seda peaasjalikult seetõttu, et antud teenus tuli avalikuks 2008. a. juulis, mil vabavaraalne *Rails* arendusraamistik *Rubi* arendajate seas suurt populaarsust kogus. Seega on *GitHub*

suurepärase koost Ruby arendajatele, kes leiavad sealt massilistes kogustes vabavaralisi komponente ning väärt nõu. *GitHubi* kasutajate hulk protsentuaalselt:



**Joonis 9.** *GitHubi kasutajate jaotus arendusplatvormi alusel. (<http://github.com/>, 2010).*

## **Turvalisus**

Kuigi *GitHub* on suurt populaarsust kogunud just avatud lähtekoodiga projektide hulgas, siis kui turvaline on antud teenust kasutada privaatsete projektide hoidmiseks ja kas näiteks *GitHubi* töötajatel on juurdepääs privaatsetele andmetele? Chris Wanstrath, üks *GitHubi* kaasautoritest, kirjutab järgmist: „me loome juurdepääsu klientide privaatsetele repositooriumitele ainult sellisel juhul, kui see on vajalik kliendi abistamiseks. Turvalisus on meie teenuse puhul esmatähtis ning me anname endast maksimumi, et see tõesti oleks turvaline.“ Kõikvõimalike turvalisuse probleemide vältimiseks ja pidevaks täiustamiseks on tööl eraldi turvaekspert. Wanstrath sõnab, et justnimelt turvalisus on peamine põhjus, miks ettevõtted peaksid selle teenuse väljast sisse ostma ja mitte hakkama ise enda serverites projekte hoidma. (Walsh, 2009).

*GitHubi* infrastruktuuri majutab *Rackspace* (<http://www.rackspace.com/>), mis on börsil noteeritud äriühing. Lisaks serveriruumide tugevale turvalisusele on palgatud eraldi spetsialistid ettevõttest *Anchor Hosting* (<http://www.anchor.com.au/dedicated-hosting/dedicated-support.py>), kes tagavad tarkvaralise poole uuenduslikkuse ja turvalisuse. Kõik *GitHubi* vahendusel liigutatavad andmed kantakse üle SSL-i.

## Olulisemad omadused

- *Git* – versioonihaldus.
- Veahaldus (*Issues*) – võimaldab sarnaselt *unfuddle* piletite süsteemile registreerida ja delegeerida vigasid.
- Wiki – Projekti info ja märkmete hoidmiseks.

## Varundamine

Kõik andmed on salvestatud vähemalt kolme eri serverisse.

## Maksumus

Mikroettevõtete jaoks pole kulud suured. Loomulikult tekib mõne aja pärast probleeme puuduliku andmemahuga. Kui juurde maksta ei raatsi ja ülearust funktsionaalsust liiga palju tundub olevat, siis peagi tuleb juttu ka soodsamatest versioonihalduse võimalustest, kuid usaldusväärne ja tõhus keskkond on *GitHub* kindlasti. Maksumuse detailinfot näeb joonisel 10.

Open Source	
<b>Free!</b> <a href="#">Sign Up!</a>	<b>Unlimited</b> Public Repositories <a href="#">?</a> <b>Unlimited</b> Public Collaborators <a href="#">?</a> <b>300 MB</b> Disk Space <sup>1</sup> <a href="#">?</a>
Personal	Business
<b>Micro \$7/month</b> <a href="#">Sign Up!</a>	<b>Large \$50/month</b> <a href="#">Sign Up!</a>
<b>5</b> Private Repositories <a href="#">?</a> <b>1</b> Private Collaborators <a href="#">?</a> <b>600 MB</b> Disk Space <a href="#">?</a> <b>Unlimited</b> Public Repos/Collaborators SSL Protection <a href="#">?</a>	<b>50</b> Private Repositories <a href="#">?</a> <b>25</b> Private Collaborators <a href="#">?</a> <b>6 GB</b> Disk Space <a href="#">?</a> <b>Unlimited</b> Public Repos/Collaborators SSL Protection <a href="#">?</a>
<b>Small \$12/month</b> <a href="#">Sign Up!</a>	<b>Mega \$100/month</b> <a href="#">Sign Up!</a>
<b>10</b> Private Repositories <a href="#">?</a> <b>5</b> Private Collaborators <a href="#">?</a> <b>1.2 GB</b> Disk Space <a href="#">?</a> <b>Unlimited</b> Public Repos/Collaborators SSL Protection <a href="#">?</a>	<b>125</b> Private Repositories <a href="#">?</a> <b>60</b> Private Collaborators <a href="#">?</a> <b>20 GB</b> Disk Space <a href="#">?</a> <b>Unlimited</b> Public Repos/Collaborators SSL Protection <a href="#">?</a>
<b>Medium \$22/month</b> <a href="#">Sign Up!</a>	<b>Giga \$200/month</b> <a href="#">Sign Up!</a>
<b>20</b> Private Repositories <a href="#">?</a> <b>10</b> Private Collaborators <a href="#">?</a> <b>2.4 GB</b> Disk Space <a href="#">?</a> <b>Unlimited</b> Public Repos/Collaborators SSL Protection <a href="#">?</a>	<b>300</b> Private Repositories <a href="#">?</a> <b>100</b> Private Collaborators <a href="#">?</a> <b>60 GB</b> Disk Space <a href="#">?</a> <b>Unlimited</b> Public Repos/Collaborators SSL Protection <a href="#">?</a>

Joonis 10. GitHub-i teenuste paketid. (<http://github.com/plans>, 2010).

### 5.1.1.3 SourceGear Vault

*SourceGear Vault* (<http://www.sourcegear.com/vault/>) on privaatne versioonihalduse teenus, mis on eelkõige mõeldud Microsofti keskse koodibaasi haldamiseks. Tegu on alternatiiviga *Microsoft Visual SourceSafe* tarkvarale. Vault on *SourceSafe*le tugevalt kandadele astunud ja seda ka põhjusega. Tegu on töökindlama ja võimekama abivahendiga kui *SourceSafe*. Vault on integreeritav *Visual Studio*, *Eclipse* ja ka *Dreamweaver*iga. Samuti pakub *SourceGear* tootele suurepärasest kliendituge.

Vault on versioonihaldus väikesest kuni keskmise suurusega arendusmeeskondadele. Vaulti saab kergelt veebipõhise liidese kaudu administreerida, mis võimaldab hallata koodi, meeskonnaliikmete õigusi ja palju muud vajalikku. Veebipõhine liides võimaldab vaadata repositooriumites olevaid faile, võrrelda neid erinevate versioonidega ning vaadata failide ajaloolist infot ja neid ka alla laadida. Arvuti operatsioonisüsteemi installeeritav teenuse klient annab juurdepääsu ja õigused repositooriumis olevate failide haldamiseks mugavalt just nagu *Visual SourSafe*.

Võib öelda, et Vault on parima hinna ja kvaliteedi suhtega versioonihaldus kõigile *Visual Studio* kasutajatele.

Kui teised siin analüüsivad keskkonnad on universaalsed ja sobilikud kõikvõimalike projektide majutamiseks, siis Vaulti soovitan küll ainult *Microsofti* arendusvahendite kasutajatele. Põhjuseks teenuse suhteliselt kallis hind ning kuna tegemist on spetsiifilise erilahendusega, siis täpsemalt hinnapoliitikast siinkohal ei räägi. Hinnakiri on leitav aadressilt <http://www.sourcegear.com/vault/pricing.html>. Kahtlemata parim ja soodsaim leida *Microsofti* keskseks arendamiseks.

### 5.1.1.4 Beanstalk

*Beanstalk* (<http://beanstalkapp.com/>) on privaatne ja turvaline *Subversion* ja *Git* majutusteenus. Seda kasutavad tarkvara meeskonnad, arendajad ja disainerid, kes ei taha kulutada ülearu aega, energiat ja raha.

#### **Omadused**

*Subversion* ja *Git* repositooriumid koos kasutajate haldusega. Annab ülevaate meeskonna produktiivsusest.

## Hind

Tasuta pakett võimaldab 100 MB kettaruumi, 1 privaatset *Git* või *SVN* repositooriumi ning kuni 3 kasutajat. Tõsisemateks ettevõtmisteks tuleb aga leida sobiv pakett vastavalt hinnakirjale (vt tabel 6).

**Tabel 6.** *Beanstalk-i hinnakiri.* (<http://beanstalkapp.com/pricing>, 2010).

	<b>Giant</b>	<b>Corporate</b>	<b>Business</b>	<b>Team</b>	<b>Personal</b>
Kettaruum (GB)	<b>48</b>	<b>24</b>	<b>12</b>	<b>6</b>	<b>3</b>
Kasutajate arv	<b>200</b>	<b>100</b>	<b>40</b>	<b>20</b>	<b>5</b>
Repos. arv	<b>150</b>	<b>65</b>	<b>30</b>	<b>15</b>	<b>10</b>
<b>Hind (\$/kuu)</b>	<b>200</b>	<b>100</b>	<b>50</b>	<b>25</b>	<b>15</b>

## Eripära

- Beanstalki *S/FTP Deployment* tööriist võimaldab soovitud versiooni repositooriumist otse soovitud serverisse eksportida (*deploy*). Lihtne veebipõhine liides, mis hõlbustab failide transportimist ja rakenduse tööle seadmist teistes serverites.
- Võimaldab liidestamist mitmete väliste teenustega, muuhulgas *Basecamp-i*, *Lighthouse-i* ja *FogBugz-ga* (veebipõhised meeskonnatöö ja produktiivsuse vahendid).

## Turvalisus

Kõik *Subversioni* repositooriumid ja serverid on majutatud *Rackspace* andmekeskustesse. Kõikidest andmetest hoitakse koopiaid *Amazoni S3* serverites. Veebikeskkond ja *Subversioni* serverid kasutavad *SSL* krüpteeringut.

## Kokkuvõte

*Beanstalki* näol on kahtlemata tegemist professionaalse ning mitmekülgse lahendusega, mille suureks eeliseks on liidestamise võimalus mitmete väliste lahendustega. Peamiseks puuduseks on tasuliste pakettide suhteliselt kõrge hind.

### 5.1.1.5 ProjectLocker

*ProjectLocker* (<http://projectlocker.com/>) on lihtne ent efektiivne abivahend kõigile, kes otsivad hea hinna ja kvaliteedi suhtega versioonihalduse teenust. Kuigi lisaks *Subversioni* ja *Git* repositooriumitele saab kasutada ka teisi tööriistu, tahaks antud teenuse puhul esile tõsta just head hinna ja kvaliteedi suhet.

#### Omadused

*Subversion-i* ja *Git-i* repositooriumid ning *Trac* (vigade raporteerimine, *wiki*, lähtekoodi lehitseja). Kombinatsioon tõhusatest tööriistadest, mida iga arendustiim vajab.

#### Maksumus

Antud teenuse plussiks on hind ja paindlikkus. Tasuta kontosid saab registreerida piiramatul hulgal ja siduda need kõik ühe e-maili aadressiga! Täpsem hinnakiri on toodud tabelis 7.

*Tabel 7. ProjectLocker'i hinnakiri. (<https://projectlocker.com/signup/startup>, 2010).*

	Free	Seed	Angel	Venture	Equity
Kettaruum (GB)	0,5	2,5	5	10	30
Kasutajate arv	5	5	5	15	30
Projektide arv	piiramatult	piiramatult	piiramatult	piiramatult	piiramatult
Repos. arv	piiramatult	piiramatult	piiramatult	piiramatult	piiramatult
<b>Hind (\$/kuu)</b>	<b>0</b>	<b>2.08</b>	<b>5</b>	<b>12</b>	<b>30</b>

#### Eripära

Kõige parema hinnaga laialt kasutatud ja usaldusväärne *Subversion* ja *Git* repositoorium. Lisaks levinud ja paljudele arendusmeeskondadele tuttav vigade raporteerimise ja koostöö vahend *Trac*. Tasulistes pakettides on võimalus kasutada ka mitmeid automatiseeritud vahendeid nagu *BuidlLocker*, mis võimaldab *Java* ja *.NET* projektide ajastatud *builde* (repositooriumis olevate failide liigutamist arenduse, testimise, *live* vms. rakendusserverisse) ning automaatset testimist. Kogu tegevuste ajalugu logitakse.

## **Turvalisus**

kogu juurdepääs andmetele käib läbi *SSH/SSL* turvakanali. Serverid asuvad monitooritud tule müüride taga. *ProjectLocker* ei vaatle ega kasuta sinu privaatsid andmeid ilma mõjuva põhjuseta.

## **Kokkuvõte**

*ProjectLocker*i näol on tegemist tõeliselt positiivse leiuga, mis on tegutsenud juba aastast 2003 ning võitnud paljude arendajate poolehoidu. Antud teenuse ainukese puudusena võib välja tuua vananenud ja kohati keerulise kasutajaliidese, kuid kui kasutada ainult *Subversioni* või *Git*i repositooriumit, siis polegi põhimõtteliselt antud keskkonda vaja rohkem külastada, kui et ainult konto loomiseks ja edasine tegevus käib juba läbi operatsioonisüsteemi installeeritud *SVN* või *Git* kliendi. Seega ei soovita ma antud teenust kasutada kui terviklikku (*All-in-one*) teenust, vaid kui turvalist hea hinna ja kvaliteedi suhtega versioonihalduse lahendust.

### **5.1.1.6 Kokkuvõte ja soovitused**

Eelmainitud keskkonnad said teiste seast välja valitud ja lähema uurimus alla võetud eelkõige seetõttu, et tegu on kvaliteetset teenust pakkuvate usaldusväärsete lahendustega. Kui isikliku serveri ja spetsiaaltarkvara soetamine ja häälestamine on kulukas ja aeganõudev ettevõtmine, siis versioonihaldusteenuse kasutamine on suhteliselt odav, kiire ja kindlasti väga kasulik ja isegi hädavajalik. Kui on soov leida ainult head versioonihalduse lahendust ja muu meeskonnatöö funktsionaalsus nii väga ei huvita, siis on parimaks lahenduseks viimati kirjeldatud *ProjectLocker*, mis tänu väga soodsatele tingimustele ja kõrgele kvaliteedile on suurepärase lahenduse koodi turvalise haldamise muredele. Kui on vaja head versioonihalduse lahendust Microsoft vahenditele, siis soovitan valida *SourceGear Vault*. Ülejäänud lahendused – *Unfuddle*, *GitHub* ja *Beanstalk* on kõik suurepärase lahendused, mis lisaks versioonihaldusele võimaldavad kasutada ka peamisi meeskonna produktiivsust tõstvaid lahendusi ning on seetõttu võrreldes *ProjectLockeriga* pisut kallimad.

## Kokkuvõte

Kuigi ettevõtluse alustamine on lihtne, siis edu saavutamise nimel tuleb näha märgatavalt rohkem vaeva. Statistika näitab, et umbes kolmandik IKT sektori mikroettevõtetest ei ole ühel või teisel põhjusel jätkusuutlikud. Üldised probleemid, mis antud valdkonda puudutavad, on madal tootlikkus ja kvaliteet, mis on tingitud eesmärgipäratust tegevusplaanist ja puudulikust algkapitalist. Selleks, et alustavas mikroettevõttes edu saavutada, on vaja plaani ehk strateegiat, mis suunaks tegevust ning kontrolliks käike. Strateegiliste eesmärkide elluviimist soodustavad oskuslikult valitud töömeetodid ja toetav infotehnoloogiline keskkond.

Oma uurimistöö vältel juhtisin tähelepanu mitmetele probleemidele ning tulin välja soovitusetega, mis aitavad mitmeid vigu vältida. Probleemide väljatoomisel ning lahenduste pakkumisel lähtusin eelkõige tarkvaraarendusega tegelevate ettevõtete vaatenurgast ning seda põhjusel, et antud uurimistöö autor on ise just selles vallas tegev ning tunneb selle keskkonna eripärasid kõige paremini.

Juhtimismudelite olemusest järeldub, et ettevõtte edukus on tugevalt seotud ettevõtte strateegilise mõtlemisega. Selleks, et teha õigeid otsuseid, peab teadma ettevõtte tugevusi ja nõrkusi ning lähtuma seatud eesmärkidest. Peamised juhtideed, mis antud uurimise käigus välja tulid, on järgmised:

- a) Ettevõtte vajab strateegiat.
- b) Strateegiast lähtuvalt valitakse protsessid.
- c) Tulemus olgu kvaliteedile suunatud.
- d) Tootlikku äritegevust toetavad IT lahendused ei pea ilmtingimata olema kallid, tähtis on osata valida.

Alustava mikroettevõtte jaoks on strateegia loomine mitteformaalne protsess. Esmase strateegiline plaan võib olla üles ehitatud küsimuste-vastuste vormis. Konkurentsieelise loomiseks ja usalduse võitmiseks pööratakse peatähelepanu hinnale, kvaliteedile, kiirusele ja paindlikkusele.

Protsessid alustavas tarkvaraarenduse ettevõttes võiksid olla paindlikud ja kohandatud vastavalt muutuvatele vajadustele. Arendusprotsessid peaksid toetuma agiilsetele meetodikatele, mis toetavad muutuvat keskkonda ja kliendi nõudmisi parimal viisil. Parim viis toote või teenuse kvaliteedi tagamiseks on vahetu suhtlus, pidev testimine ja kliendi kaasatus.

Infotehnoloogiliste vahendite valikul ei kehti alati põhimõte „mida kallim, seda parem“. Oskuslikult valitud töökeskkonna ja produktiivsust tõstva tarkvaralise platvormi saab üles ehitada vägagi piiratud ressurssidega. Tuleb osata valida. Teenuspõhine tarkvara vabastab ettevõtjad kallitest litsentsilepingutest ja suurtest summadest, mis kuluks tarkvara uuendamise peale.

Tarkvaraettevõtte pühaks reliikviaks on tarkvara kood. Klientide usalduse võitmiseks ja efektiivseks meeskonnatööks tuleb ühelt poolt tarkvarakoodi tugevalt kaitsta, teiselt poolt teha see kõigile meeskonnaliikmetele lihtsalt kättesaadavaks. Internetipõhised versioonihalduskeskkonnad, mis võtavad vastutuse enda peale ning vabastavad alustavad ettevõtjad spetsiaalselt seadistatud faili-, varundamis- vms. serverite soetamisest, on siinkohal suuresti abiks.

Planeerides ettevõtte tegemisi veidi ette - sõnastades peamised eesmärgid, kirjeldades põhiprotsessid ja valides infotehnoloogilised vahendid oskuslikult, saab säästa nii aega kui ka raha. Toimides läbimõeldud tegevusmustril alusel ja pidevalt kontrollides enda tegevuse vastavust seatud eesmärkidele, loob iga alustav (tarkvara) ettevõtte märkimisväärse konkurentsieelise.

## Summary

The title of the master thesis is „Recommendations on Choosing IT Tools and Methods for Productivity Management in a Starting Software Development Company“.

Keywords: productivity management, software start-up, tools and methods for productivity.

The main objectives of this master thesis are to analyze the main mistakes and problems in a software start-up company and to come up with recommendations to reduce the risk of failure and to increase productivity.

This master thesis gives an overview of management models, strategic components, software development processes and IT tools for productivity. The author of this thesis recommends choosing agile development methodologies which help to reduce the cost of change and assure better quality. All the decisions in a software start-up should follow the simple rules established by strategic objectives.

Choosing the best IT tools that money can buy is essential to team's productivity. But not having the unlimited budget for an IT tools is not a big problem anymore. The author of this master thesis recommends *SaaS* solutions to immediately start your business and gain productivity. Special focus is set on version control systems.

## Kasutatud kirjandus

1. **Statistikaamet.** (2009). Statistikaameti andmebaas. URL: <http://pub.stat.ee/px-web.2001/Dialog/statfile2.asp> (05.05.2010)
2. **IKT sektori lõpparuanne.** (2009). Rozeik, H. Jürgenson, A. Poliitikauuringute Keskus PRAXIS. Eesti info- ja kommunikatsioonitehnoloogia sektori ettevõtete uuring. URL: <http://www.itl.ee/?dl=186> (05.05.2010)
3. **Eesti infotehnoloogia tulevikuvaated.** (2009). Tiits, M. Rebane, K. URL: <http://www.itl.ee/?dl=43> (05.05.2010)
4. **Riigikontroll.** (2010). Mattson, T. Sageli ei ole riigi infosüsteemide arendamine tulemuslik.  
URL:[http://www.riigikontroll.ee/fake\\_index.php?lang=et&uri=%2Fpressiteated.php%3Fkirjeldused%3D1%26lang%3Det](http://www.riigikontroll.ee/fake_index.php?lang=et&uri=%2Fpressiteated.php%3Fkirjeldused%3D1%26lang%3Det) (05.05.2010)
5. **Leimann, Skärvad, Teder.** (2001) Strateegiline juhtimine.
6. **Pryor, M. G., Anderson, D., Toombs, L., Humphreys, J. H. (2007).** Strategic implementation as a core competency: The 5P's model. Journal of management Research.
7. **The McKinsey 7S Framework.** URL: [http://www.mindtools.com/pages/article/newSTR\\_91.htm](http://www.mindtools.com/pages/article/newSTR_91.htm) (05.05.2010)
8. **Mintzberg, H.** (1994) The Rise and Fall of Strategic Planning.
9. **Valmra, E.** (2007) URL: <http://www.juhtimine.ee/strateegiline-juhtimine-vaikefirmas> (05.05.2010)
10. **Pryor, M.G. Toombs, L., Anderson, D. White, J.C.** What Management and Quality Theories Are Best for Small Businesses? Journal of Management and Marketing Research.
11. **Walsh, B.** (2009). The Web Startup Success Guide.
12. **Spring, M., Boaden, R.** „One more time, how do you win orders?“. a critical reappraisal of the Hill Manufacturing Strategy Framework. URL: <http://www.emeraldinsight.com/Insight/ViewContentServlet?Filename=/published/emeraldfulltextarticle/pdf/0240170803.pdf> (05.05.2010)
13. **Cockburn, A.** (2003). Process: the 4th dimension.  
<http://alistair.cockburn.us/Process:+the+4th+dimension> (05.05.2010)
14. **Boehm, B., Turner, R.** (2004) Balancing Agility and Discipline.
15. **Scrum** (2009). *Miks Scrum?* URL: <http://www.scrum.ee/scrum> (05.05.2010)

16. **Beck, K.** (2000) *Extreme Programming Explained: Embrace Change*.
17. **Extreme Programming Project.** URL:  
<http://www.extremeprogramming.org/map/project.html> (05.05.2010)
18. **Cockburn, A.** (2006). *Agile Contracts*. URL:  
<http://alistair.cockburn.us/Agile+contracts> (05.05.2010)
19. **Gartner Group.** (2006). *Gartner Says 25 Percent of New Business Software Will Be Delivered As Software As A Service by 2011*. URL:  
<http://www.gartner.com/it/page.jsp?id=496886> (05.05.2010)
20. **Wikipedia.** *Software as A Service*. URL:  
[http://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Software_as_a_service) (05.05.2010)
21. **Gartner Group.** (2008). *User Survey Analysis: Software as a Service, Enterprise Application Markets, Worldwide*.
22. **Computerworld.** (2010). *More Than 95% of Organizations to Maintain or Grow SaaS Use*. URL: <http://computerworld.com.ph/more-than-95-of-organizations-to-maintain-or-grow-saas-use> (05.05.2010).

## Kasutatud lühendid ja võõrkeelsed terminid

IT – infotehnoloogia.

IKT – info- ja kommunikatsioonitehnoloogia.

SaaS – teenuspõhine tarkvara (*Software as a Service*).

Agiilne tarkvaraarendus – väledatel (agiilsetel) meetodikatel põhinev tarkvaraarendus.

XP – ekstreemprogrammeerimine. Agiilne tarkvaraarenduse meetodika.

Scrum – agiilne tarkvaraarenduse meetodika.

Wiki – veebipõhine koostööplatvorm.

Subversion (*SVN*) – versioonihalduse süsteem.

Git – versioonihalduse süsteem.

Repositorium – informatsiooni (projekti koodi) säilitav andmebaas.

SSL – infoturbe protokoll (*Secure Sockets Layer*).

SSH – turvaline võrguprotokoll (*Secure Shell*).

Microsofti Visual SourceSafe (*VSS*) – Microsofti versioonihaldustarkvara.