

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Rain Varblane 192861IADB

AHP meetodi kasutamine PIM tarkvara valikul

Bakalaureusetöö

Juhendaja: Jaanus Pöial
PhD

Tallinn 2023

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Rain Varblane

15.05.2023

Annotatsioon

Käesolev bakalaureusetöö käsitleb Thomas L. Saaty analüütiliste hierarhiate meetodi (AHP) rakendamise sobilikkust ettevõttes. Töö eesmärgiks on analüüsida AHP meetodi sobivust tarkvara võrdlemiseks ettevõtte olustikus konkreetse juhtumi näitel ning seeläbi leida, kas Saaty meetod sobiks töös käsitletud tarkvaraarendusega tegelevasse ettevõttesse lahendamaks töö käigus tekkivaid otsustusprobleeme.

Töös antakse esmalt ülevaade kasutatud tarkvarast ning uurimus- ja otsustusmeetoditest ning viiakse läbi juhtumianalüüs, mille käigus vaadeldakse tooteinfosüsteemide võrdlemist AHP meetodiga. Vaatluse tulemusi valideeritakse, tehes võrdlemisel valituks osutunud Dynamicweb tarkvarast prototüüp.

Põhinedes vaatluse ja prototüüpimise tulemustele ning intervjuude käigus kogutud tagasisidele kinnitati, et analüütiline hierarhiate meetod sobib ettevõtte olustikus tarkvara võrdlemiseks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 20 leheküljel, 6 peatükki, 8 joonist, 3 tabelit.

Abstract

PIM Software Selection Using the AHP Method

This bachelor thesis deals with the appropriateness of applying Thomas L. Saaty's analytic hierarchy process (AHP) method in a company environment. The aim of the thesis is to analyse the suitability of the AHP method for comparing software in a business environment by using a specific case study as an example, and thus find out whether Saaty's method would be suitable for the software development company in order to solve decision problems that arise in the course of the work.

The paper gives an overview of the software used and the employed research and decision methods. A case study is conducted in which the comparison of product information systems with the AHP method is observed. The results of the observation are validated by creating a prototype of the chosen Dynamicweb software.

Based on the results of the observation and the prototyping, and the feedback collected during the interviews, it was confirmed that the analytical hierarchy process method is suitable for comparing software in a company environment.

The thesis is in Estonian and contains 20 pages of text, 6 chapters, 8 figures, 3 tables.

Lühendite ja mõistete sõnastik

AHP	Analüütiliste hierarhiate meetod, <i>Analytic hierarchy process</i>
CLI	Käsurea kasutajaliides, <i>Command-line interface</i>
ERP	Ettevõtte ressursside planeerimine, <i>Enterprise Resource Planning</i>
IDE	Integreeritud arenduskeskkond, <i>Integrated Development Environment</i>
IIS	<i>Internet Information Services</i> veebiserveri tarkvara
NuGet	Microsofti pakihaldussüsteem
OSL V3	Open Software License Version 3
PaaS	Pilveteenuse mudel, <i>Platform as a Service</i>
PIM	Tooteinfo haldamine, <i>Product Information Management</i>
SaaS	Tarkvara tarneviis, <i>Software as a Service</i>
SDK	Tarkvaraarenduskomplekt, <i>Software Development Kit</i>

Sisukord

Autorideklaratsioon	2
Annotatsioon.....	3
Abstract.....	4
Lühendite ja mõistete sõnastik	5
Sisukord.....	6
Jooniste loetelu	8
Tabelite loetelu	9
Sissejuhatus	10
1 Teoreetilised alused	12
1.1 Töös kasutatavad meetodikad	12
1.1.1 Juhtumianalüüs	12
1.1.2 AHP meetod	13
1.2 Tarkvara ja tööriistad.....	14
1.2.1 Tooteinfo haldamise süsteem	15
1.2.2 Akeneo PIM Community Edition.....	15
1.2.3 Dynamicweb PIM.....	15
1.2.4 Priority Estimation Tool - PriEsT.....	16
2 AHP Analüüs.....	17
1.1 Analüüsi sisend.....	17
2.1.1 Arenduspartneri sisend hindamisele	17
2.1.2 Nõuded tarkvarale	18
2.2 AHP analüüsi eeltöö	18
2.2.1 Eesmärk ja kriteeriumite defineerimine	18
2.2.2 PIM tarkvara alternatiivide eelvalik	19
2.3 Võrdlemine AHP meetodiga	20
2.3.1 Kriteeriumite võrdlemine	20
2.3.2 Alternatiivide võrdlemine.....	22
3 Tarkvara prototüüp	23

3.1 Laiendatavuse võimalused.....	23
3.2 Prototüübi disain.....	24
3.3 Prototüübi loomine	25
3.3.1 Lisaarendus.....	25
3.3.2 Kohandamine läbi kasutajaliidese	26
4 Järeldused	27
4.1.1 Prototüüp	27
4.1.2 AHP meetoodika	27
4.1.3 Eksperiment.....	28
5 Kokkuvõte	29
Kasutatud kirjandus	30
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	32
Lisa 2 – AHP võrdlus	33
Lisa 3 – Prototüüp	37

Jooniste loetelu

Joonis 1. Näide hierarhiselt struktureeritud AHP otsustusprobleemist [12]	13
Joonis 2. Akeneo PIM CE kasutatavad tehnoloogiad [17]	15
Joonis 3. Dynamicweb 10 tarkvarakomplektis olevad moodulid	16
Joonis 4. Hierarhiselt struktureeritud otsustusprobleem	20
Joonis 5. Hinnangud kriteeriumitele	21
Joonis 6. Kriteeriumite võrdlemise graaf	21
Joonis 7. Hinnangud alternatiividele	22
Joonis 8. <i>Publisher-Subscriber</i> muster [22]	23

Tabelite loetelu

Tabel 1. Saaty absoluutarvude skaala põhiskaala [14].....	14
Tabel 2. Määratletud tarkvarakriteeriumid.....	19
Tabel 3. Tarkvara alternatiivid	19

Sissejuhatus

Valikute langetamine, täpsemalt tarkvara valimine, on tänapäeva ärimaailmas oluline, kuid samas keeruline tegevus. Õige tarkvara võib aidata ettevõttel tõhustada oma protsesse, suurendada tootlikkust või parandada klientide rahulolu.[1][2] Samuti võib tarkvara valikul olla suur mõju ettevõtte kuludele, nii lühikeses kui ka pikas perspektiivis.[2] Tarkvara võrdlemine hõlmab erinevaid aspekte, nagu näiteks funktsionaalsus, hind, kasutusmugavus, tugi ja hooldus jpm.[3]

Loodud on erinevaid otsustusmudeleid[4][5], raamistike[6], meetodikaid[7] ning tarkvara[8][9] toetamiseks otsuste tegemist. Siiski võib olla keeruline leida õiget vahendit, mis aitaks langetada otsuseid tarkvara valikul. Kuigi ei ole olemas ühte õiget lahendust, mis kataks kõiki olukordi, on käesoleva töö ajendiks leida lihtsalt kasutatav tööriist, mis valikute tegemisel aitaks.

Teema kasvas välja e-kaubanduse tarkvaraarendusega tegelevast ettevõttest, kus puudus süsteem või raamistik, mille alusel hinnata potentsiaalse tarkvara sobilikkuse tehnilisi aspekte. Probleemi aktuaalsus kasvas ka seetõttu, et ettevõtte soovis arendajaid enam kaasata müügi- ja analüüsiprotsessi.

Käesolevas töös käsitletakse ühe potentsiaalse lahendusena analüütiliste hierarhiate meetodit ehk Saaty meetodit[7]. Valikut kindlustas tõsiasi, et AHP meetodi edukat kasutust on kajastatud ka mitmetes teadustöodes[10][11][12][13].

Lisaks, töid Seyhan Sipahi ja Mehpare Timor oma 2010 aasta artiklis[10] välja valdkondadena, kus AHP meetodit on rakendatud, näiteks tootmistööstuse ja tervishoiu ja ka IT valdkonda. Määravaks said ka meetodika lihtsus, paindlikkus ning probleemide kajastuv subjektiivsus.

Eesmärk on testida AHP meetodi sobivust tarkvara võrdlemiseks ettevõtte olustikus ning seeläbi leida, kas Saaty meetod sobiks tarkvaraarendusega tegelevasse ettevõttesse lahendamaks otsustusprobleeme, mis tekivad tarkvaraarendajate igapäevatöö käigus.

Töö koosneb neljast sisulisest peatükist. Teooria osas antakse ülevaade kasutatud tarkvarast ning uurimus- ja otsustusmeetoditest. Teises peatükis vaadeldakse tooteinfosüsteemide võrdlemist AHP meetodiga. Kolmandas peatükis luuakse valituks osutunud tarkvarast prototüüp. Neljandas, järelduste peatükis, tehakse tagasiwaade tehtud tööle, analüüsitakse tulemusi, võetakse kokku tagasiside ning antakse eksperimendile lõplik hinnang.

1 Teoreetilised alused

Teooria peatükis käsitletakse ülesandepüstitust ja töö eesmärki ning tutvustatakse edasist tegevuskava. Lisaks antakse ülevaade töös käsitletud metoodikatest, uurimus- ja otsustusmeetoditest ning kasutatud tarkvarast.

1.1 Töös kasutatavad metoodikad

Lisaks eelnevalt mainitud metoodikatest, uurimus- ja otsustusmeetoditest ülevaate andmisele, käsitletakse juhtumianalüüsi peatükis ka ülesandepüstitust, töö eesmärki ning tegevuskava. Samuti antakse ülevaade AHP meetodist, üldisest teooriast ning kuidas seda antud töö käigus kasutatakse.

1.1.1 Juhtumianalüüs

Käesoleva bakalaureusetöö eesmärk oli läbi viia kvalitatiivne juhtumianalüüsi eksperiment leidmaks kinnitust, kas AHP meetod on sobiv kasutamiseks ettevõtte olustikus.

Töös keskenduti e-kaubanduse tarkvaraarendusega tegeleva ettevõtte ühele tiimile, kus üks klientidest vajab oma e-kaubanduse platvormile uut tooteinfo haldamise süsteemi (PIM) [13]. Ettevõtte soovib arendajaid rohkem kaasata müügi- ja analüüsiprotsessis, kuid samas puudus ühtne metoodika või süsteem tarkvara võrdlemiseks ja sobivuse hindamiseks.

Juhtumianalüüsi käigus kasutati kolme uurimismeetodit:

1. Intervjuu
2. Vaatlus
3. Prototüüpimine

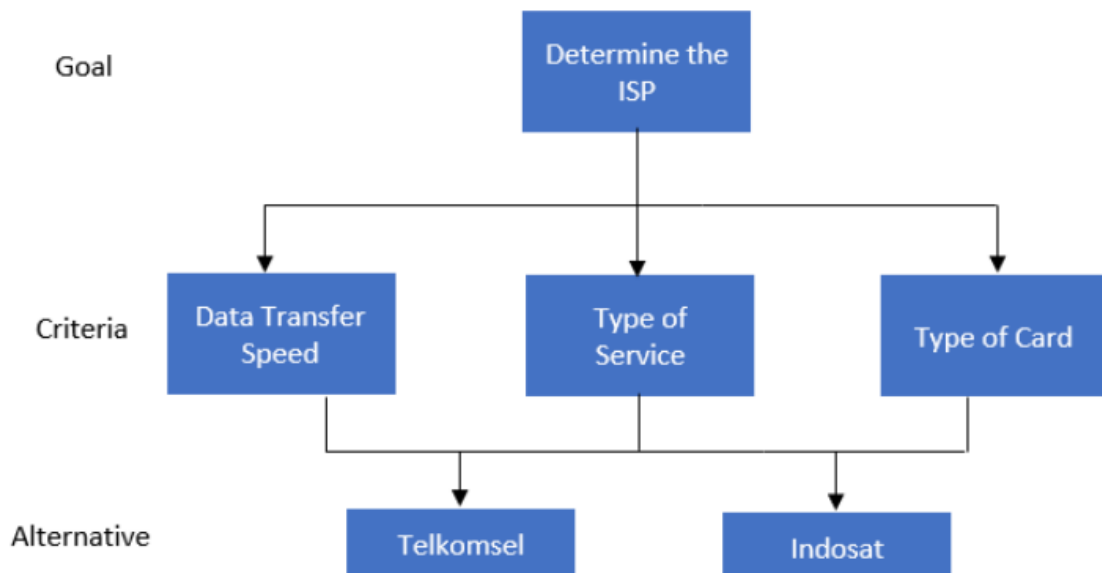
Esmalt määratleti intervjuu käigus lähtetingimused PIM tarkvara võrdluse läbiviimisele. Seejärel vaadeldi AHP meetodil PIM tarkvara võrdlemist, kus töö autoril oli võrdlemise läbiviija roll.

Eksperimendi toetamiseks kasutati valituks osutunud tarkvara prototüüpimist kinnitamaks võrdlemise edukust ja AHP meetodi sobivust. Läbivalt, töö käigus, toimusid tagasiside saamise eesmärgil intervjuud, kus osalesid ettevõtte arendajad ning tiimi- ja projektijuht.

1.1.2 AHP meetod

AHP meetodi peamine kasutusviis on mitme kriteeriumiga otsustusprobleemide lahendamine.[7] 1970-ndatel aastatel arendas Thomas L. Saaty analüütiliste hierarhiate meetodi (AHP), mis jaotab otsustusprobleemi hierarhiliselt kui eesmärk, kriteeriumid, alamkriteeriumid ja alternatiivid.[10]

Teisiti võib hierarhia elemente kirjeldada ka kui: ülesande lühikirjeldus (eesmärk), objekti omadused (kriteeriumid) ja objektid, mille vahel langetatakse otsus (alternatiivid).[11]



Joonis 1. Näide hierarhiliselt struktureeritud AHP otsustusprobleemist [12]

Tänu meetodika lihtsusele ja paindlikkusele, leiab Saaty meetod kasutust nii tootmis, ehitus- kui ka energiavaldkonnas ning mujalgi.[10] Meetodika valikul saidki määravaks meetodika lihtsus, paindlikkus ning probleemide kajastav subjektiiivsus.

Hierarhiliselt struktureeritud elemente võrreldakse paarikaupa ülema klasteri suhtes. Võrreldavaid elemente hinnatakse kasutades Saaty absoluutarvude põhiskaalat[14], mis on kujutatud järgnevas tabelis.

Tabel 1. Saaty absoluutarvude skaala põhiskaala [14]

Numbriline väärtus	Verbaalne skaala	Selgitus
1	Võrdväärset elementide	Elementide on võrdsed
2	Vahepealne väärtus	Üks kriteerium on teisest minimaalselt eelistatavam
3	Mõõdukas paremus	Üks kriteerium on teisest eelistatavam
4	Vahepealne väärtus	Kompromiss naaberhinnangute vahel
5	Oluline paremus	Üks kriteerium on teisest tugevalt eelistatud
6	Vahepealne väärtus	Kompromiss naaberhinnangute vahel
7	Väga tugev paremus	Kinnitatud praktikas, et üks kriteerium on teisest tugevalt eelistatud
8	Vahepealne väärtus	Kompromiss naaberhinnangute vahel
9	Ekstreemne paremus	Eelistatud kriteerium

Võrdluse teostamiseks, ebakõlade kontrollimiseks ja eelistuste tuletamiseks kasutati Priority Estimation Tool (PriEsT) tarkvara[15].

1.2 Tarkvara ja tööriistad

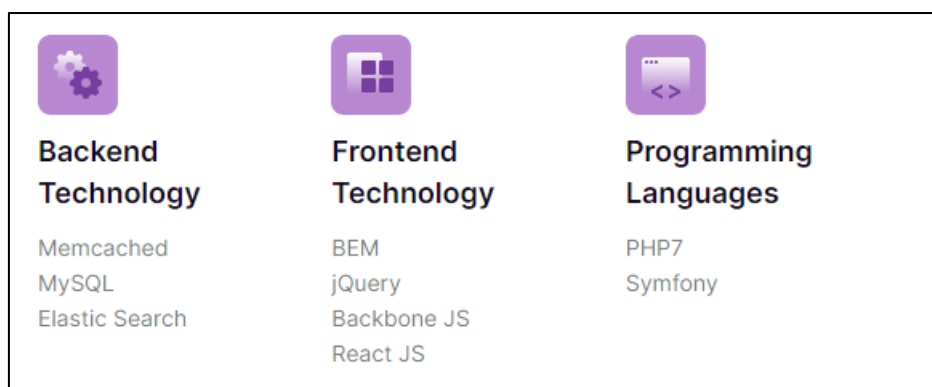
Järgnevalt antakse ülevaade töös käsitletud tarkvarast ja tööriistadest. Käsitletakse analüüsi käigus kasutatud AHP meetodil baseeruvat tarkvara ning antakse ülevaade tooteinfo haldamise süsteemist. Lisaks tutvustatakse ka hindamise käigus käsitletud tooteinfo haldamise süsteemi alternatiive.

1.2.1 Tooteinfo haldamise süsteem

Tooteinfo haldamise (PIM) süsteem on ärirakendus, mis pakub ühtset kohta tooteinfo kogumiseks, haldamiseks ja rikastamiseks, tootekataloogi loomiseks ning selle levitamiseks müügi- ja e-kaubanduse kanalitele. [13]

1.2.2 Akeneo PIM Community Edition

Akeneo PIM CE (Community Edition)[17] on avatud lähtekoodiga[16], OSL V3 litsentsiga tooteinfo haldamise süsteem, mis on ehitatud kasutades Symfony veebirakenduse raamistikku ja php programmeerimiskeelt. CE versioonil puuduvad *SaaS* ja *PaaS* kasutusmudelid ning rakenduse majutamise eest peab hoolitsema kasutaja.



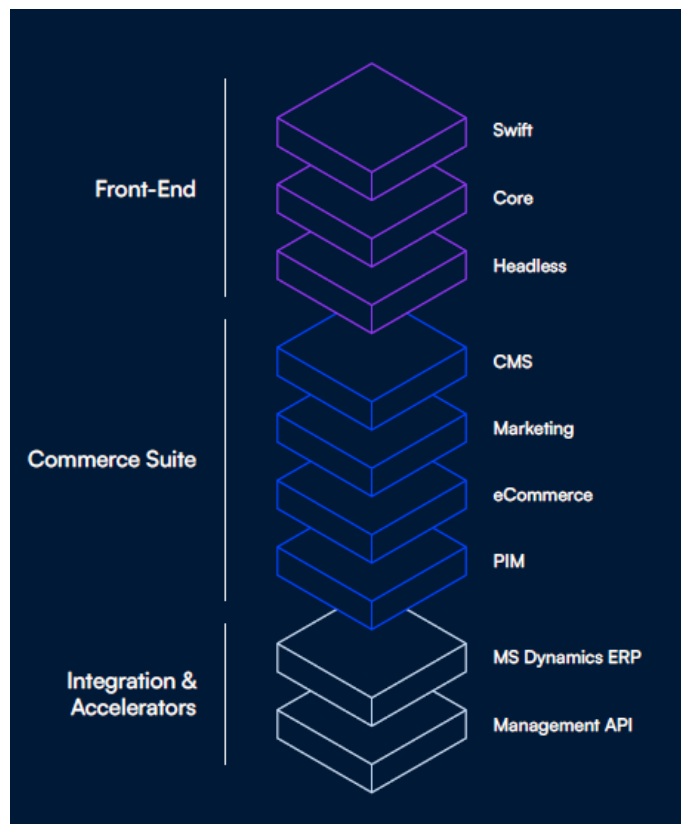
Joonis 2. Akeneo PIM CE kasutatavad tehnoloogiad [17]

Akeneo PIM võimaldab ühtset kohta tooteinfo kogumiseks, haldamiseks ja rikastamiseks. Standardsete funktsioonide hulka kuuluvad paindlik andmemudel, tooteimport ja -eksport, versioonikontroll, andmekontroll ja töövoogude haldamine. Tarkvara on eelkõige mõeldud väikestele ja keskmise suurusega ettevõtetele.

1.2.3 Dynamicweb PIM

Dynamicweb 10 Composable Commerce Suite on komposiitne MACH (*Microservices, API-first, Cloud, Headless*) arhitektuuril baseeruv e-kaubanduse tarkvarakomplekt, mis on ehitatud kasutades .NET tehnoloogiat.[18] Käesoleva töö kirjutamise hetkel ei ole Dynamicweb 10 täisversioon veel saadaval ning avalikult saadaval olev ametlik versioon on RC2 (*Release Candidate 2*), mis on mõeldud testimiseks ja tagasiside kogumiseks enne ametlikku väljalaset.

Erinevaid mikroteenuseid on võimalik komplekteerida ja arendada vastavalt kasutaja vajadustele. Tarkvarakomplektis olevad moodulid on kujutatud joonisel 3.



Joonis 3. Dynamicweb 10 tarkvarakomplektis olevad moodulid

Käesolevas töös käsitletakse PIM ja Core mooduleid. *Core* moodul tähendab serveri poolset vaadete renderdamist kasutades Razor märgendussüntaksit ja Dynamicweb vaadete malle.[18]

Dynamicweb 10 rakenduse arendamise eeldusteks on Microsoft SQL Server 2016 andmebaas, Microsoft IIS 10 veebiserver ja .NET 7 SDK.[19]

1.2.4 Priority Estimation Tool - PriEsT

PriEsT on avatud lähtekoodiga, tasuta saadaval olev ning AHP meetodil baseeruv otustuse langetamise tarkvara[15], mida kasutatakse käesolevas töös AHP analüüsi läbi viimiseks. Erinevalt alternatiividest nagu ExpertChoice ja HIPRE, võimaldab PriEsT visualiseerida ebakõlasid otsuste tegemisel.[15]

2 AHP Analüüs

Analüüsi käigus leiti AHP meetodit kasutades kliendile sobivaim tooteinfo haldamise süsteem. Kliendiks on Eestis tegutsev ehituskaupade müügiga tegelev ettevõtte. Arenduspartneriks on Eestis e-kaubanduse tarkvaraarendusega tegelev ettevõtte. Valituks osutunud tarkvarast tehti demokeskkond ehk prototüüp, mida kasutati edasises analüüsis ja järelduste tegemisel.

AHP analüüsi ehk hindamise teostas arenduspartneri poolt määratud kolmeliikmeline tiim ehk hindajad. Hindamise läbiviijaks oli töö autor. Hindajad lähtusid hindamisel kliendi- ja arenduspartneri poolsest sisendist. Analüüsi läbiviija tutvustas hindajatele metoodikat, vaatles analüüsi tööprotsessi, esitas suunavaid küsimusi ning dokumenteeris tehtud tööd.

1.1 Analüüsi sisend

Kliendipoolse sisendi edastas osalejatele projektijuht ja arenduspartneri poolsed nõuded edastas tiimijuht. Lisaks tarkvaranõuetele määrati piirangud ka hindamise läbiviimisele, eesmärgiga võimalikult täpselt simuleerida tavapärasest tööolukorda.

2.1.1 Arenduspartneri sisend hindamisele

Kuna eesmärk oli hinnata metoodika sobivust ettevõtte tingimustes, siis hindamise lähtetingimused olid sätestatud võimalikult sarnaselt tavaolukorrale.

1. Osalejate arv – Kuni kolm liiget.
2. Ajakulu – Alternatiivide kohta info otsimine, analüüs ja hindamine võib kokku võtta kuni kaks tööpäeva st 16 tundi.
3. Väljund – Hindamise väljundiks on valituks osutunud tarkvara. Prototüübi loomisele kuluvat aega ei arvestata enam ülal sätestatud ajapiirangu sisse.

2.1.2 Nõuded tarkvarale

Arenduspartneri poolt oli välistatud tarkvara, kus rakendust sai modifitseerida ainult läbi kasutajaliidese. Rakendust pidi saama muuta ja täiendada läbi tarkvaraarenduste. Võimalikult palju kliendi poolt soovitud funktsioone pidid olema süsteemi baasversiooni poolt kaetud.

Kliendi sooviks oli välja vahetada vana ja aegunud PIM lahendus, mis kasutas aegunud tehnoloogiaid ning ei katnud piisaval määral kliendi vajadusi, uue ja kaasaegsema vastu. Kliendi nõue oli, et tarkvara kataks nende poolt sätestatud funktsionaalsed ja mittefunktsionaalsed nõuded.

2.2 AHP analüüsi eeltöö

Esmalt, enne võrdlemisega alustamist, tuli luua probleemist hierarhiline mudel. Analüüsis osalejad pidid enda jaoks lahti mõtestama analüüsi eesmärgi, kriteeriumid ja alternatiivid.

2.2.1 Eesmärk ja kriteeriumite defineerimine

Eesmärgiks oli tervikliku ülesande lühikirjeldus.[20] Antud töö kontekstis on eesmärk: „Leida kliendile sobivaim tooteinfo haldamise süsteem“.

Kriteeriumite defineerimise käigus tuli vaadelda eesmärki, tarkvara ja tooteinfo haldamise süsteemi abstraktsemalt. Defineerimise käigus piiritleti eesmärgi erinevad aspektid. Antud töö kontekstis tuli määratleda need tarkvara tahud, mille alusel hinnati tarkvara sobivust.

Hindajate soov oli kriteeriumeid defineerida võimalikult abstraktselt, et samu kriteeriumeid oleks võimalik taaskasutada. Kriteeriumite valikul tekkinud arutelu muutus ajakulukaks, mis on kohati põhjendatav ka meetodi esmakordse kasutamisega. Kriteeriumite määratlemisele mõeldud koosolekud kestsid kokku u. 6 tundi. Analüüsi läbiviija soovitusel võeti aluseks tarkvara kvaliteedi aspektid ISO 9126[21] standardis kirjeldatud tarkvara kvaliteedi mudelist. Mudelit täiendati koosolekute käigus vastavalt hindajate konsensusele. Lisaks täiendati kriteeriumite loetelu lähtudes hindajate varasemast kogemusest sarnaste ülesannete läbiviimisel. Järgnevas tabelis (Tabel 2) on välja toodud lõplikult määratletud kriteeriumid koos hindajate selgitusega.

Tabel 2. Määratletud tarkvarakriteeriumid

Kriteerium	Selgitus
Funktsionaalsus	Tarkvara olemasolev funktsionaalsus. Kui hästi täidab tarkvara <i>out-of-the-box</i> kliendi vajadusi ja potentsiaalseid tulevikusoove?
Turvalisus	Teadaolevad vead ja riskid, potentsiaalsed ohud, riskide maandamise keerukus.
Kohandatavus	Rakenduse modifitseerimine. Olemasoleva funktsionaalsuse muutmine, visuaalse disaini muutmine, lisafunktsionaalsuse lisamise keerukus.
Jätkusuutlikkus	Turuseis, populaarsus, kogukond, dokumentatsioon, uuendused.
Hind	Rakendusega seonduv kulu: litsentsid, valmis pistikprogrammide (<i>plugin</i>) ostmine, veebimajutus, tehniline tugi jms.

2.2.2 PIM tarkvara alternatiivide eelvalik

Hindajatel tuli eeltöö käigus leida alternatiivid s.t potentsiaalselt sobivad PIM rakendused, mida lõpuks võrdlema hakati.

Järgnevas tabelis (Tabel 3) on hindajate poolt välja valitud alternatiivid koos selgitustega.

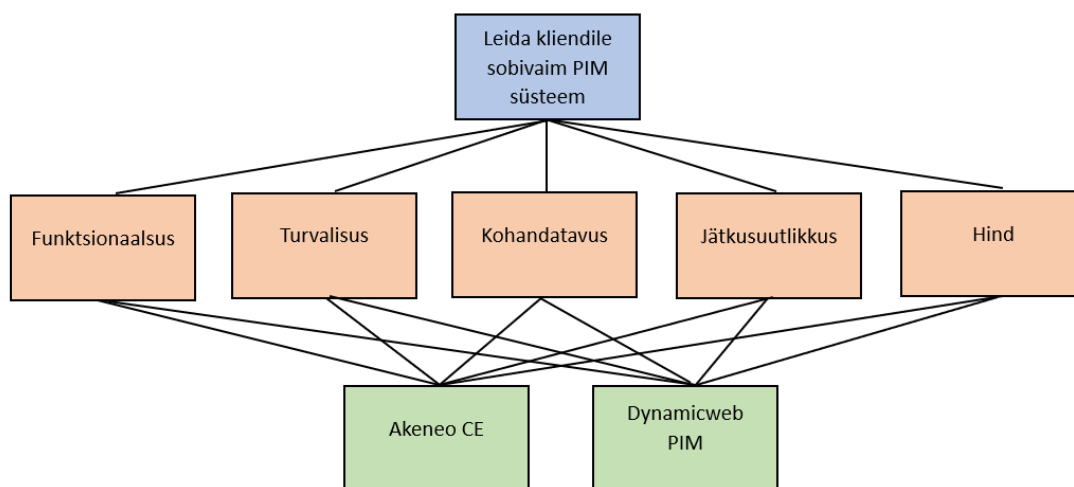
Tabel 3. Tarkvara alternatiivid

Alternatiiv	Selgitus
Akeneo PIMCommunity Edition (CE)	Üks turuliidreid nii välismaal, kui ka Eestis. Arenduspartneri analüütikute sõnul kohtavad nad potentsiaalsete klientide puhul just enim Akeneo PIM tarkvara. Akeneo on loodud kasutades php programmeerimiskeelt ja Symfony raamistikku.
Dynamicweb PIM	Dynamicweb PIM valiku puhul said määravaks Dynamicweb ökosüsteem, tootega kaasasolev esialgne funktsionaalsus ning kohandatavuse lihtsus. Dynamicweb PIM on loodud kasutades C# programmeerimiskeelt ja ASP.Net Core 7 raamistikku.

Akeneo PIM valiti kui üks kohalikke turuliidreid. Lisaks on tegemist avatud lähtekoodiga tasuta tarkvaraga. Dynamicweb PIM valiti esindama kinnise lähtekoodiga tasuta alternatiivi. Esialgu oli kolmanda alternatiivina arutelu all ka täiesti uue rakenduse ehitamine, kuid seda hinnati liiga ressursikulukaks ning eemaldati valikust.

2.3 Võrdlemine AHP meetodiga

Ülesanne on hierarhiliselt jaotatud ning defineeritud on eesmärk, kriteeriumid ja alternatiivid nagu on kujutatud alloleval joonisel. See omakorda võimaldas meil jätkata hierarhiliselt samal tasemel olevate elementide võrdlemist. Kriteeriumite ja alternatiivide võrdlemisele kulunud aeg oli u. 4 tundi. See tähendab, et analüüsile kokku kulus umbes 10 töötundi ning jäädi sätestatud ajapiirangu sisse. Teostatud võrdluse detailsem ülevaade on saadaval lisas: Lisa 2: AHP võrdlus. Hierarhiliselt struktureeritud otsustusprobleem on kujutatud alloleval joonisel (Joonis 4).



Joonis 4. Hierarhiliselt struktureeritud otsustusprobleem

2.3.1 Kriteeriumite võrdlemine

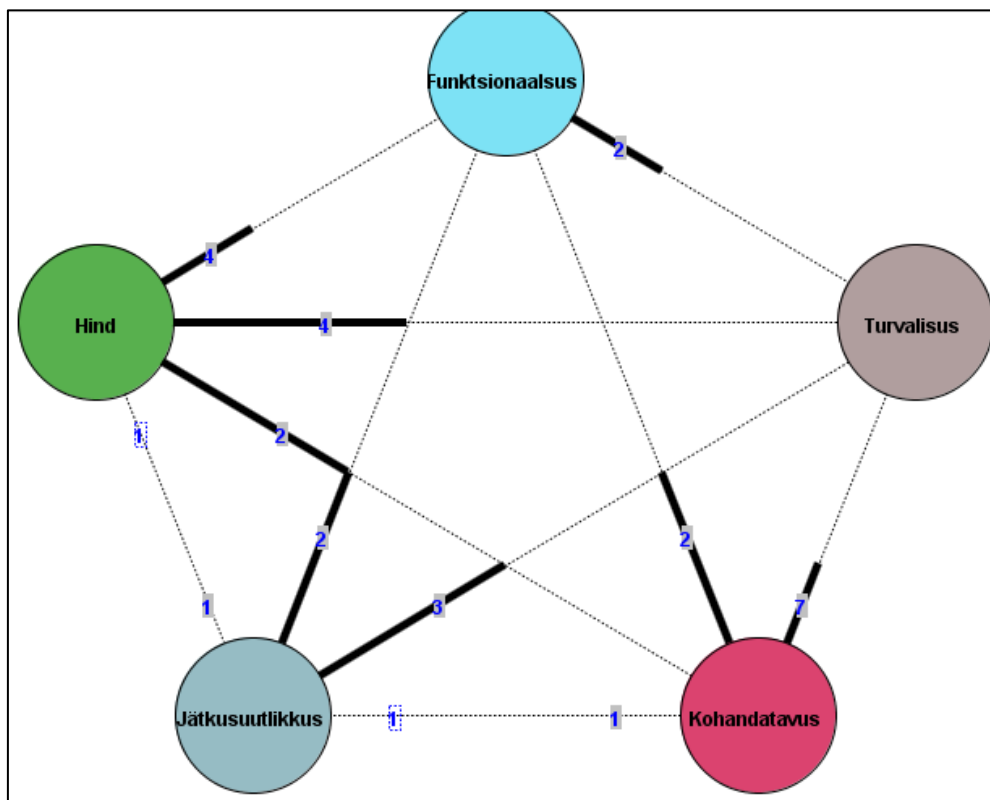
Järgnevalt võrdlesid hindajad paarikaupa defineeritud kriteeriume. Kusjuures, kui kriteeriumite defineerimisel lähtuti tarkvarast ja täpsemalt PIM tarkvarast üleüldiselt, siis nüüd lähtuti hinnangute andmisel konkreetsest projektist, seejuures võeti arvesse kliendi ja arenduspartneri poolseid nõudeid ning konkreetse projekti eripärasid. Hinnangud kriteeriumitele on kujutatud ka järgneval joonisel (Joonis 5).

	Funktsionaalsus	Turvalisus	Kohandatavus	Jätkusuutlikkus	Hind
Funktsionaalsus	1,00	2,00	0,50	0,50	0,25
Turvalisus	0,50	1,00	0,14	0,33	0,25
Kohandatavus	2,00	7,00	1,00	1,00	0,50
Jätkusuutlikkus	2,00	3,00	1,00	1,00	1,00
Hind	4,00	4,00	2,00	1,00	1,00

Joonis 5. Hinnangud kriteeriumitele

Siit võime välja lugeda, et võrreldes teiste kriteeriumitega pidasid arendajad näiteks turvalisust igal juhul ebaolulisemaks. Peamine argument oli, et tegu on kinnise süsteemiga, mida hakkavad kasutama peamiselt ettevõtte enda töötajad. Eri tahkudest olulisemaks sai hind, mis võrdlustes oli alati kas olulisem või samaväärne teiste kriteeriumitega.

Kriteeriumite võrdlust saab väljendada ka alloleva joonisena (Joonis 6), kus on välja toodud võrdluste olulisemaks hinnatud pool (paksu joonega). On näha, et hind on võrdlustes olulisim, samas turvalisust on peetud kõige ebaolulisemaks.



Joonis 6. Kriteeriumite võrdlemise graaf

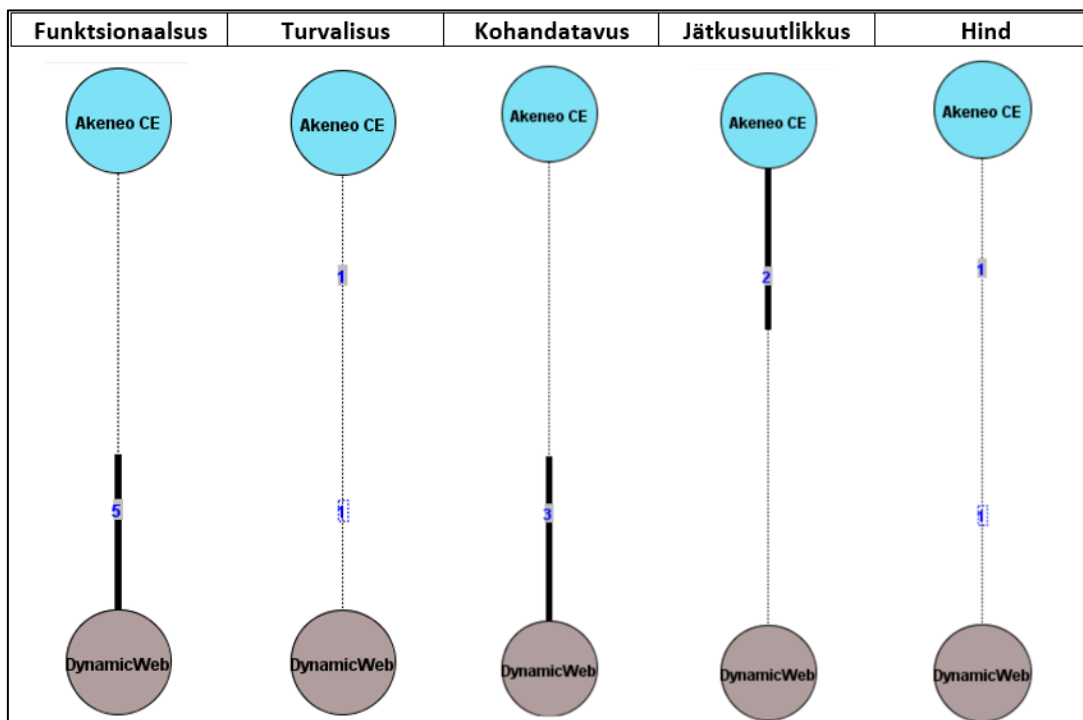
2.3.2 Alternatiivide võrdlemine

Viimase sammuna võrreldi alternatiive (Akeneo CE ja Dynamicweb PIM) nende vastavusele kriteeriumitega.

Dynamicweb-i tasuline lahendus oli olemasoleva funktsionaalsuse poolest oluliselt paremaks hinnatud Akeneo tasuta versioonist. Näiteks hinnati kõrgemalt Dynamicwebi liidestatvust teiste rakendustega ning paremaid võimalusi toodete ja tootekategooriate rikastamisel.

Dynamicweb-i kõrge hinnang funktsionaalsuses ning mõõdukas paremus kohandatavuses saidki määravaks eelistatuse välja selgitamisel: vastavalt 0,56 Dynamicweb ja 0,44 Akeneo CE.

Hinnangud alternatiividele väljendusid järgneva joonisega (Joonis 7).



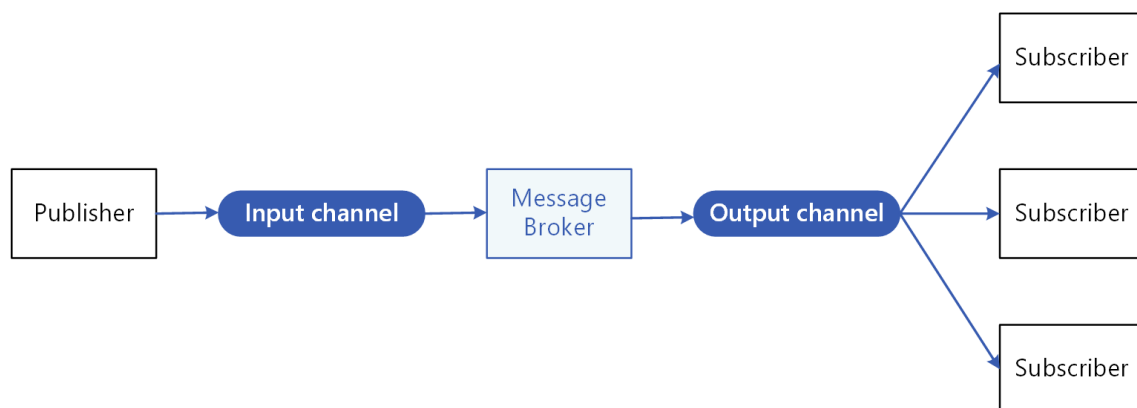
Joonis 7. Hinnangud alternatiividele

3 Tarkvara prototüüp

Prototüübi loomisel oli kaks eesmärki. Esiteks demonstreerib prototüüp Dynamicweb tarkvaralahendust ning vastavust hindamise käigus tehtud eeldustele. See võimaldas paremini formuleerida ettekujutust tarkvara eri tahkudest, valideerida analüüsi käigus tehtud oletusi ning anda lõplik hinnang tarkvara sobivusele. Teiseks võimaldas eelmainitud hinnang tarkvara sobivusele anda omakorda hinnangu AHP meetodika kasutamise sobivusele antud ülesande täitmisel.

3.1 Laiendatavuse võimalused

Dynamicwebi funktsionaalsuse laiendamiseks on eri viise. Võimalik on kasutada *Publisher-Subscriber* mustrit, mis võimaldab rakendusel asünkroonselt teavitada sündmustest (näiteks toote lisandumine poe nimekirja), mida teised süsteemi komponendid saavad kasutada, seejuures sidumata saatjaid ja vastuvõtjaid.[22] *Publisher-Subscriber* mustrit on kujutatud ka alloleval joonisel (Joonis 8)



Joonis 8. *Publisher-Subscriber* muster [22]

Kasutada saab ka kalenderplaaniga taustatöid. Näiteks saab seadistada taustatöid e- kirjade väljastamiseks või välistest kanalitest andmete pärimiseks[23]. Rakenduse esialgset funktsionaalsust saab muuta ka *Provider* klasside abil. Näiteks *PriceProvider*

võimaldab muuta hinnaarvutamise meetodit, *TaxProvider* klassiga saab määratleda maksude arvutamist jpm[24]. *ConfigurableAddIn* klass võimaldab kohandatud sisenditega täiendada haldusliidese teatud valdkondi. Näiteks saab vajadusel allahindluse arvutamiseks lisada kohandatud sisendina riigi või vanuserühma, mille alusel allahindlus kehtib.[24]

Kui eelmainitud lahendused olid olemas ka Dynamicweb-i varasemates versioonides, siis Dynamicweb 10-ga lisandusid menüü kohandamine, uute vaadete loomine, vanade kohandamine ja vaadete komplekteerimine. Lisandusid ka käsklused, päringud ja mudelid vaadetesse sisu lisamiseks.[25]

3.2 Prototüübi disain

Prototüübi disain sätestas prototüübi loomise eesmärgid ning kuidas hinnati eelneva analüüsi käigus määratletud tarkvara komponente. Disain määratleti koostöös eelnevas analüüsis osalenud hindajatega intervjuu käigus.

Tarkvara funktsionaalsust hinnatakse kasutades baasrakendust. Täpsemalt sätestatakse kasutajatele kasutajarollid ning rollidele vastavad õigused. Kasutajaliides peab võimaldama rollide põhiselt menüüpunktide peitmist. Tootehalduses peavad kasutajad saama muuta ainult lubatud välju.

Kohandatavuse hindamine jaguneb kahte ossa. Esiteks hinnatakse kohandatavust läbi kasutajaliidese. Teiseks tuleb teha rakendusele lisaarendus, mis võimaldab hinnata rakenduse kohandatavuse võimekust läbi tarkvaraliste muudatuste. Lisaarendusena luuakse rakendusesse uus vaade ja menüüpunktid, kuid vaadeldakse ka teisi laiendatavuse võimalusi.

Kuna lisaarenduse teostamiseks tuli kasutada Dynamicwebi poolt võimaldatud ressursse, siis võimaldas see hinnata ka mõningaid tulevikukindluse ja hinna aspekte nagu näiteks dokumentatsioon, kogukond ja lisafunktsionaalsuse lisamise hind.

Järgnevalt antakse selles peatükis ülevaade prototüübi loomisest ja kasutamisest. Prototüübi loomise ja kasutamisega seotud hinnanguid on kirjeldatud järeluste peatükis.

3.3 Prototüübi loomine

Andmebaasiserverina on kasutusel lokaalne Microsoft SQL Server 2022 Developer rakendus ning andmebaasi haldamiseks on kasutusel Microsoft SQL Server Management Studio 18. Programmeerimiskeskonnana on kasutusel IntelliJ Rider IDE. Prototüübi loomine ja lisaarenduse tegemine on täpsemalt lahti kirjeldatud lisa 3 – Prototüüp.

Esmalt loodi andmebaasiserverisse uus andmebaas ja kasutaja, kellele määrati ka *sysadmin* serveri roll. Uus Dynamicweb 10 rakendus loodi käsureal kasutades Dynamicweb 10 mallide *NuGet* paketti. Rakenduse esmasel käivitamisel kuvati seadistamise vaade, kus määratakse failide kaust, andmebaas, administraator kasutaja ja litsents. Litsentsina on kasutusel 30-päeva proovilitsents. Selleks, et paremini simuleerida tarkvaraarenduse protsessi, loodi uus projekt lisaarenduste tegemiseks.

3.3.1 Lisaarendus

Lisaarenduse raames loodi uus menüüpuu *StatusSection* PIM-i peamenüüpunkti alla. Menüüpuule alammenüüpunkti lisamiseks tuli implementeerida oma versioon *NavigationNodeProvider* klassist ning üle kirjutada *GetRootNodes* meetod.

Kohandatud vaate (*screen*) loomiseks tehti esmalt uus andmemudeli klass, mis implementeerib *DataViewModelBase* klassi. Kohandatud vaade peab implementeerima ühte kolmest olemasolevast vaate tüübist: *ListScreen*, *EditScreen* või *OverviewScreen*. Staatusinfo kuvamiseks valiti *OverviewScreen*.

Vaate loomisel tulid ka kõige paremini esile probleemid, mis olid tingitud avaliku lähtekoodi puudumisest ja puudulikust dokumentatsioonist. Seetõttu puudus võimalus leida juhend, mis aitaks implementeerida *OverviewScreen*-i meetodit *BuildOverviewScreen*. Seetõttu tuli näidete leidmiseks *Dynamicweb.CoreUI* pakett decompileerida, kasutades JetBrains dotPeek rakendust.

Kasutades Dynamicweb CLI rakendust, lisati loodud lähtekoodi failid veebirakenduse *Files* kausta. Rakenduse taaskäivitamisel kuvati *Products* peamenüüpunkti all *Overview* menüüpuu, kus asus alamenüüpunkt *Status*, mis omakorda avas kohandatud vaate.

3.3.2 Kohandamine läbi kasutajaliidese

Läbi kasutajaliidese loodi kaks kontot eri rollidega: Tootehaldaja ning Tarnija. Mõlemale kasutajale võimaldati ligipääs ainult PIM-moodulile ning kindlatele vaadetele.

Tarnija puhul võimaldati kohandatud õigustega ligipääs tema hallatavale laole ning Tootehaldaja puhul võimaldati kohandatud ligipääs kõikidele ladudele ja poodidele. Toodete haldamisel on võimalik kasutajaõigusi väljade kaupa piirata või peita kasutajate eest ebavajalikku infot.

4 Järeldused

Käesolevas peatükis, tehakse tagasisivaade tehtud tööle. Analüüsitakse AHP analüüsi ja prototüüpimisega seonduvaid tulemusi ja tagasiside ning antakse eksperimendile lõplik hinnang.

4.1.1 Prototüüp

Prototüüpimise tulemusena loodi lokaalselt majutatud PIM tarkvaralahendus, milleks kasutati Dynamicweb 10 tarkvarakomplekti.

Projektis osalejate kollektiivsel hinnangul katab tarkvaralahendus kliendi baasvajadused. Süsteem on piisaval määral konfigureeritav, mis tagab autoriseeritud ligipääsu andmetele ning saab eeldada, et süsteemi modifitseerides on võimalik katta ka potentsiaalsed tulevikunõuded.

Peamised probleemid olid tingitud faktist, et käesoleva töö kirjutamise ajal ei olnud Dynamicweb 10 täisversioon veel saadaval. Enim pärssis prototüübi loomist dokumentatsiooni puudumine. Samas võib Dynamicweb 9 dokumentatsiooni põhjal hinnata, et täisversiooni dokumentatsioon saab olema piisav. Lisaks hõlbustab uue tarkvara kasutuselevõtmist Dynamicwebi partnerite kaasamise programm[26], korraldatavad koolitused ja üritused[27] ning aktiivne foorum. Eelnevale infole tuginedes ei saa ka järeldada, et eelmainitud kitsendused tõstaks tarkvara kasutamisega või lisaarenduste teostamisega seonduvat ajakulu või hinda.

4.1.2 AHP meetoodika

Prototüüpimisel tekkinud järelduste põhjal saab öelda, et analüüsi käigus tehtud hinnangud pidasid paika, millest omakorda võib järeldada, et AHP meetoodika kasutamine ettevõtte olustikus oli edukas.

Siinkohal tuleb tunnistada, et analüüsi käigus tehtud töö tarkvaraga tutvumisel on niigi vältimatu osa analüüsist ning selles aspektis ei toonud AHP meetoodika kasutamine mingit

muudatust. Lisaks suurendas AHP meetoodika kasutamine märgatavalt analüüsile kulunud ajakulu.

Siiski andis meetoodika kasutamine analüüsile teatava raamistiku. Meetoodika kasutamise käigus väljendusid tarkvara erinevad tahud, mis andsid analüüsile fookuse ja raamid. Kasvanud ajakulu üheks põhjuseks on kindlasti ka meetoodika esmakordne kasutamine. Võib eeldada, et meetoodika kasutamise järgnevate kordadega tekib ka vilumus ning väheneb ajakulu. Meetoodika korduv kasutamine aitab ka järgnevatel kasutuskordadel täiustada ja täpsustada analüüsis kasutatud tarkvara kriteeriumeid.

4.1.3 Eksperiment

Intervjuu käigus kogutud info ettevõtte tööprotsesside kohta andis aluse hüpoteesile, et AHP meetod võiks sobida ettevõtte olustikus tarkvara võrdlemiseks. AHP meetodil PIM tarkvara võrdlemise vaatlemine tõi välja meetoodika rakendamise probleemid, nagu lisandunud keerukus ja ajakulu. Samas saab eeldada, et tööprotsessi korrates väheneb ajakulu ning kogemuste kasvades paraneb potentsiaalselt ka AHP meetoodika kriteeriumite defineerimine ning omakorda analüüsi väljund. Paraku ei saa neid eeldusi käesoleva tööga kinnitada. Eelduste kinnitamine vajab edasist uurimist.

Prototüüpimise kasutamisest saadud tagasiside võimaldas kinnitada AHP analüüsi käigus tehtud oletusi ning saab kinnitada, et AHP meetod on sobiv kasutamiseks ettevõtte olustikus.

5 Kokkuvõte

Käesoleva bakalaureusetöö eesmärk oli analüüsida AHP meetodi sobivust ettevõtte olustikus. Sobivuse hindamiseks viidi läbi juhtumianalüüs, mille käigus vaadeldi tooteinfo haldamise tarkvara võrdlemist AHP meetodiga.

Eksperimendi toetamiseks loodi valituks osutunud tarkvarast prototüüp. Prototüüpimise tulemusena loodi lokaalselt majutatud PIM tarkvaralahendus, kasutades Dynamicweb 10 tarkvarakomplekti.

Analüüsides tehtud tööd ja tagasiside ning kasutades prototüüpi kinnitati, et analüütiline hierarhiate meetod sobib ettevõtte olustikus tarkvara võrdlemiseks.

Kasutatud kirjandus

- [1] I. Shin, "Adoption of Enterprise Application Software and Firm Performance," *Small Bus Econ*, vol. 26, pp. 241-256, 2006. doi: 10.1007/s11187-005-0215-9
- [2] S. Williams and N. Williams, *The Profit Impact of Business Intelligence*. Elsevier, Jul. 2010.
- [3] P. Bourque and R.E. Fairley, eds., *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society, 2014; www.swebok.org.
- [4] Zimmermann, O., Gschwind, T., Küster, J., Leymann, F., and Schuster, N., "Reusable Architectural Decision Models for Enterprise Application Development," in: Overhage, S., Szyperski, C.A., Reussner, R., Stafford, J.A. (eds.), *Software Architectures, Components, and Applications. QoSA 2007*. Berlin, Heidelberg: Springer, 2007, pp. 23-38, doi: 10.1007/978-3-540-77619-2_2.
- [5] Gati, I., Levin, N., Landman-Tal, S. (2019). Decision-Making Models and Career Guidance. In: Athanasou, J.A., Perera, H.N. (eds) *International Handbook of Career Guidance*. Springer, Cham. Available: https://doi.org/10.1007/978-3-030-25153-6_6.
- [6] A. T. Misirli and A. B. Bener, "Bayesian Networks For Evidence-Based Decision-Making in Software Engineering," in *IEEE Transactions on Software Engineering*, vol. 40, no. 6, pp. 533-554, 1 June 2014, doi: 10.1109/TSE.2014.2321179.
- [7] E. H. Forman and S. I. Gass, "The Analytic Hierarchy Process: An Exposition," *Operations Research*, vol. 49, no. 4, pp. 469-486, Jul./Aug. 2001.
- [8] 1000minds, "Decision Making," [Võrgumaterjal]. Available: <https://www.1000minds.com/decision-making>. [Kasutatud: Apr. 22, 2023].
- [9] S. Siraj, L. Mikhailov and J. A. Keane, "PriEsT: an interactive decision support tool to estimate priorities from pairwise comparison judgments," *International Transactions in Operational Research*, vol. 22, no. 2, pp. 217-235, Mar. 2015.
- [10] S. Sipahi and M. Timor, "The analytic hierarchy process and analytic network process: an overview of applications," *International Journal of Production Research*, vol. 48, no. 6, pp. 775-808, Mar. 2010.
- [11] A.Frolova, "Väikeettevõttele laotarkvara valimine AHP meetodi abil," *Bakalaureusetöö, Tarkvarateaduse Instituut, Tallinna Tehnikaülikool, Tallinn, Estonia*, 2018.
- [12] S.N.A Darmantya, "Application of Analytical Hierarchy Process (AHP) Method as Decision Support in Determining Internet Service Providers," May 2020, doi: <https://doi.org/10.31219/osf.io/9skxy>.
- [13] Akeneo, "What is a PIM?," [Võrgumaterjal]. Available: <https://www.akeneo.com/what-is-a-pim/>. [Kasutatud: 10.04.2023]
- [14] Saaty, T.L. Decision making — the Analytic Hierarchy and Network Processes (AHP/ANP). *J. Syst. Sci. Syst. Eng.* 13, 1–35 (2004). <https://doi.org/10.1007/s11518-006-0151-5>
- [15] Siraj, S., Mikhailov, L. and Keane, J.A. (2015), PriEsT: an interactive decision support tool to estimate priorities from pairwise comparison judgments. *Intl. Trans. in Op. Res.*, 22: 217-235. <https://doi.org/10.1111/itor.12054>
- [16] Akeneo, "Akeneo PIM Community Edition" [Võrgumaterjal]. Available: <https://www.akeneo.com/akeneo-pim-community-edition/> [Kasutatud 22.04.2023].

- [17] GitHub, “Akeneo PIM Development Repository”, [Vörgumaterjal]. Available: <https://github.com/akeneo/pim-community-dev> [Kasutatud: 21.04.2023].
- [18] Dynamicweb, “Introducing Dynamicweb 10,” [Vörgumaterjal]. Available: <https://Dynamicweb.com/resources/downloads/white-papers/introducing-Dynamicweb-10> [Kasutatud: 10.04.2023]
- [19] Dynamicweb, “dw10-quickstart,” [Vörgumaterjal]. Available: <https://doc.Dynamicweb.com/dw10-quickstart> [Kasutatud: 10.04.2023].
- [20] E. H. Forman and M. A. Selly, Decision By Objectives: How To Convince Others That You Are Right, expanded 3rd ed. Alexandria, VA, USA: Decisions and Designs Inc., 2001.
- [21] F. Losavio, L. Chirinos, A. Matteo, N. Lévy, and A. Ramdane-Cherif, “ISO quality standards for measuring architectures,” *Journal of Systems and Software*, vol. 72, no. 2, pp. 209–223, Jul. 2004, doi: [https://doi.org/10.1016/s0164-1212\(03\)00114-6](https://doi.org/10.1016/s0164-1212(03)00114-6).
- [22] Microsoft, “Publisher-Subscriber pattern” [Vörgumaterjal]. Available: <https://learn.microsoft.com/en-us/azure/architecture/patterns/publisher-subscriber> [Kasutatud: 02.05.2023].
- [23] Dynamicweb, “Scheduled Tasks,” [Vörgumaterjal]. Available: <https://doc.dynamicweb.com/documentation-9/platform/platform-tools/scheduled-tasks> [Kasutatud: 02.05.2023].
- [24] Dynamicweb, “3.3 Extending Ecommerce,” [Vörgumaterjal]. Available: <https://doc.dynamicweb.com/training/training/certifications/t3-platform-developer/t3-platform-developer/3-3-extending-ecommerce> [Kasutatud: 02.05.2023].
- [25] Dynamicweb, “Extensibility & Customization,” [Vörgumaterjal]. Available: <https://doc.dynamicweb.com/dw10-quickstart/frontpage/developers/extensibility-customization> [Kasutatud: 02.05.2023].
- [26] Dynamicweb, “Partner Enablement Program,” [Vörgumaterjal]. Available: <https://dynamicweb.com/become-a-partner/partner-onboarding-program> [Kasutatud: 10.04.2023].
- [27] Dynamicweb, “Events,” [Vörgumaterjal]. Available: <https://dynamicweb.com/resources/events> [Kasutatud: 10.04.2023].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Rain Varblane

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose AHP meetodi kasutamine PIM tarkvara valikul, mille juhendaja on Jaanus Pöial
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

15.05.2022

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – AHP võrdlus

Kriteeriumite defineerimine

Funktsionaalsus	Kui hästi täidab <i>OOTB</i> tarkvara kliendi hetkevajadused ja potentsiaalsed tulevikusoovid.
Turvalisus	Teadaolevad turvavead, potentsiaalsed ohud, riskide maandamine
Kohandatavus	Rakenduse modifitseerimine. Kui lihtne on muuta olemasolevat funktsionaalsust, lisada pluginaid või muuta rakenduse välimust/kujundust?
Tulevikukindlus	Populaarsus. Foorumi ja kasutajatoe olemasolu.
Hind	Litsents, pluginate ostmise, hosting jms

Kommentaariid

Hinnangute andmisel peab analüüsi läbiviija kriteeriumid lahti mõtestama just konkreetse projekti raames. Kriteeriumid ja nende kirjeldused omavad suunavat ja soovituslikku eesmärki. Näiteks avatud lähtekood, tehniline tugi või API olemasolu ei pruugi konkreetse projekti ja tarkvara puhul üldse oluline olla.

Kas on oluline *event driven* arhitektuur, REST funktsionaalsus või midagi muud?
Kas tarkvarauuendused, populaarsus tehnoloogiamaastikul või aktiivne kogukond on kliendi või arendajate jaoks oluline?

Funktsionaalsus

Funktsioonid, mis peavad kohe alguses kaetud olema. (Logimine, Liidestatavus, Taustatööd, Activity log jms). Tarkvara olemasolev (*ootb*) funktsionaalsus ja selle suhe kliendiga.

- Mis on olemas?
- Kui palju tööd on kohe alguses vaja teha?
- Kui pikalt saab olemasolevat funktsionaalsust muutusteta kasutada?
- Kliendi tulevikuplaanide katmine (orgaaniline kasv, teistesse riikidesse laienemine, versioonivahetus jms)

Turvalisus

Sõltub kliendist (mõttele maksuamet vs väikeettevõtja).

- Kas arendajal *live* andmebaasile on ligipääs?
- Kas arendaja näeb klientide isikuandmeid?

- Kas rakendus on avalikult kõigile kättesaadav või asub lokaalses serveris ja on väljaspoole kättesaamatu?
- Kas majutuse tagab arenduspartner, pilveteenus või mõni muu lahendus?

Kohandatavus

Olemasoleva täiendamine, uue loomine, disaini muutmine. Klasside ülekirjutamine ja laiendamine.

- Kuidas toimub sisu haldamine? Vabad käed vs kontrollitud keskkond.
- Olemasoleva funktsionaalsuse muutmise lihtsus/võimalikkus - Open source kood, Versioonivahetuste läbiviimine, Otsingumootori vahetamine.
- Lisafunktsionaalsuse lisamise lihtsus/võimalikkus - Kas on olemas pluginad, kas saame ise teha enda omi juurde. Liidestatavus: navision jms.
- Rakenduse visuaalse disaini muutmise lihtsus/võimalikkus - Kas arendame nullist, või cms, või api?

Turuseis/jätkusuutlikkus

- Populaarsus - Turuseis, populaarsus, kasutajate arv.
- Kogukond - Foorumid vms suhtluskeskkonnad, üritused, sündmused, kasutajatugi, litsentsid/koolitused
- Dokumentatsioon
- Uuendused - Uuenduste väljastamise sagedus. Kas üldse uuendatakse?

Hind

- Toote baashind, püsivad kulud ning kohustused tootja ees
- Lisafunktsionaalsuse lisamise hind
- Hostinguga seonduvad kulud
- Kasutajatoe hind.

AHP hinnangud

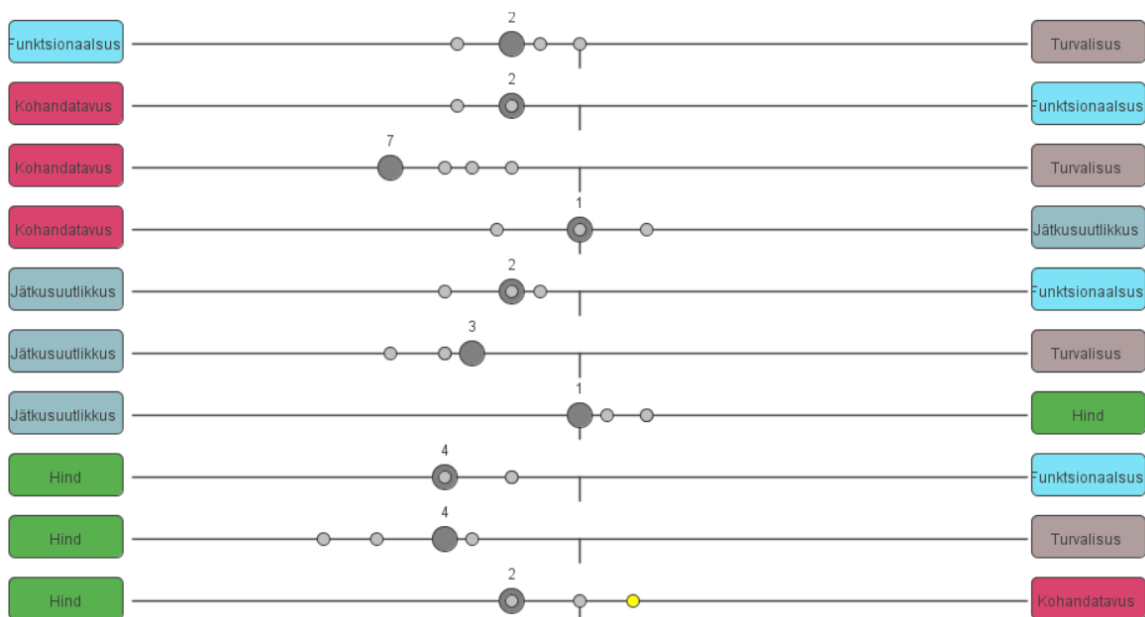
Funktsionaalsus vs Turvalisus	Kinnine süsteem. Turvalisus on parandatav. Hea funktsionaalsus eeldab turvalisust.
Funktsionaalsus vs Kohandatavus	Kohandatavuse kerge paremus. Isegi kui esialgsed vajadused on maksimaalselt kaetud, siis pikas perspektiivis on lisafunktsionaalsuse võimekus rohkem väärt.
Funktsionaalsus vs Jätkusuutlikkus	Pigem jätkusuutlikkus, kuigi Antud projekti raames on oluline katta ära baasvajadused, siis dokumentatsioon ja esialgne tugi kasutusele võtul on olulisem.
Funktsionaalsus vs Hind	Hind on olulisem, sest me saame lisafunktsionaalsust juurde luua. Jooksvad kulud + arendajate tasud. Meie pakkumistel on hind oluline!

Turvalisus vs Jätkusuutlikkus	Pikaajaline efektiivsus ning ressursside säästmise. Uuendused, dokumentatsioon.
Turvalisus vs Hind	Projekti hinnatundlikkus. Funktsionaalsuse aspekt toetab turvalisust.
Kohandatavus vs Jätkusuutlikkus	Võrdsed. Kohandatavus langeb, kui meil puudub dokumentatsioon. Aegunud toodet ei ole hea kohandada, see ei käi ajaga kaasas. Samas deprecated toode võib katta vajadused.
Kohandatavus vs Turvalisus	Kohandatavus on kindlalt olulisem, sest puuduva turvalisuse saab juurde arendada. Samas nullist turvalisuse aspekti üles ehitada on liiga kulukas.
Kohandatavus vs Hind	Hind on kergelt olulisem. Meie projektides on alati mingigi kohandatavuse element. Eriti PIMI-i konteksti pole max kohandatavus nii oluline.
Hind vs Jätkusuutlikkus	Klient kardab tundmatut tv (arendaja ka). Ei ole usaldusväärne. Samas hinnatundlikkus. Arvamused lähevad lahku, mõlemale tahetakse väiksemat paremust. Kompromissina võrdväärsus.

Kriteeriumite hinnangumaatriks

	Funktsionaalsus	Turvalisus	Kohandatavus	Jätkusuutlikkus	Hind
Funktsionaalsus	1,00	2,00	0,50	0,50	0,25
Turvalisus	0,50	1,00	0,14	0,33	0,25
Kohandatavus	2,00	7,00	1,00	1,00	0,50
Jätkusuutlikkus	2,00	3,00	1,00	1,00	1,00
Hind	4,00	4,00	2,00	1,00	1,00

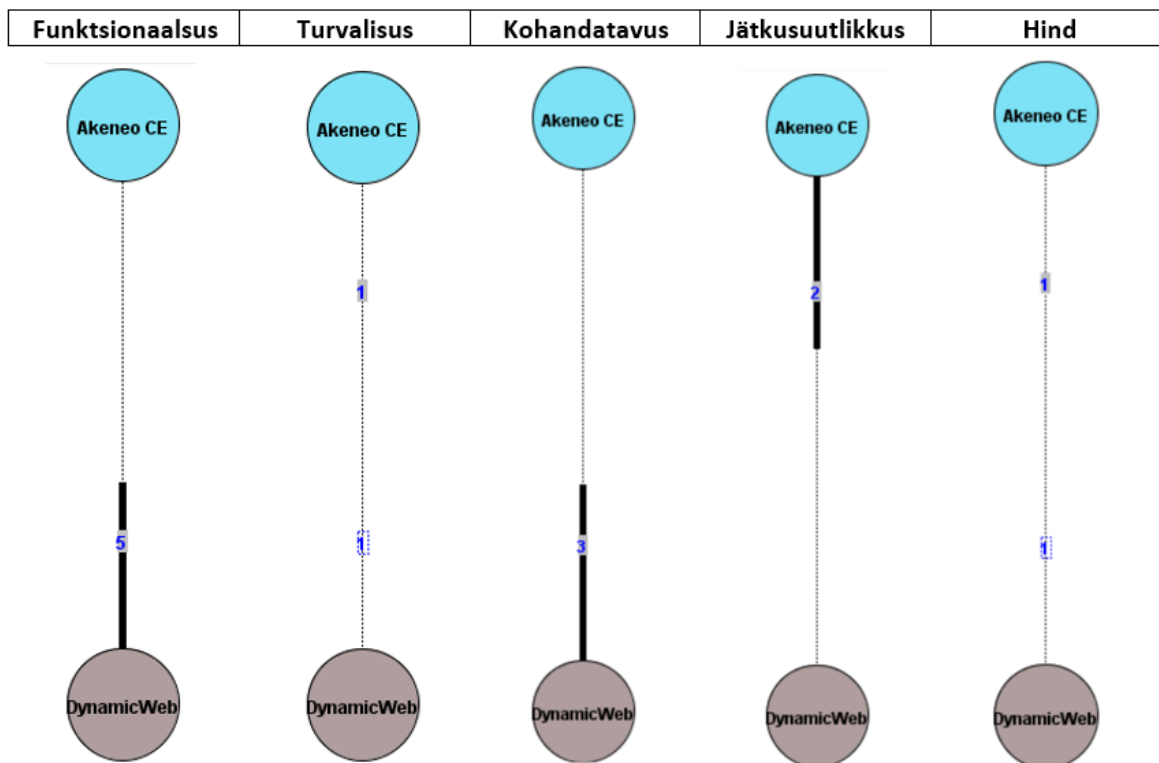
Kriteeriumite hinnangud



Alternatiivide võrdlemine:

Funktsionaalsus	Dynamicweb mikroteenused, põhjalikum infrastruktuur. Lisamoodulid: (nt e-commerce, marketing jm.). API tugi mõlemal. DW kategooriate rikastamine,
Turvalisus	Võrdsed. Rollide, kasutajate ja gruppide haldus. Andmete varundamine ja taastamine.
Kohandatavus	Muudatuste tegemine läbi kasutajaliidese (DW built-in parem võimekus). DW Headless rakendus, API tugi mõlemal. Erinevused muudatuste teostamisel. DW ERP plug-in moodulid. Pigem DW.
Jätkusuutlikkus	DW automaatsed uuendused ja kasutajatugi. Akeneo on populaarsem, tuntum, suurem kogukond. Akeneo dokumentatsioon parem. Akeneo kerge paremus.
Hind	Võrdsed. Akeneo on tasuta, kuid vajab tõenäoliselt kohe parandusi ja täiendamist. DW on tasuline, kuid on rohkem <i>ootb</i> funktsionaalsust, hinnas on ka näiteks kasutajatugi.

Alternatiivide võrdluse väärtused



Lisa 3 – Prototüüp

Keskkond

Dynamicweb 10

Dynamicweb CLI

SQL Server 2022 Developer

.NET 7.0

Kasutatud rakendused

IntelliJ Rider

MS SQL Server Management Studio 18

Keskkonna seadistamine

Andmebaas

Server: localhost\MSSQLSERVER01

Database: dw_demo01

User: rvarbl01

```
CREATE DATABASE dw_demo01;  
CREATE LOGIN rvarbl01 WITH PASSWORD = 'pw123';  
ALTER SERVER ROLE sysadmin ADD MEMBER rvarbl01;  
CREATE USER admin01 FOR LOGIN rvarbl01;  
ALTER USER admin01 WITH DEFAULT_SCHEMA=dbo;
```

DW10 template package

```
dotnet new install Dynamicweb.ProjectTemplates::0.0.7
```

```
dotnet new dw10-suite
```

DW10 setup

DynamicWeb

- Welcome to Dynamicweb 10
- Create files repository
- Setup database
- Set administrator password
- You are ready to go

Welcome to DynamicWeb 10 setup guide
This guide will help you set up DynamicWeb for the first time

You can read more about the installation guide at
<https://doc.dynamicweb.com/>

[Get started](#)

DynamicWeb

- Welcome to Dynamicweb 10
- Create files repository
- Setup database
- Set administrator password
- You are ready to go

Create files repository
This step will set up a files repository for storing images, documents and other files.

Create or attach /Files

- Create a new /Files folder
- Map to an existing /Files folder

[Previous step](#) [Create files](#)

Setup database

Set up database to use or install a new DynamicWeb database.
This step allows you to create a blank database or attach to an existing Dynamicweb database

Choose your database type

- SQL-Server database (2016 or newer)
Use this option if you want to use Microsoft SQL-Server locally on your network
- Azure database
Use this option if you want to use Microsoft Azure
- Manually specify connection string
Use this option if you want to specify your own connection string (Advanced)

Connection settings

Server

ie: localhost\SQLEXPRESS

Database

Create database if it does not exist.
The specified user must have permission to create a database on the server

Integrated mode
This uses the current Windows user for authentication

Username

Password

Dynamicweb connected successfully.

Set administrator password

You are ready to go

Configure administrator

Specify a username and password for the administrator account

Name

Email

User name

Password

Confirm password

© 2023 DynamicWeb Software A/S. All rights reserved.

DynamicWeb

- Let's install the license
- Login / Trial installation
- Pick a customer
- Select a license type
- Select the environment
- Congratulations!

Your trial is ready

The installation is now setup and ready for you to make awesome websites and ecommerce solutions

You are now ready to go click the button to start the administration of DynamicWeb

The trial will expire in 30 days!

[Previous step](#) [Go to administration](#)

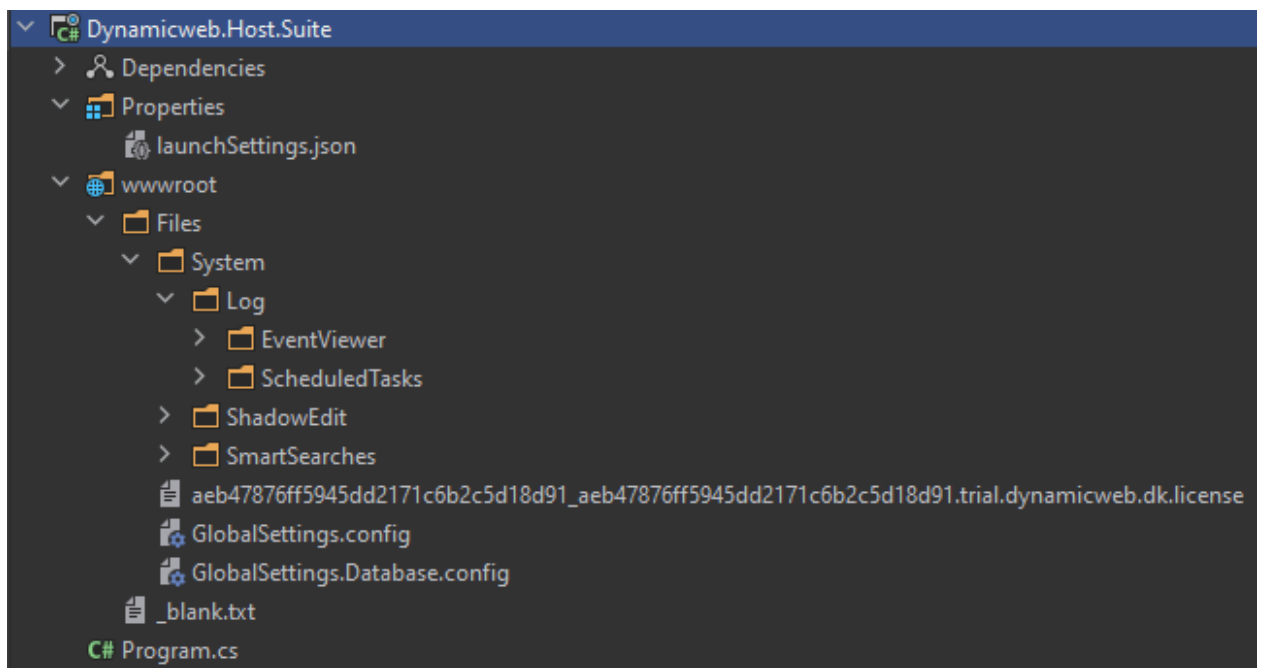
DynamicWeb

- INSIGHTS
- CONTENT
- ASSETS
- USERS
- PRODUCTS
- COMMERCE
- MARKETING
- INTEGRATION
- APPS
- SETTINGS

Insights

- Monitoring
 - Reports
 - Event viewer
 - Health
 - Notifications
 - Logs
- Analytics

Projekti failistruktuur:



launchSettings.json

```
{
  "profiles": {
    "Dynamicweb.CoreUI.Host": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": true,
      "launchUrl": "https://localhost:6001/admin",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      },
      "applicationUrl":
"https://localhost:6001;http://localhost:6000",
      "nativeDebugging": true
    }
  }
}
```

Program.cs

```
using Microsoft.AspNetCore.Builder;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddDynamicweb(builder.Environment, builder.Configuration);

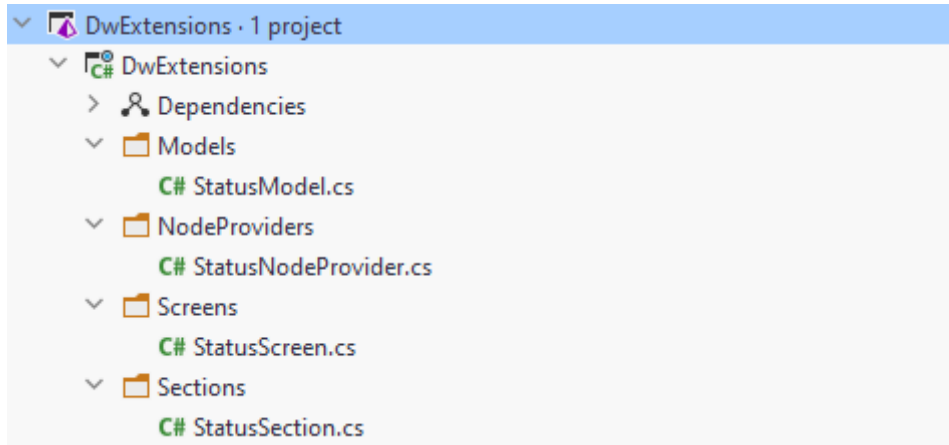
var app:WebApplication = builder.Build();

app.UseDynamicweb();

app.Run();
```

Rakenduse kohandamine

Uus projekt arendustega



```
dotnet add package Dynamicweb.Suite --version 10.0.0-RC002
```

StatusSection.cs

```
using Dynamicweb.CoreUI.Navigation;
using Dynamicweb.Products.UI;

namespace DwExtensions.Sections;

public class StatusSection: NavigationSection<ProductsArea>
{
    public StatusSection(NavigationContext context) : base(context)
    {
        Name = "Status";
        Sort = 10;
    }
}
```

StatusModel.cs

```
using Dynamicweb.CoreUI.Data;

namespace DwExtensions.Models;

public class StatusModel : DataViewModelBase
{ }
```

StatusNodeProvider.cs

```
using DwExtensions.Sections;
using Dynamicweb.CoreUI.Actions;
using Dynamicweb.CoreUI.Actions.Implementations;
using Dynamicweb.CoreUI.Icons;
using Dynamicweb.CoreUI.Navigation;
using Dynamicweb.Insights.UI.Queries;
using Dynamicweb.Insights.UI.Screens;

namespace DwExtensions.NodeProviders;

public class StatusNodeProvider :
    NavigationNodeProvider<StatusSection>
{
    public override IEnumerable<NavigationNode> GetRootNodes()
    {
        return new List<NavigationNode>
        {
            new NavigationNode
            {
                Name = "Overview",
                Id = "Overview",
                Icon = Icon.Compass,
                HasSubNodes = true,
                ContextActions = GetContextActions(),
            }
        };
    }

    public override IEnumerable<NavigationNode>
    GetSubNodes(NavigationNodePath parentNodePath)
    {
        var nodes = new List<NavigationNode>
        {
            new NavigationNode
            {
                Name = "Status",
                Id = "Status",
                Icon = Icon.Adjust,
                ContextActions = GetContextActions(),
                NodeAction = new NavigateScreenAction<StatusScreen>(),
            }
        };

        return nodes;
    }

    private static List<ActionNode> GetContextActions()
    {
        return new List<ActionNode>
        {
            new ActionNode
            {
                Name = "Configure",
                Icon = Icon.Cog,
                Sort = 10,
                NodeAction = new AlertAction("Not implemented!")
            }
        };
    }
}
```

```
    }  
}
```

StatusScreen.cs

```
using DwExtensions.Models;  
using Dynamicweb.CoreUI;  
using Dynamicweb.CoreUI.Displays.Widgets;  
using Dynamicweb.CoreUI.Screens;  
  
namespace DwExtensions.Screens;  
  
public class StatusScreen : OverviewScreenBase<StatusModel>  
{  
    // Override default values  
    protected override string GetScreenName() => "Status";  
  
    protected override void BuildOverviewScreen()  
    {  
        AddComponent(GetStatusWidget(), "Status information");  
    }  
  
    private UiComponentBase GetStatusWidget()  
    {  
        return new InfoCardWidget  
        {  
            Value = new CardInfo  
            {  
                AdditionalInfo = new Dictionary<CardInfo.InfoKey,  
CardInfo.InfoValue>()  
                {  
                    {  
                        new CardInfo.InfoKey("Key"),  
                        new CardInfo.InfoValue("Value")  
                    }  
                }  
            }  
        };  
    }  
}
```

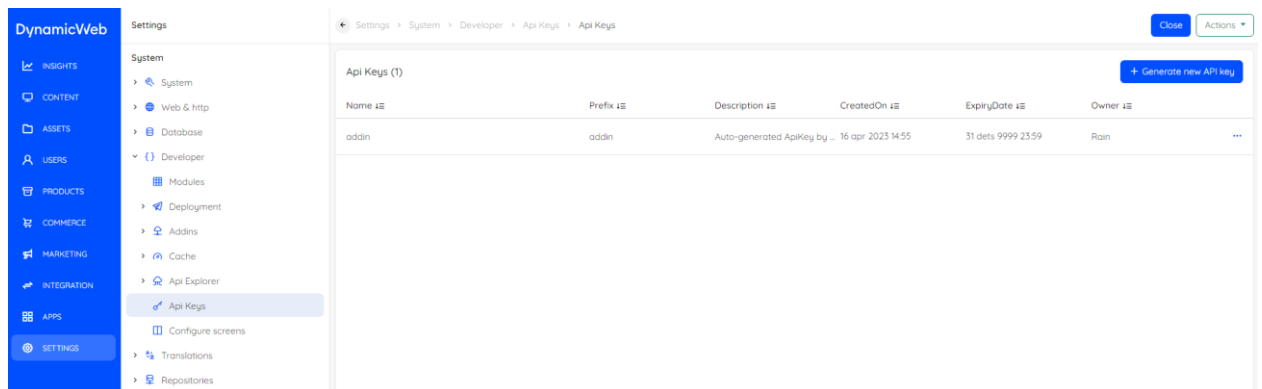
Lisaarenduse projekti lisamine

Paigalda Dynamicweb CLI - <https://github.com/Dynamicweb/CLI>

Logi sisse ja genereeri API võti:

```
C:\Users\Kasutaja>dw login
? environment: dw_demo01
? username: rvarbl
? password: [hidden]
The environment specified is missing parameters, please specify them
? protocol: https
? host: localhost:6001
Your current environment is now dw_demo01
```

Rakenduses kuvatakse genereeritud võti:



Expordi loodud failid DW rakenduse Files kausta

dw files <dirPath> <outPath> -fre

Löpptulemus

The screenshot displays the DynamicWeb administration interface. On the left is a blue navigation sidebar with the following menu items: INSIGHTS, CONTENT, ASSETS, USERS, PRODUCTS (highlighted), COMMERCE, MARKETING, INTEGRATION, APPS, and SETTINGS. The main content area is titled 'Products' and contains a sub-section 'Status'. Under 'Status', there is an 'Overview' section with a 'Status' item selected. Below this are sections for 'Queries' (My favorites, My queries, Shared queries, All products), 'Dynamic workspaces', 'Channels' (SHOP1), 'Data models', and 'Feeds'. The right-hand pane shows 'Status information' with a table header 'Key' and 'Value'. A breadcrumb trail at the top reads 'Products > Status > Overview > Status > Status'. A 'Close' button is located in the top right corner. The browser address bar at the bottom left shows 'https://localhost:6001/admin'.