

TARTU ÜLIKOOL  
MATEMAATIKA-INFORMAATIKATEADUSKOND  
Arvutiteaduse instituut  
Infotehnoloogia eriala

Karl Blum  
**Nõuete haldamise tarkvara Volere metoodikale**  
Bakalaureusetöö (4 AP)

Juhendaja: Indrek Sander, MSc

Autor: ..... “.....” mai 2008  
Juhendaja: ..... “.....” mai 2008

Lubada kaitsmisele

Professor: ..... “.....” mai 2008

## Sisukord

Sissejuhatus .....	3
1. Volere tehnika .....	4
1.1 Volere nõuete mall .....	4
1.2 Volere nõuete kategooriad .....	4
1.3 Nõuete parameetrid .....	6
2. Nõuded tarkvarale .....	7
3. Tarkvaraprojekti kirjeldus .....	13
3.1 Kasutatud tehnoloogiad .....	13
3.2 Kasutatud raamistikud .....	14
3.3 Andmemudeli kirjeldus .....	14
4. Loodud tarkvara kirjeldus .....	16
4.1. Ülevaade .....	16
4.2. Kasutamine .....	16
4.3. Paigaldamine .....	21
Kokkuvõte .....	23
Kasutatud kirjandus ja materjalid .....	24
Requirements Managment Software For Volere Method .....	25
Lisad .....	26
Lisa 1. Programmi lähtekood ja installatsioonifail CD-l .....	26

## Sissejuhatus

Käesoleva bakalaureusetöö üheks eesmärgiks on tutvuda tarkvara nõuete haldamise metoodikaga *Volere*. Teine eesmärk on luua programm, mille abil on võimalik hallata tarkvaranõudeid, kasutades *Volere* metoodikat. Loodav tarkvara on tõenäoliselt esimene omalaadne [6] – mugav, lihtne ja praktiline vahend nõuete haldamiseks *Volere* metoodika järgi. Hetkel ei ole allikas [6] välja toodud nimistus ühtegi programmi, mis spetsialiseeruks *Volere* metoodikale. Antud töö raames loodava tarkvara funktsionaalsus jääb suhteliselt kitsastesse piiridesse – eesmärk on tagada põhifunktsioonide (otsimine, lisamine) lihtsus ja kasutusmugavus.

Töö esimeses peatükis tutvume *Volere* metoodikaga ja selgitame *Volere* nõuete malli sisu ning olemust. Teises peatükis püstitame nõuded loodavale tarkvarale *Volere* metoodikat rakendades. Kolmandas peatükis kirjeldame tarkvaraprojekti tehnilist külge ning neljandas peatükis võtame kokku loodud tarkvaraga seotud teemad. Valminud tarkvara on salvestatud lisas asuvale CD'le.

## 1. Volere tehnika

Sõna *Volere*, tuleneb itaalia keelsest sõnast „soovima“. Selle sõnaga hakkasid neli ettevõtte Atlantic System Guild töötajat kirjeldama tarkvara nõuetega seotud teatud tüüpi tehnikat. Selles kontekstis mõeldakse *Volere* all nõuete kogumise protsessi, nõuete malli ja sellega seotud koolitusi ning treeninguid. Käesoleva töö raames on käsitletud *Volere* nõuete kirjeldamise malli [1]. *Volere* põhiline eesmärk on luua ühtne vorm kõikide tarkvara nõuete haldamiseks ja seda läbi universaalse nõuete malli. *Volere* tehnikad on skaleeruvad ja paindlikud, mis teeb nende kasutamise võimalikuks väga erinevates projektides. *Volere*-t on edukalt rakendatud juba paljudes organisatsioonides [5].

### 1.1 Volere nõuete mall

Üks kõige olulisem osa *Volere*-st on nõuete haldamise mall. Selles mallis on ära kirjeldatud 27 suuremat nõuete kategooriat, mille eesmärk on ära katta kõik loodava tarkvaraga seotud nõuded ja kirjeldused. Põhikategooriad võib jaotada viite erinevasse gruppi, kuigi nõuete kirjeldamisel gruppi kuuluvust otseselt ei määrata, vaid viidatakse ühele konkreetsele põhikategooriale ja selle alamkategooriale.

### 1.2 Volere nõuete kategooriad

*Volere* nõuete mall sisaldab 27 suuremat nõuete kategooriat, millest iga kategooria sisaldab omakorda hulka alamkategooriaid.

Põhikategooriate jaotused:

1. Projekti edasiviijad
  - 1) Projekti eesmärk
  - 2) Klient, tellija, mõjutatud osapooled
  - 3) Tarkvara kasutajad
2. Projekti piirangud
  - 4) Kohustuslikud kitsendused
  - 5) Nimetusviisid ja definitsioonid
  - 6) Faktid ja oletused
3. Funktsionaalsed nõuded
  - 7) Töö skoop

- 8) Toote skoop
- 9) Funktsionaalsed- ja andmenõuded
- 4. Mittefunktsionaalsed nõuded
  - 10) Väljanägemine
  - 11) Kasutatavus ja inimlikkus
  - 12) Töökiirus
  - 13) Füüsiline töökeskkond
  - 14) Hallatavus ja kasutajatugi
  - 15) Turvalisus
  - 16) Kultuur ja poliitika
  - 17) Õiguslikud nõuded
- 5. Projekti arutlused
  - 18) Avatud probleemid
  - 19) Olemasolevad tooted
  - 20) Uued probleemid
  - 21) Ülesanded
  - 22) Versiooni vahetus
  - 23) Riskid
  - 24) Maksumus
  - 25) Kasutajajuhendid ja väljaõpe
  - 26) Ooteruum
  - 27) Ideed ja lahendused

Igal põhikategoorial on hulk alamkategooriaid, mis kirjeldavad juba ühte konkreetset nõude liiki täpsemalt. *Volere* malli põhjal kirjeldatud nõuete hierarhia on kahetasandiline, kuid nõudeid seostatakse vaid madalaima tasemega. Ehk iga kirjeldatav nõue kuulub ühe põhikategooria alamkategooriasse. Näiteks olgu meil tarvis kirjeldada tarkvarale andmenõudeid. *Volere* järgi kuuluvad tarkvarale kirjeldatavad andmenõuded üheksanda põhikategooria teise alamkategooriasse. Ametlikus *Volere* mallis tähistatakse alamkategooriad tähtedega (a,b,c,...), kuid antud töö raames loodavas tarkvaras on võetud lihtsuse mõttes tähtede asemel kasutusele alamkategooria numbrita tähistamist (1,2,3,...). Seega andmenõuete tüüp oleks: 9.2 (ametliku *Volere* malli järgi 9.b).

### 1.3 Nõuete parameetrid

Kõikidel nõuetel, kõikides alamkategoriates on ühtne kirjeldamise vorm ja ühised sisu komponendid. Ühe konkreetse nõude kirjeldamise vormi nimetatakse nõude kestaks (inglise keeles *snowcard*). Nõude kest näeb oma olemuselt välja kui üheleheline vorm, millele on ära paigutatud nõude kirjeldamiseks vajalikud lüngad:

- 1) Nõude identifikaator – unikaalne number, mille alusel on võimalik viidata ühele konkreetsele nõudele
- 2) Nõude tüüp – nõude tüübiks on üks konkreetne *Volere* malli kategooria number
- 3) Kasutuslood – viited kasutuslugude identifikaatoritele, mis on antud nõudega seotud
- 4) Nõude kirjeldus – nõuet kirjeldav tekst
- 5) Nõude vajalikkus – põhjendus, miks nõuet tarvis on
- 6) Algataja – nõude algne tekitaja
- 7) Sobivustingimus – kriteerium, mille alusel on hiljem võimalik kontrollida nõude täidetust
- 8) Kliendi rahulolu – 5 palli süsteemis hinnang kliendi rahulolu kohta antud nõude täitmisel. Suurem number tähistab suuremat rahulolu.
- 9) Kliendi rahulolematust – 5 palli süsteemis hinnang kliendi rahulolematusele nõude realiseerimata jätmisel. Suurem number tähistab suuremat rahulolematust.
- 10) Prioriteet – 5 palli süsteemis hinnang nõude olulisusele. Suurem number tähistab kõrgemat prioriteeti.
- 11) Konfliktid – viited antud nõudega vastuolulistele nõuete identifikaatoritele
- 12) Lisamaterjalid – viited lisamaterjalidele (dokumendid, pildid, skeemid)
- 13) Muudatuste ajalugu – kirjeldus muudatustest mis on antud nõudega toimunud

*Volere* nõude kestas olev informatsioon ei pruugi olla iga nõude kohta alati siiski täielik ja seetõttu on võimalus mallilt viidata välistele lisamaterjalidele. Üldjuhul nõude kestas asuv informatsioon on täiesti piisav nõudega tegelemiseks.

## 2. Nõuded tarkvarale

Järgnevas kirjeldame *Volere* metoodikast lähtudes nõuded antud töös loodavale tarkvarale. Selguse mõttes toon sulgudes välja nõude tüübi nimetuse iga kirjeldatud nõude juures.

### Nõue #1

<b>Tüüp</b>	1.2 (Projekti eesmärgid)		
<b>Kirjeldus</b>	Luuu tarkvaranõuete haldamise süsteem, mis baseerub <i>Volere</i> metoodikal.		
<b>Nõude vajalikkus</b>	Hetkel ei ole olemas mõistlikku süsteemi <i>Volere</i> metoodikal põhinevat nõuete haldamise keskkonda.		
<b>Sobivustingimus</b>	Valmis nõuetekohane tarkvara, mis aktsepteeritakse juhendaja poolt.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
5	5	4	

### Nõue #2

<b>Tüüp</b>	4.6 (Ajalised piirangud)		
<b>Kirjeldus</b>	Esitamise tähtaeg 28. mai.		
<b>Nõude vajalikkus</b>	Ülikooli esitamise tähtaeg.		
<b>Sobivustingimus</b>	Nõuetekohane tarkvara installatsioonifail ja kasutusjuhend on tähtajaks üle antud.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
5	5	1	

### Nõue #3

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)		
<b>Kirjeldus</b>	Saab nõudeid sisestada vormilt mis on sarnane <i>Volere</i> metoodikas kirjeldatud nõuete malliga.		
<b>Nõude vajalikkus</b>	Tarkvara ülesanne on lihtsustada <i>Volere</i> metoodika kasutamist.		
<b>Sobivustingimus</b>	Olemas on <i>Volere</i> -st lähtuv andmete sisestamise võimalus.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
5	5	5	

### Nõue #4

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)		
<b>Kirjeldus</b>	Olemas on nõuete nimekiri, millelt on võimalik otse vastavat nõuet avada.		

<b>Nõude vajalikkus</b>	Nõuete haldamiseks on tarvis näha nõuete nimekirja.	
<b>Sobivustingimus</b>	Nõude vahetu realiseatsioon tarkvaras.	
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>
<b>4</b>	4	4

#### Nõue #5

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)		
<b>Kirjeldus</b>	Nõuete nimekirja peab saama filtreerida järgmiste parameetrite järgi: <ul style="list-style-type: none"><li>• Nõude tüüp</li><li>• Koostamise aeg</li><li>• Viimase muutmise aeg</li><li>• Prioriteetid</li><li>• Algataja</li><li>• Konfliktid</li><li>• Sõne sisaldus</li></ul>		
<b>Nõude vajalikkus</b>	Kui nõudeid tekib nimekirja palju, siis on nende haldamine keeruline. Filtreerimine võimaldab korraga töötada just hetkel vajalike nõuetega.		
<b>Sobivustingimus</b>	Võimalik valida soovitud kriteeriumid ja nende alusel nõuete nimekirja filtreerida.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
4	3	3	

#### Nõue #6

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)	
<b>Kirjeldus</b>	Nõuete nimekirjaga peab koos töötama eelvaate aken, mis näitaks nimekirjast valitud nõude sisu vastavalt eelvaate parameetritele (kirjeldus, nõude vajalikkus, sobivustingimus).	
<b>Nõude vajalikkus</b>	Eelvaate aknaga lihtsustatakse soovitud nõude otsimist, kuna nõudest ülevaate saamiseks ei pea seda avama. Näiteks saab sirvida järjest nõuete kirjeldusi.	
<b>Sobivustingimus</b>	Loodud eraldi aken, milles kajastub vastavalt valitud nõudele selle sisu osa.	
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>
2	2	1

#### Nõue #7

<b>Tüüp</b>	9.2 (Andmenõuded)
<b>Kirjeldus</b>	Kogutakse iga nõude kohta järgnev info: <ul style="list-style-type: none"> <li>• Kirjeldus</li> <li>• Põhjendus</li> <li>• Sobivustingimus</li> </ul>



- Prioriteet
- Kliendi rahulolu
- Kliendi rahulolematus
- Algataja
- Kasutuslood
- Ressursid
- Konfliktid
- Loomise aeg
- Muutmise aeg

<b>Nõude vajalikkus</b>	Andmeväljad on <i>Volere</i> mallist lähtuvad.		
<b>Sobivustingimus</b>	Nõude vahetu realiseerimine tarkvaras.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
4	5	4	

#### Nõue #8

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)		
<b>Kirjeldus</b>	Iga nõude kohta on võimalik märkida sellega konfliktis olevad nõuded. Konflikt on nähtav mõlemas konfliktis olevas nõudes.		
<b>Nõude vajalikkus</b>	Nõuete kogumisel on võimalik, et mingis arendusfaasis on nõudeid, mis on üksteisega vastuolus. Need on vaja ära kirjeldada, et probleemi oleks võimalik hiljem lahendada.		
<b>Sobivustingimus</b>	On võimalus lisada iga nõude kohta temaga konfliktis olevate nõuete numbreid. Konflikt on nähtav mõlemas konfliktis olevas nõudes.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
3	4	3	

#### Nõue #9

<b>Tüüp</b>	9.2 (Andmenõuded)		
<b>Kirjeldus</b>	Iga nõude kohta peab saama kirjeldada sellega seotud ressursid tekstilisel kujul.		
<b>Nõude vajalikkus</b>	Võib olla vajadus konkreetsest nõudest välisele objektile (link, dokument vms) viitamiseks.		
<b>Sobivustingimus</b>	Võimalik sisestada tekstilisel kujul viiteid ressurssidele.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
3	3	2	

#### Nõue #10

<b>Tüüp</b>	10.2 (Nõuded programmi stiilile)		
<b>Kirjeldus</b>	Nõuete aknad peavad avanema programmi põhiakna sees. Põhiakna alamaknaid		

	peab olema võimalik paigutada vastavalt soovile.		
<b>Nõude vajalikkus</b>	Tarvis nõuete mugavaks haldamiseks ja mitme nõudega paralleelselt töötamiseks.		
<b>Sobivustingimus</b>	Võimalik peaaknas avada mitu nõuet korraga ja aknaid erinevatesse kohtadesse paigutada.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
3	4	2	

### Nõue #11

<b>Tüüp</b>	11.3 (Nõuded õpitavusele)		
<b>Kirjeldus</b>	Tarkvara peab olema võimalik probleemivabalt kasutada peale kasutusjuhendi läbi töötamist ja <i>Volere</i> metoodikaga tutvumist.		
<b>Nõude vajalikkus</b>	Kasutajaliides ei pruugi olla täiesti intuitiivne, kuid on kergesti õpitav.		
<b>Sobivustingimus</b>	Kui kasutaja loeb läbi kasutusjuhendi ja on kursis <i>Volere</i> metoodikaga, siis suudab ta nõudeid lisada ja hallata.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
4	5	3	

### Nõue #12

<b>Tüüp</b>	12.1 (Kiiruse ja latentsuse nõuded)		
<b>Kirjeldus</b>	Tarkvara peab vähem kui tuhande nõude haldamisel kõiki toiminguid nõuete nimekirja haldusaknas teostama vähemalt viie sekundi jooksul.		
<b>Nõude vajalikkus</b>	Tarkvaraprojektides olevate nõuete arv tõenäoliselt jääb alla 500.		
<b>Sobivustingimus</b>	Filtreerimise, lisamise ja kustutamise toimingud võtavad vähem aega kui kirjelduses mainitud piirang.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
4	4	5	

### Nõue #19

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)		
<b>Kirjeldus</b>	Nõuete tüüpide algsel valimisel ja hiljem muutmisel peavad nad asuma puus ning puust peab olema neid võimalik otsida märksõna järgi.		
<b>Nõude vajalikkus</b>	Nõuete puu on üsnagi suur ja sealt on keeruline sirvides kõike vajalikku leida.		
<b>Sobivustingimus</b>	Võimalus otsida nõuete puust nõuet märksõnade järgi.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
4	5	3	

#### Nõue #16

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)		
<b>Kirjeldus</b>	Konkreetselt nõuet saab avada otse nõude numbriga järgi.		
<b>Nõude vajalikkus</b>	Oluline teadaoleva nõude kiirelt koodiga avamiseks.		
<b>Sobivustingimus</b>	Võimalus sisestada nõude number ja avada see konkreetne nõue ühe nupu vajutusega.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
2	3	4	

#### Nõue #13

<b>Tüüp</b>	12.6 (Nõuded mahtudele)		
<b>Kirjeldus</b>	Tarkvara projekti andmebaas peab suutma hoida vähemalt 1000 erinevat nõude kirjet.		
<b>Nõude vajalikkus</b>	1000 on piir mida ilmselt ületama ei hakata.		
<b>Sobivustingimus</b>	Tarkvara võimaldab sisestada ja hallata kuni 1000 nõuet		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
3	4	2	

#### Nõue #14

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)		
<b>Kirjeldus</b>	Valitud nõuete eksportimise võimalus <i>HTML</i> faili.		
<b>Nõude vajalikkus</b>	Selleks, et teha vajadusel programmivälist töötlust või lihtsalt nõudeid trükkida.		
<b>Sobivustingimus</b>	Võimalus algselt valitud nõuete eksporti <i>HTML</i> faili.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
3	4	2	

#### Nõue #15

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)		
<b>Kirjeldus</b>	Nõude tüüpe peab olema võimalik valida nii uue projekti loomisel kui hiljem nõudeid hallates.		
<b>Nõude vajalikkus</b>	<i>Volere</i> metoodikas kirjeldatakse palju erinevaid nõuete tüüpe ja on väga tõenäoline, et mingi kindla projekti tarbeks on vaja ainult alamhulka nendest. Vajadused võivad projekti kestel ka muutuda.		
<b>Sobivustingimus</b>	Uue projekti loomisel on võimalik märkida algselt soovitud nõuete tüübid. Valikuid on võimalus hiljem lisada ja muuta.		
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>	
4	3	4	

#### Nõue #17

<b>Tüüp</b>	4.1 (Piirangud lahendusele)	
<b>Kirjeldus</b>	Iga projekt peab asuma eraldi failis.	
<b>Nõude vajalikkus</b>	Võimaldab projekte programmiväliselt kustutada ja arhiveerida.	
<b>Sobivustingimus</b>	Iga uue projekti loomisel luuakse uus andmebaasi fail.	
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>
3	5	4

#### Nõue #18

<b>Tüüp</b>	9.1 (Funktsionaalsed nõuded)	
<b>Kirjeldus</b>	Nõudeid on võimalik grupiviisiliselt eemaldada, tehes valiku nõuete nimekirjas.	
<b>Nõude vajalikkus</b>	Kui on tarvis suuremat hulka teatud nõudeid kustutada.	
<b>Sobivustingimus</b>	Võimalus valida nõuete nimekirjas soovitud nõuded ja need ühe nupu vajutusega eemaldada.	
<b>Prioriteet</b>	<b>Kliendi rahulolu</b>	<b>Kliendi rahulolematus</b>
2	3	1

### 3. Tarkvaraprojekti kirjeldus

Projekti tulemusena valmis *Volere* metoodikal baseeruv tarkvara, mis võimaldab selle metoodikaga kogutud nõudeid hallata. Programm võimaldab teostada *Volere* metoodikast lähtuvalt põhilisi operatsioone: nõuete lisamine, otsimine, muutmine, filtreerimine. Tarkvarast on liigse mahu tõttu välja jäetud nõuete ajaloo säilitamine, mida *Volere* mall tegelikult kirjeldab. Programmi lisadena valmisid inglisekeelne kasutusjuhend ja demofail. Enne arendustöö alustamist tuli tutvuda sügavamalt *Volere* metoodikaga. Esmalt tuli selgeks teha *Volere* ametliku malli sisu ja põhimõtted, seejärel koostada nõuded loodavale tarkvarale ning tutvuda arendusvahendite ja programmeerimiskeelega. Nõuete kogumisel rakendati samuti *Volere* nõuete kogumise malli. Arendusprotsessis on lähtutud erinevatest agiilsetest tarkvaraarenduse võtetest. Kuna tegu on ühe inimese ja väikese projektiga, siis mingite jäikade arendusprotsesside järgimine mõtet ei omanud.

#### 3.1 Kasutatud tehnoloogiad

Uue väljakutsena oli soov õppida tundma *.NET* tehnoloogiat ja *C#* programmeerimiskeelt ning seetõttu sai tarkvara arendatud just selles keeles. Varasemad teadmised piirdusid *.NET* tehnoloogia üldisemate printsiipide tundmisega. Kuna varasem programmeerimiskogemus keeles *C#* puudus, siis oli selle tundma õppimine omamoodi väljakutse. Projekti algusfaasis on tuginetud erinevatele õppematerjalidele, mis tutvustasid *.NET* platvormi ja *C#* programmeerimiskeelt. Üheks kõige kasulikumaks raamatuks kujunes „**PRO C# 2008 AND THE NET 3.5 PLATFORM**“ [2]. Kuna arendusplatvormiks sai valitud *.NET*, siis selle kõige sobivamaks arenduskeskkonnaks osutus Microsoft Visual Studio, mida arendustöös ka kasutati.

Seoses *.NET* arendusplatvormi kasutamisega oli üheks eesmärgiks tutvuda ja rakendada *LINQ* üle *SQL*[3] tehnoloogiat, mis tänapäeval on uudne, huvitav ja võimas. Valminud tarkvaras salvestatakse iga loodav projekt arvuti kõvakettale eraldi andmebaasi failina. Andmebaasisüsteemiks valitud *Microsoft SQL Compact* versiooniga aga esines *LINQ* üle *SQL* tehnoloogiat kasutades mõningaid probleeme. Kuna *LINQ* üle *SQL* on suhteliselt uudne ja *Microsoft* ei ole *Visual Studio 2008*-le loonud korralikku võimalust andmebaasi failist *C#* klasside moodustamiseks, siis tuli selleks kasutada muid vahendeid. Üldjuhul andmebaasi *C#*

klasside loomine toimis hästi, kuid nende käsitsi muutmist ning meetodite lisamist tuli teha peale igat uut genereerimist.

### 3.2 Kasutatud raamistikud

Antud tarkvara loomisel on lisaks *.NET* standardteekidele kasutatud võimsat mitme dokumendi liidese (*MDI – Multiple Document Interface*) raamistikku. Arendamisel võeti kasutusele kõige uuem olemasolev *.NET* raamistiku versioon 3.5.

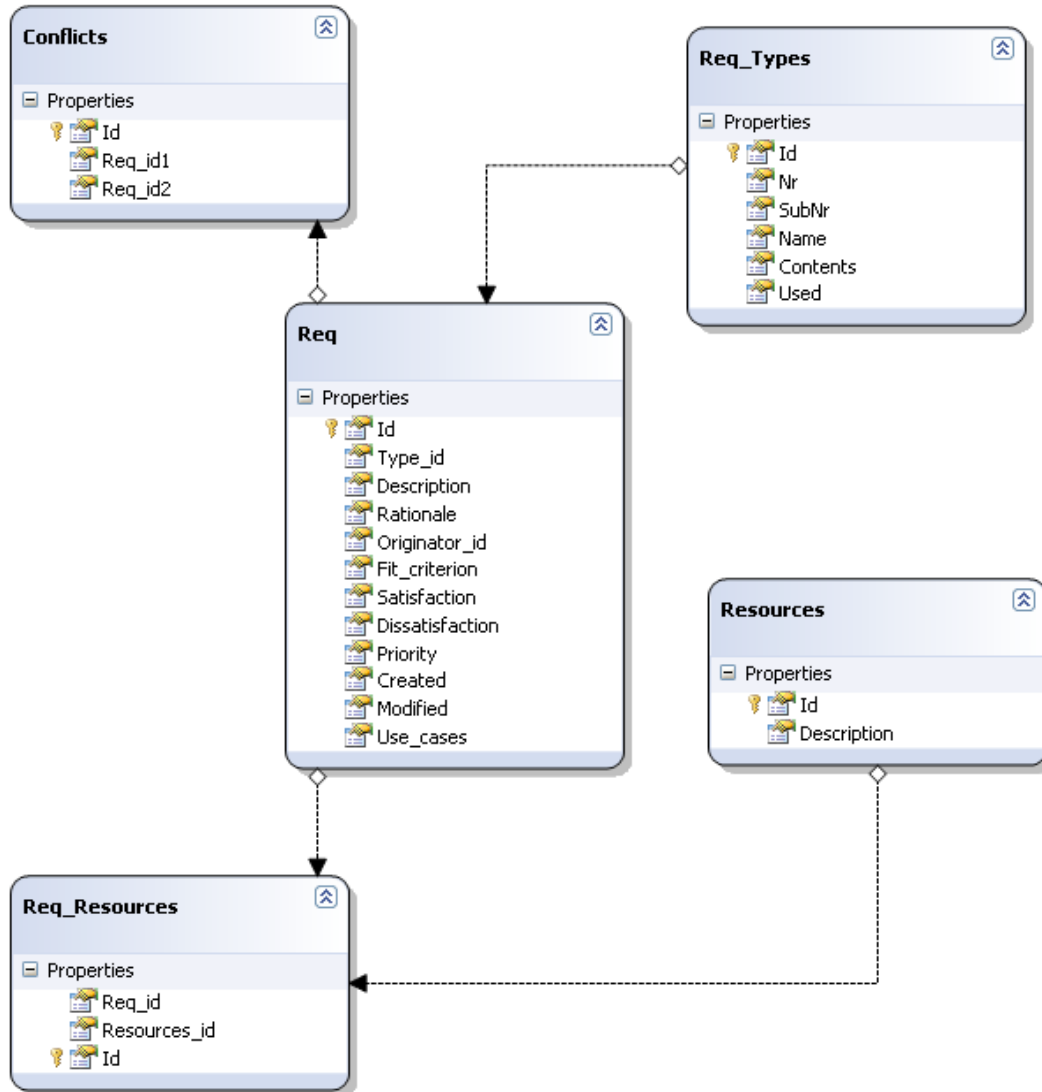
Programmis kasutatud *MDI* liidesena on kasutatud vabavaralist lahendust *Dock Panel Suite* 2.2 [4]. Selle raamistiku abil oli võimalik luua *Visual Studios* kasutatavale *MDI* liidesele sarnane keskkond. Raamistik aitas oluliselt kergemini luua kasutajasõbralikku ja produktiivset keskkonda nõuete haldamiseks. Sarnase raamistiku ise arendamine ei oleks käesoleva töö raames mahu tõttu mõeldav olnud.

### 3.3 Andmemudeli kirjeldus

Andmemudel antud tarkvara loomisel sisaldab järgmiseid objekte:

- Konflikt
- Nõude tüüp
- Nõue
- Ressurss

Programmis kasutatav andmemudel, mis seob eelpool toodud objekte, kajastub joonisel 1.



Joonis 1. Andmemudel

## 4. Loodud tarkvara kirjeldus

Loodud tarkvara vastab kõikidele peatükis 3. kirjeldatud nõuetele. Tarkvara loomisel on lähtutud aspektist, et lõpp-produkt peab olema kasutajale mugav ja tagatud peab olema kõik põhiline funktsionaalsus *Volere* metoodika rakendamiseks. See ülesanne ka täidetud sai. Võimalik on antud tarkvara abil nõudeid lisada, hallata, sorteerida, filtreerida ja otsida.

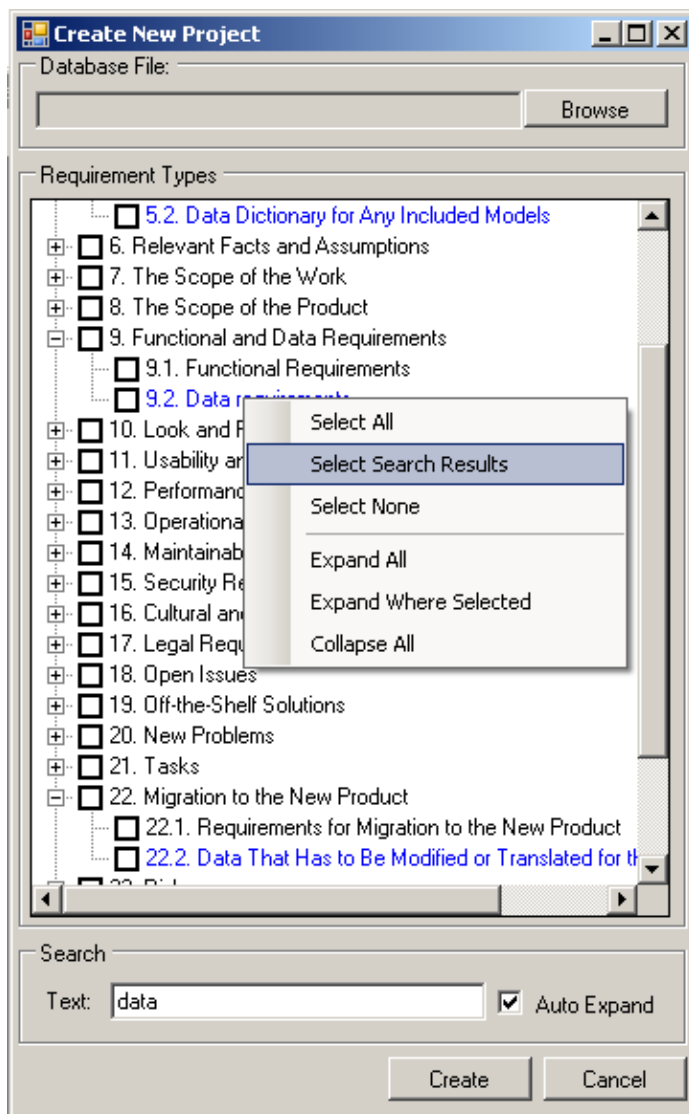
### 4.1. Ülevaade

Tarkvaras kasutatav nõuete kogumise kest võimaldab seda kasutada kui *Volere* nõude kesta. Üks oluline eesmärk oli rõhuda kasutusmugavusele ja sellega seoses sai kasutatud mõningaid täiesti uudseid lähenemisviise. Näitena võiks siinkohal välja tuua nõuete tüüpide puust nõuete otsimine nõude tüübi teksti järgi ja nende otsingutulemuste kiire selekteerimine. Lisaks on tarkvaras arvestatud sellega, et kasutajal võib olla tarvis töötada korraga terve grupi nõuetega. Selle lihtsustamiseks on loodud erinevate tingimuste järgi nimekirjade filtreerimine ning erinevad operatsioonid nõuete gruppidega: kõikide avatud nõuete salvestamine, sulgemine, kustutamine jms.

### 4.2. Kasutamine

Loodud tarkvaraga tööd alustades, tuleb esmalt luua uus projekti fail või avada olemasolev. Uue projekti loomisel tuleb kasutajal valida mingi alamhulk nõuete kategooriatest, mida ta esialgu vajalikuks peab. *Volere* nõuete puust on võimalik endale sobivad nõuete alam-kategooriad leida otsingut kasutades nagu on näha ekraanivormil 1.





**Ekraanivorm 1. Nõuete kategooriate valimine**

Peale uue projekti loomist või olemasoleva avamist kuvatakse kasutajale tööaken, kus vaiki-  
misi näidatakse nõuete haldurit (Requirements Manager) ja eelvaate akent (Preview). Avatud  
projekt on nähtav ekraanivormil 2.

Volere Manager - C:\Documents and Settings\All Users\Desktop\vm\_requirements.vdb

File Project View Help

Requirements Manager

Filter Add Remove Open #

#	Description	S	D	P	Type
8	Iga nõude kohta on võimalik märkida sellega konfliktis olev...	4	3	3	9.2.Data requirements
9	Iga nõude kohta peab saama kirjeldada sellega seotud res...	3	2	3	9.2.Data requirements
10	Nõuete aknad peavad avanema programmi põhiakna sees...	4	2	3	10.2.Style Requirements
11	Tarkvara peab olema võimalik probleemivabalt kasutada p...	5	3	4	11.3.Learning Requirements
12	Tarkvara peab <10000 nõude haldamisel kõiki toimimuid n...	4	5	4	12.1.Speed and latency requirements
16	Konkreetset nõuet saab avada otse nõude numbr järgi.	3	4	2	9.1.Functional Requirements
13	Tarkvara projekti andmebaas peab suutma hoida vähemalt...	4	2	3	12.6.Capacity requirements
14	Andmete eksportimise võimalus html'i.	4	2	3	9.1.Functional Requirements

Preview

Description

Iga nõude kohta peab saama kirjeldada sellega seotud ressursid tekstiliselt kujul.

#8 Iga nõu... #11 Tarkvar... #9 Iga nõu...

#8 Created: 2.05.2008 Requirement Type: 9.2.Data requirements Last Edited: 18.05.2008

Requirement

Description:

Iga nõude kohta on võimalik märkida sellega konfliktis olevad nõuded.

Rationale:

Nõuete kogumisel on võimalik, et mingis arendusfaasis on nõudeid, mis on üksteisega vastuolus. Need on vaja ära kirjeldada, et probleemi hiljem lahendada oleks võimalik.

Fit Criterion:

On võimalus lisada iga nõude kohta temaga konfliktis olevate nõuete numbreid.

Priority: 3

Customer Satisfaction: 4

Customer Dissatisfaction: 3

Conflicts:

Add Remove

Originator: Karl Blum Use cases:

Active Requirements:18

## Ekraanivorm 2. Programmi tööaken

Uue nõude loomisel kuvatakse kasutajale tühja nõude blankett, mis vastab *Volere* mallis kirjeldatud nõuete kestile. Kasutusmugavuse huvides on ressursside osa viidud blanketi teisele alamlehele (Resources Tab). Nõude blankett on nähtav ekraanivormil 3.

**New\***

#

Created:

Last Edited:

Requirement Type: 10.2 Style Requirements

Requirement

Resources

Description:

Rationale:

Fit Criterion:

Priority

Customer Satisfaction

Customer Dissatisfaction

Conflicts

Add

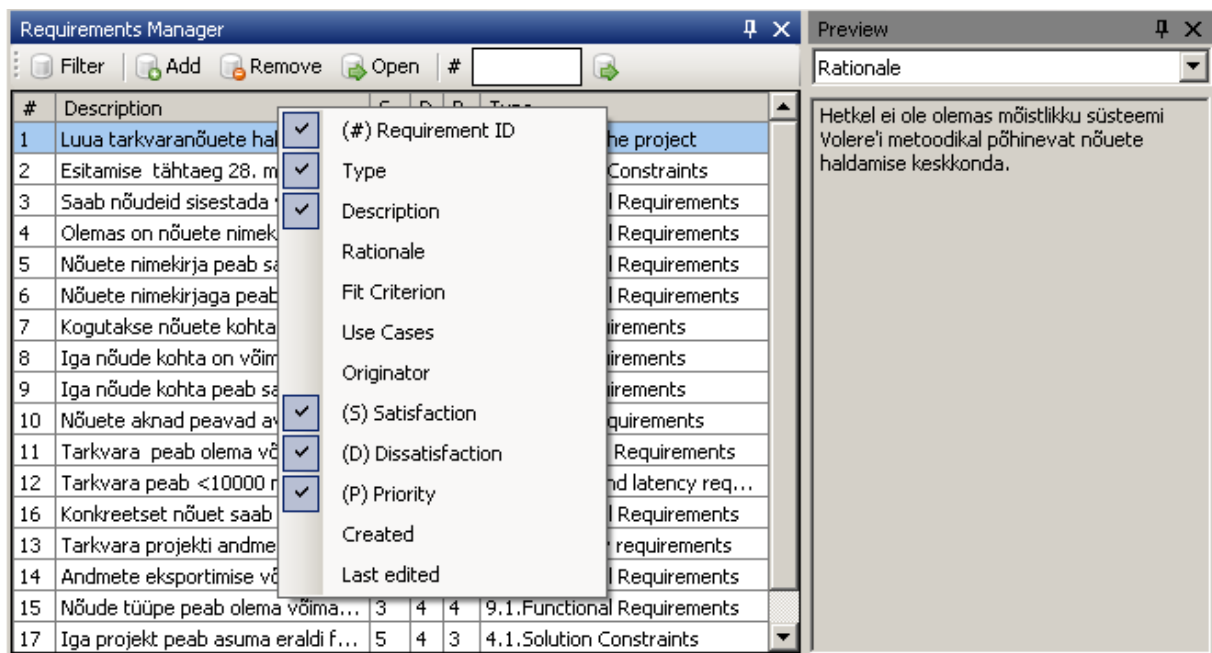
Remove

Originator:

Use cases:

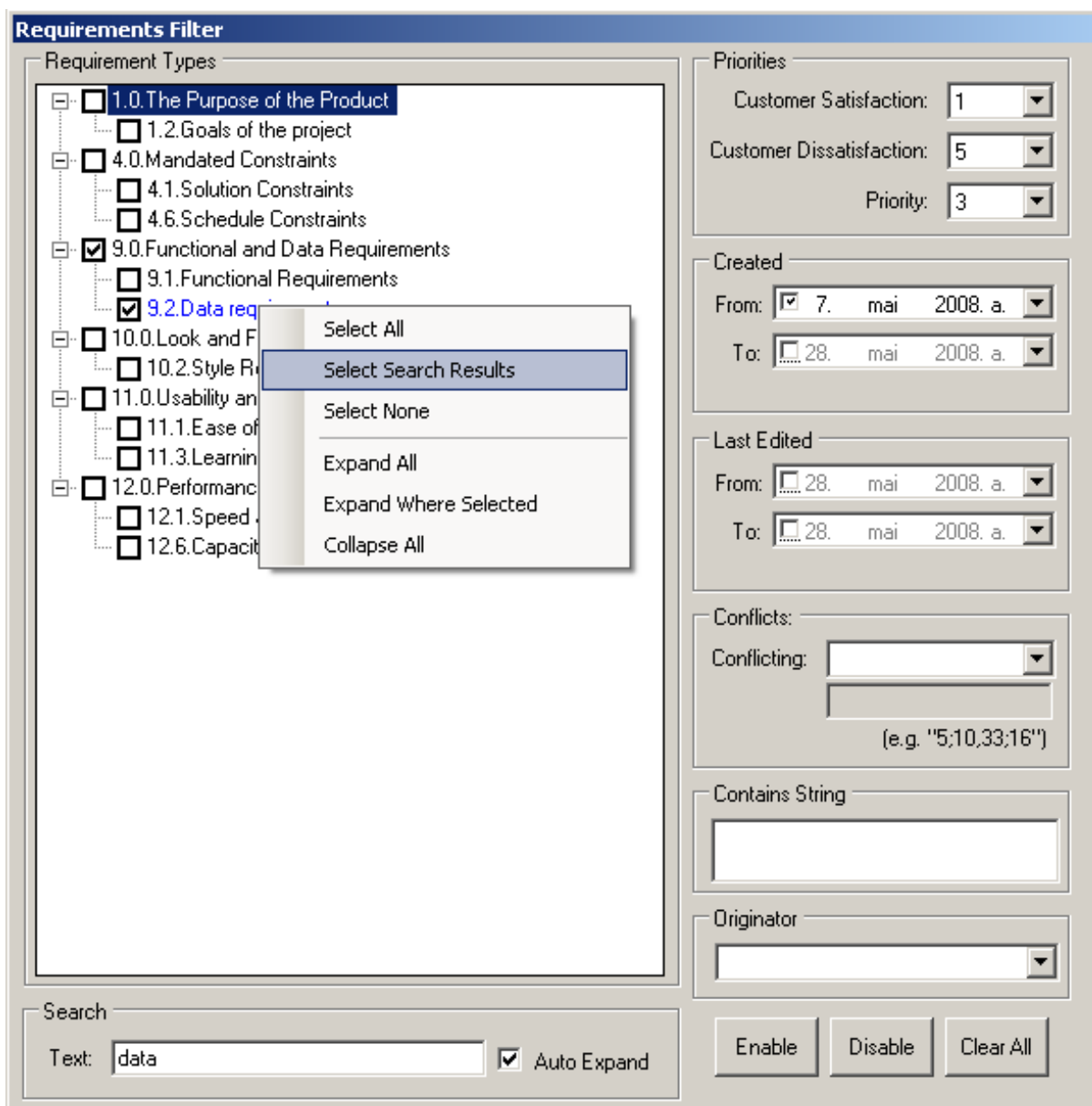
### Ekraanivorm 3. Uue nõude loomine

Uued loodud nõuded ilmuvad nõuete haldurisse nähtavale kohe peale nõude salvestamist. Nõuete halduris on näha nimekiri kasutaja valitud nõuete väljadega. Nõuete nimekirjas mugavaks sirvimiseks on loodud eelvaate aken, mis kuvab nimekirjast nõude valimisel koheselt sellega seotud sisu. Nõuete haldurit ja eelvaate akent näeme ekraanivormil 4.



**Ekraanivorm 4. Nõuete haldur ja eelvaate aken**

Nõuete haldur annab meile hea võimaluse nõudeid sirvida, kuid suuremate mahtude korral muutub nimekiri pikaks ja sellest vajaliku kirje otsimine oluliselt keerulisemaks. Nimekirjas sirvimise abistamiseks on loodud nõuete filtreerimise võimalus vastavalt kasutaja seadistatud parameetritele. Filtrit aktiveeritakse ja deaktiveeritakse nõuete haldurist. Nõuete filtrit näeme ekraanivormil 5.



Ekraanivorm 5. Nõuete filtreerimine

#### 4.3. Paigaldamine

Tarkvara paigaldamiseks on lisana kaasas CD installatsioonifailidega. Tarkvara paigaldamiseks on vajalik operatsioonisüsteem *Windows XP* või *Windows Vista*. Lisaks peab tarkvara käivitamiseks olema arvutisse paigaldatud *.NET 3.5*, mille olemasolu kontrollitakse installatsiooni käivitamisel ning vajadusel laetakse arvutisse automaatselt üle interneti. Peale installeerimist on võimalik käivitada programm *Windowsi* töölaualt. Demofaili avamiseks

CD-lt (demo/vm\_requirements.vdb), tuleb see eelnevalt kõvakettale kopeerida, kuna programm vajab failile kirjutamise õiguseid.

## Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli tutvuda *Volere* metoodikal baseeruvale nõuete haldamisega ja selle lihtsustamiseks luua tarkvara, mis vastaks *Volere* mallis kirjeldatud nõuetele ja soovitustele. Antud töö raames analüüsiti *Volere* põhimõtteid, tutvuti kõigi 27 *Volere* mallis kirjeldatud nõude kategooriaga ning programmeeriti vastava tarkvara. Tarkvara loomisel tutvuti esmalt *.NET* arendusplatvormi ning *Microsofti* uute arendusvahenditega. Bakalaureusetöö kestel omandati põhiteadmised *.NET* põhimõtetest ja *C#* programmeerimiskeelest. Tarkvara arendamisel on lähtutud üldistest agiilsetest arendusmeetoditest ja rangeid distsipliine kasutusele ei võetud. Arendusprotsessis esines mõningaid tehnoloogiast ja arendajast tingitud tagasilööke, kuid üldjuhul valitud vahendid, olemasolevad oskused ja õppimisvõime alt ei vedanud.

Valminud tarkvaras on realiseeritud kõik dokumenteeritud nõuded, ning valminud tarkvara saab täiesti kasutatavaks lugeda. Arvestades bakalaureusetöö mahtu ei olnud võimalik lisada mõningat funktsionaalsust, mis tõenäoliselt tõstaks omakorda produktiivsust selle tarkvara kasutamisel. Tähtis on aga see, et sai täidetud eesmärk luua ainulaadne ja mugav *Volere* metoodikal põhinev tarkvaranõuete halduskeskkond.

## Kasutatud kirjandus ja materjalid

- [1] James & Suzanne Robertson „**VOLERE REQUIREMENTS SPECIFICATION TEMPLATE**“ (2006)
- [2] Andrew Troelsen „**PRO C# 2008 AND THE NET 3.5 PLATFORM**“ (2008)
- [3] D. Kulkarni, L. Bolognese, M. Warren, A. Hejlsberg, K. George „**LINQ TO SQL: .NET LANGUAGE-INTEGRATED QUERY FOR RELATIONAL DATA**“ (2007)  
<http://msdn.microsoft.com/en-us/library/bb425822.aspx>
- [4] Weifen Luo „**DOCK PANEL SUITE**“ <http://sourceforge.NET/projects/dockpanelsuite/>
- [5] „**EXPERIENCES OF VOLERE USERS**“ <http://www.volere.co.uk/experience.htm>
- [6] „**REQUIREMENTS TOOLS**“ <http://www.volere.co.uk/tools.htm> (2007)

*Kõik viited viimati väisatud 16.05.2008*



# Requirements Managment Software For Volere Method

## Bachelor's Thesis

Karl Blum

## Resume

Herein baccalaureate thesis had two main goals. First was to familiarize with software requirements management method *Volere*. The second goal was to develop software for managing software requirements if *Volere* method is used. The developed software is probably one of a kind – convenient, easy to use and practical for managing requirements collected by *Volere* method. This software's functionality is not overwhelmed and provides the main functions (searching, adding, removing, modifying) as easy as possible.

At first the *Volere* template consisting of 27 different categories of requirements was analyzed. Within the development phase new technologies such as: *.NET* and *C#* were learnt. The software development followed some agile techniques methods and did not use any strict rules. The developed software meets all the documented requirements and it can be used to manage software requirements when *Volere* is used. This software includes some neat features to give the user convenient ways to manage requirements.

## Lisad

### Lisa 1. Programmi lähtekood ja installatsioonifail CD-l

Käesoleva tööga kaasas oleval CD-l on valminud tarkvara installatsioonifail ning lähtekood. Kaasasoleva plaadi sisu struktuur on kirjeldatud tabelis 1.

Tabel 1. CD sisu

Kataloog CD'l	Sisu kirjeldus
<b>Setup/</b>	Tarkvara paigaldusfail.
<b>Source/</b>	Tarkvara lähtekood ja <i>Visual Studio 2008</i> projekti failid.
<b>Volere/</b>	Sisaldab <i>Volere</i> nõuete malli.
<b>Demo/</b>	Nõuete haldamise tarkvara näidisprojekt (vm_requirements.vdb).