



Jani Leppänen

# Terveydenhuollon tietojärjestelmien kehittäminen ketterillä menetelmillä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintäteknikka

Insinöörityö

27.5.2024

# Tiivistelmä

Tekijä:	Jani Leppänen
Otsikko:	Terveydenhuollon tietojärjestelmien kehittäminen ketterillä menetelmillä
Sivumäärä:	55 sivua + 4 liitettä
Aika:	27.5.2024
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Tieto- ja viestintätekniikka
Ammatillinen pääaine:	Hyvinvointi- ja terveysteknologia
Ohjaajat:	Lehtori Juha Havukumpu Toimitusjohtaja Karoliina Kaijasilta-Järvenpää

---

Insinööriyön aiheena oli kehittää Pharmac Finland Oy:lle ketterän ohjelmistokehityksen malli, joka ottaisi huomioon lääkkeiden koneellisen annosjakelun sekä muun Pharmacin toiminnan regulaatio- ja liiketoimintavaatimukset. Pharmacilla on käynnissä suuri tilaus-, toimitus- ja laskutusjärjestelmän uudistus, jonka puitteissa on tarkoitus kerätä tarpeita ohjelmistokehitysmalliin ja toisaalta saada työkaluja projektin tuleviin vaiheisiin. Lopputuotoksena oli tavoitteena saada konkreettisia työkaluja ja malleja Pharmacin ohjelmistokehityksessä hyödynnettäväksi.

Työ jakaantuu tiedonkeruuosioon ja ohjelmistokehitysmallin työstöön. Tiedonkeruuosiossa olemassa olevan tiedon osalta kartoitetaan Pharmacin toimintaan vaikuttavaa regulaatiota, jo tunnistettuja vaatimuksia uudelle järjestelmälle ja eri ohjelmistokehitysmalleja. Tiedonkeruuosion uuden tiedon osalta kerääminen tapahtuu eri kohderyhmille suunnatuilla työpajoilla. Ohjelmistokehitysmallin työstäminen tapahtuu kerätyn teorian tiedon ja Pharmacilta kartoitettujen tarpeiden pohjalta.

Pharmacille suunnitellussa mallissa päädyttiin hybridiratkaisuun, jossa hyödynnetään sekä perinteistä sovelluskehityksen vesiputousmallia että uudempaa, ketteriin menetelmiin perustuvaa Scrum-viitekehystä. Malli kattaa ohjelmistoprojektin alkuvaiheen suunnittelun, varsinaisen toteutuksen, sovelluksen käyttöönoton ja tämän jälkeen tapahtuvan ylläpidon sekä jatkuvan kehityksen. Insinööriyöhön ei kuulu itse Pharmacin uuden järjestelmän kehitysprojekti, mutta sen tiimoilta saatiin tunnistettua joitain suunnitellun mallin jatkokehitystarpeita.

Avainsanat: annosjakelu, sote-järjestelmät, sosiaali- ja terveydenhuollon tietojärjestelmät, ketterät menetelmät, ohjelmistokehitys

---

Tämän opinnäytetyön alkuperä on tarkastettu Turnitin Originality Check -ohjelmalla.

## Abstract

Author: Jani Leppänen  
Title: Developing healthcare information systems with agile methods  
Number of Pages: 55 pages + 4 appendices  
Date: 27 May 2024

Degree: Bachelor of Engineering  
Degree Programme: Information and Communications Technology  
Professional Major: Health Technology  
Supervisors: Juha Havukumpu, Senior Lecturer  
Karoliina Kaijasilta-Järvenpää, CEO

---

The study aimed to develop an agile software development model for Pharmac Finland Oy. The model should consider the regulatory and business requirements of automated dose dispensing as well as other business of Pharmac. Pharmac is planning an order, delivery and billing system, within which there is an intention to gather needs for the software development model and on the other hand obtain tools for future phases of the project. The final goal was to create concrete tools and models to be utilized in Pharmac's software development.

The study was divided into a data collection section and the development of the software model. The data collection section involves existing information mapping the regulation affecting Pharmac's operations, already identified requirements for the new system and various software development models. The collection of new information occurred through workshops designed for different target groups. The software development model was developed based on the theoretical knowledge gathered and the needs identified at Pharmac.

In the model designed for Pharmac, a hybrid solution was adopted that utilizes both the traditional waterfall model of application development and the more recent, agile-based Scrum framework. The model covers the initial planning phase of the software project, the actual developing, deployment, maintenance and ongoing development. The thesis does not include the actual development project of Pharmac's new system, but it identified some further development needs for the planned model.

Keywords: Dose dispensing, Healthcare and social welfare systems, Agile methods, Software development

# Sisällys

## Lyhenteet ja termit

1	Johdanto	1
2	Insinööriyö	2
2.1	Pharmacin kehitysprojekti	3
2.2	Menetelmät	5
2.3	Aineisto	6
2.4	Vaiheistus	7
3	Tiedonkeruu	8
3.1	Regulaatio	8
3.1.1	Lääkkeiden koneellinen annosjakelu	9
3.1.2	Lääketykku	12
3.1.3	Lääkkeiden hinnoittelu	13
3.1.4	Muut palvelut	14
3.1.5	Sosiaali- ja terveydenhuollon tietojärjestelmät	15
3.1.6	Regulaatio tulevaisuudessa	18
3.2	Tietojärjestelmien kehittäminen	19
3.2.1	Vesiputousmalli	19
3.2.2	Ketterät menetelmät	21
3.2.3	Vesiputousmalli vs. ketterät menetelmät	23
3.3	Toiminnalliset vaatimukset	25
3.4	Työpajat	28
3.4.1	Tietosuojavastaava	29
3.4.2	IT-arkkitehti	30
3.4.3	Product Owner	31
3.4.4	Hankepäällikkö	33
4	Ehdotus kehitysmallista	34
4.1	Alkuvaiheen suunnittelu	35
4.1.1	Vaatimusmäärittely	36
4.1.2	Tiekartta	37
4.1.3	Arkkitehtuurisuunnitelma	38
4.2	Toteutus	41

4.2.1	Modulaarinen ja uudelleenkäytettävä toteutus	42
4.2.2	Scrum-käytänteet	44
4.2.3	MVP	47
4.2.4	Testaus	48
4.3	Käyttöönotto	50
4.4	Ylläpito ja kehitys	51
5	Loppusanat	51
	Lähteet	53
	Liitteet	
	Liite 1: Työpajan runko – Tietosuojavastaava	
	Liite 2: Työpajan runko – IT-arkkitehti	
	Liite 3: Työpajan runko – Product Owner	
	Liite 4: Työpajan runko – Hankepäällikkö	

## Lyhenteet ja termit

Annosjakelu:	Palvelu, jossa apteekki toimittaa loppukäyttäjälle hänen säännöllisesti käyttämänsä lääkkeet ja muut valmisteet (esim. ravintolisät ja vitamiinit) valmiiksi annosteltuina. Usein puhutaan myös koneellisesta annosjakelusta tai lääkkeiden koneellisesta annosjakelusta.
DevOps:	<i>Development and Operations</i> on viitekehys ohjelmistokehityksen, joka keskittyy ohjelmistokehityksen (Dev) ja IT-ylläpidon (Ops) tiiviiseen yhteistyöhön ja integraatioon.
EMA:	<i>European Medicines Agency</i> , Euroopan lääkevirasto vastaa ihmisille ja eläimille tarkoitettujen lääkevalmisteiden tieteellisestä arvioinnista, valvonnasta ja turvallisuusseurannasta EU:n alueella.
Fimea:	<i>Lääkealan turvallisuus- ja kehittämiskeskus</i> Fimea on sosiaali- ja terveysministeriön alainen virasto, jonka tehtävänä on valvoa ja kehittää lääkealaa Suomessa.
GDP:	<i>Good Distribution Practice</i> . Lääkkeiden hyvät jakelutavat ovat kokoelma vaatimuksia lääkkeiden toimitusketjuun, eli käsittelyyn, säilytykseen ja toimitukseen.
GDPR:	<i>General Data Protection Regulation</i> on Euroopan unionin asetus, jonka tarkoituksena on varmistaa, että yritykset käsittelevät henkilötietoja laillisin perustein ja varmistavat henkilötietojen turvallisuuden, tietosuojan ja salassapidon.

- GMP: *Good Manufacturing Practice*. Lääkkeiden hyvät tuotantotavat ovat kokoelma vaatimuksista lääkkeiden valmistukseen.
- Hila: *Lääkkeiden hintalautakunta* on Suomen viranomainen, joka vastaa lääkkeiden, kliinisten ravintovalmisteiden ja perusvoiteiden hinta- ja korvauskäytäntöjen säätelystä. Hila määrittää, mitkä tuotteet ovat Kela-korvattavia ja asettaa näille tuotteille enimmäishinnat.
- Kanta: Kanta-palvelut ovat Suomen sosiaali- ja terveydenhuollon kansallinen sähköinen palvelukokonaisuus, jota ylläpitää Kela. Nämä palvelut kattavat sähköisen reseptin, potilastiedon arkiston, Omakannan sekä muut sähköisen asiointin palvelut.
- Kela: *Kansaneläkelaitos* vastaa Suomessa sairausvakuutuksen hallinnoinnista, mukaan lukien lääkekorvausten maksamisesta. Kela ylläpitää Kanta-palveluja, jotka ovat kansallisia terveydenhuollon tietojärjestelmiä. Kela varmistaa, että terveydenhuollon tietojärjestelmät ovat yhteensopivia Kanta-palveluiden kanssa
- MVP: *Minimum Viable Product*. Pienin elinkelpoinen/julkaistava tuote on IT-terminä käsitettynä kehitettävän järjestelmän pienin mahdollinen, pakolliset vaatimukset täyttävä versio.
- STM: *Sosiaali- ja terveystieteiden ministeriö* vastaa kansallisesti lääkepolitiikan suunnittelusta ja toteutuksesta. STM kehittää lainsäädäntöä ja politiikkoja, jotka ohjaavat lääkkeiden käyttöä, myyntiä ja valvontaa Suomessa. Se vastaa myös lääkkeiden hinta- ja korvausjärjestelmien kehittämisestä.

THL: *Terveyden ja hyvinvoinnin laitos* on Suomen valtion tutkimus- ja kehittämislaitos, joka toimii STM:n alaisuudessa. THL:n tehtävä on edistää kansanterveyttä ja hyvinvointia sekä kehittää sosiaali- ja terveystalvveluita.

Valvira: *Sosiaali- ja terveysalan lupa- ja valvontavirasto* Valvira valvoo sosiaali- ja terveydenhuollon, varhaiskasvatuksen, alkoholielinkeimon sekä ympäristöterveydenhuollon asianmukaista toimintaa.



## 1 Johdanto

Insinööriyö on toteutettu osana Metropolia Ammattikorkeakoulun tieto- ja viestintätekniikan insinööritutkintoa hyvinvointi- ja terveysteknologian tutkinto-ohjelmassa. Työn tilaajana on suomalainen lääketeollisuusalan yritys Pharmac Finland Oy (myöhemmin Pharmac).

Pharmac pääliiketoimintana on lääkkeiden koneellinen annosjakelu, mutta yritys tuottaa ja tarjoaa myös erilaisia lääkitysturvallisuuspalveluita ja digitaalisia palveluita sosiaali-, terveydenhoito- ja lääkealalle. Pharmac on perustettu vuonna 2008 ja sen liikevaihto oli vuonna 2023 noin 30,4 M€. Yritys työllistää noin 120 eri ammattikunnan työntekijää, joista suurin osa on farmasian ammattilaisia. Lääkkeiden koneellista annosjakelua tuottaa Suomessa neljä yritystä ja Pharmacilla on yli 40 %:n markkinaosuus loppukäyttäjistä. Pharmac tuottaa annosjakelupalvelua yli 400 apteekille ja n. 42 000 loppukäyttäjälle. [1.] Kuvassa 1 on esitelty koneellisessa annosjakelussa tuotettava annosjakelurulla.



Kuva 1. Pharmacin tuottama annosjakelurulla (Pharmac Finland Oy)

Lääkkeiden koneellisessa annosjakelussa loppukäyttäjä saa hänelle lääkärin määräämät lääkkeet annosteltuina tiettyyn päivämäärään ja kellonaikaan

otettaviin annospusseihin. Julkisen ja yksityisen puolen terveydenhuollon toimijat tilaavat annosjakelupalvelun valitsemansa apteekin kautta kuten muutkin loppukäyttäjälle määrätyt lääkkeet. Myös yksittäisen loppukäyttäjän on mahdollista tilata annosjakelupalvelu itselleen apteekin kautta.

Insinööriyössä on tavoitteena rakentaa Pharmacille malli ketterään ohjelmistokehitykseen, jossa pystytään ottamaan huomioon erilaiset regulaatiovaatimukset, jotka ohjaavat Pharmacin toimintaa sekä toisaalta terveydenhuollon tietojärjestelmien kehitystä. Insinööriyö linkittyy oleellisesti kehitteillä olevaan järjestelmään ja siinä käytettyihin toimintamalleihin Pharmacilla. Tässä työssä ei kuitenkaan toteuteta itse järjestelmäprojektia.

## 2 Insinööriyö

Perinteisesti sosiaali- ja terveydenhuollon tietojärjestelmien kehitys on tehty vesiputousmallin mukaisesti muun muassa sen takia, että kehitettävät järjestelmät ovat suhteellisen suuria ja kompleksisia kokonaisuuksia. Pharmacilla järjestelmien kehitys haluttiin kuitenkin tehdä ketterillä menetelmillä, joka mahdollistaisi lopputuotoksen mahdollisimman hyvän vastaavuuden liiketoimintatarpeisiin sekä mahdollisen MVP-version (Minimum Viable Product, pienin julkaisukelpoinen tuote) käyttöönoton. Myös mahdolliset regulaatiomuutokset, jopa projektin aikaiset, haluttiin ottaa huomioon ketterämmin järjestelmän kehitystyössä.

Insinööriyössä on tavoitteena rakentaa Pharmacille malli ketterään ohjelmistokehitykseen, jossa pystytään ottamaan huomioon erilaiset regulaatiovaatimukset, jotka ohjaavat Pharmacin toimintaa sekä terveydenhuollon tietojärjestelmiin liittyvät regulaatiovaatimukset. Vaatimukset koostuvat pääosin Lääkelaista [2] johdetuista määräyksistä ja suosituksista, Sosiaali- ja terveystieteiden lupa- ja valvontavirasto Valviran määrittelemistä vaatimuksista sosiaali- ja terveydenhuollon tietojärjestelmille [3], THL:n määräyksistä ja ohjeista sote-alan tiedonhallinnasta [4], sosiaali- ja terveydenhuollon asiakastietojen sähköisen käsittelyn laista [5] sekä EU:n yleisestä tietosuojasetuksesta (GDPR, General Data Protection Regulation) [6]. On tunnistettu, että koneelliseen annosjakeluun

sovellettavat lait tai lain soveltamisohjeet voivat muuttua lähitulevaisuudessa huomattavasti.

Insinööriyön tavoitteeksi kiteytyi ”Luoda Pharmacille ketterän ohjelmistokehityksen malli ottaen huomioon regulaatio ja sen mahdollinen muuttuminen tulevaisuudessa”. Lopputuotostavoitteena on mahdollisimman konkreettinen kokoelma työkaluja Pharmacille ketterien projektien suunnitteluun, johtamiseen ja toteuttamiseen. Työkalujen suunnittelun ja kokoamisen mahdollistamiseksi tulisi perehtyä Pharmacin toimintaan ja toimintaa ohjaavaan regulaatioon, Pharmacin liiketoimintatarpeisiin, sovelluskehitysmalleihin ja käynnissä olevaan järjestelmäprojektiin.

## 2.1 Pharmacin kehitysprojekti

Pharmac on rakentamassa käyttöönsä uutta tilaus-, toimitus- ja laskutusjärjestelmää sekä siihen liittyvää itsepalvelua apteekkeille ja muille Pharmacin asiakkaille. Osa järjestelmien vaatimusmäärittelystä on tehty syksyn ja talven 2021 aikana, jonka jälkeen kartoitettiin muun muassa erilaisista valmiista ERP-järjestelmistä (Enterprise Resource Planning, toiminnanohjausjärjestelmä) Pharmacille soveltuvaa ratkaisua. Valmisohjelmistoista ei kuitenkaan löytynyt Pharmacin tarpeisiin soveltuvaa tuotetta.

Tämän jälkeen kartoitettiin Euroopan tasolla muiden annosjakelutoimijoiden käyttämiä järjestelmiä haastatteleamalla näitä toimijoita. Millään toimijalla ei ollut käytössä valmisohjelmistoa, vaan kaikkien käyttämät järjestelmät olivat enemmän tai vähemmän heille rakennettuja ratkaisuja. Pienemmät toimivat käyttivät muun muassa Exceliä ja erilaisia suoria tiedostonsiirtoratkaisuja annosjakelun tuotannonohjaukseen. Valtaosa järjestelmistä alkoi olla elinkaarensa päässä, tai järjestelmissä oli tunnistettu todella suuria kehityskohteita.

Kartoitusten jälkeen järjestelmää päätettiin lähteä tekemään kokonaan uutena järjestelmänä ns. pitkistä tavarasta. Pharmacilla on itsellään pieni ohjelmistokehitystiimi, mutta koska rakennettava järjestelmä tulisi olemaan niin iso projekti,

päädyttiin hakemaan toteutustyöhön Pharmacin sisäisen tiimin lisäksi ulkoinen kumppani.

Kuvassa 2 näkyy uuden järjestelmäkokonaisuuden liiketoiminta-arkkitehtuuri ylätasolla. Toteutettava kokonaisuus sisältää myös integraatioalustan, jonka avulla haetaan ja lähetetään dataa liittyviin järjestelmiin. Osa integroitavista järjestelmistä on Pharmacin omia ja osa ulkoisten toimijoiden järjestelmiä.



Kuva 2: Ylätason arkkitehtuurikuva uudesta järjestelmästä

Pharmacille saapuvat annosjakelutilaukset tulevat suoraan apteekkijärjestelmästä Pharmacin toteuttamaa rajapintaa pitkin. Lääkkeiden hinnat ja lisätiedot tuodaan muutamasta erillisestä ulkoisesta lähteestä. Koska pääosa saapuvista tilauksista käsitellään samana ja viimeistään seuraavana päivänä, on tuotannon suunnittelu ja -ohjaus oleellinen osa uutta järjestelmää. Järjestelmä tullaan integroimaan kahdensuuntaisesti Pharmacin puhdastiloissa käyttämiin annosjaku- ja kuvauslaitteisiin.

Integraatioalustan kautta välitetään tietoa ulospäin Pharmacin sisäisiin tarpeisiin mm. laskutuksen, raportoinnin ja kirjanpidon osalta. Tuotannonohjauksen tarpeisiin välitetään reaaliaikaisesti dataa erilaisiin tuotannonohjausnäyttöihin.

Dataa tarjotaan ulospäin myös muun muassa Pharmacin Extranet-portaaliin ja mobiilisovellukseen sekä Pharmacin asiakkaiden omiin palveluihin.

## 2.2 Menetelmät

Työn tyyppi on kehitysprojekti, joka jakautuu tiedon keräämiseen ja analysointiin, ehdotukseen ketterän ohjelmistokehityksen suuntaviivoista Pharmacille ja tunnistettuihin jatkokehitysaihioihin. Kuvassa 3 on esitelty työn ositus ylätason kokonaisuuksineen ja kustakin osiosta valmistuva lopputuotos.

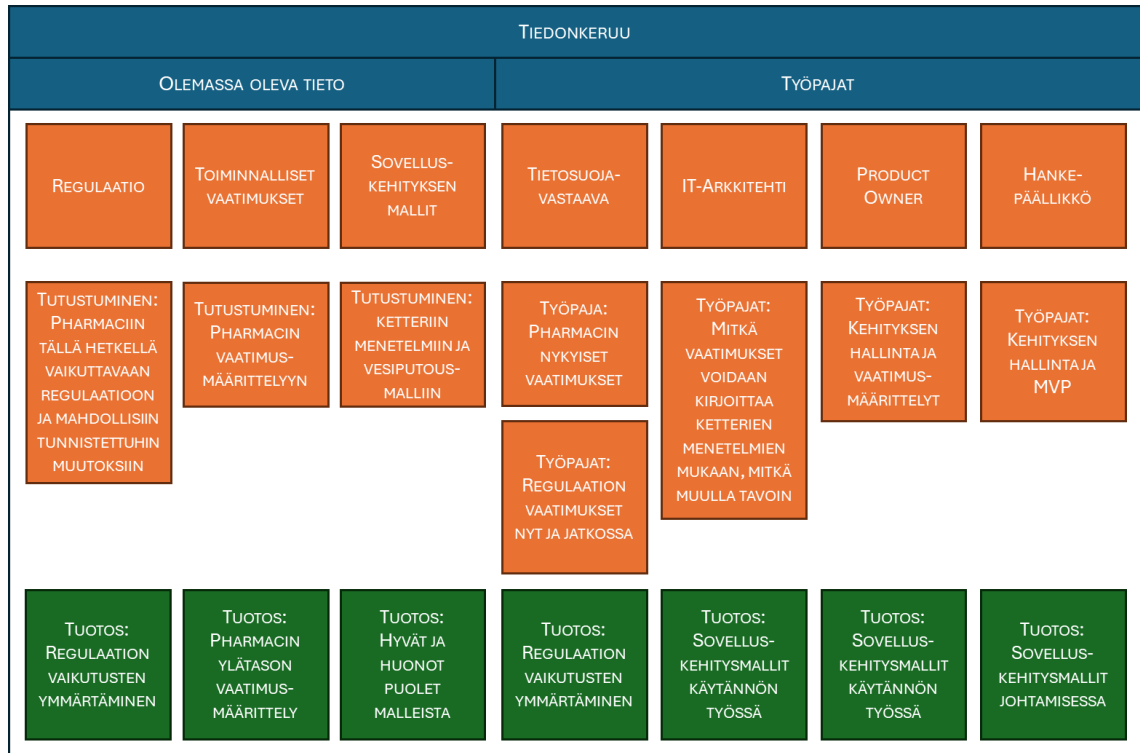
OPINNÄYTETYÖ						
SUUNNITELMA	TIEDONKERUU		EHDOTUS	ESITYS	JATKOTARPEET	DOKUMENTOINTI
KEHITYS- PROJEKTIN SUUNNITTELU	OLEMASSA OLEVA TIETO	TYÖPAJAT	OHJELMISTO- KEHITYSMALLIN SUUNNITTELU	MALLIN ESITTELY PHARMACILLE	JATKOKEHITYS- TARPEIDEN TUNNISTAMINEN LÄPIKÄYNNISTÄ	DOKUMENTOINTI OPINNÄYTE- TYÖKSI
OPINNÄYTETYÖN SUUNNITTELU	REGULAATIOON, VAATIMUKSIIN JA KEHITYSMALLEIHIN TUTUSTUMINEN	TYÖPAJAT ERI VASTUU- HENKILÖIDEN KANSSA	MALLIN SUUNNITTELU TIEDONKERUUN POHJALTA			
TUOTOS: PROJEKTI- SUUNNITELMA	TUOTOS: REGULAATIO, VAATIMUKSET JA KEHITYSMALLIT KOOSTETTUNA	TUOTOS: KOOSTEET WORKSHOPEISTA	TUOTOS: OHJELMISTO- KEHITYSMALLI DOKUMENTOITU	TUOTOS: ESITTELY- TILAISUUS PIDETTY	TUOTOS: JATKOKEHITYS- TARPEET DOKUMENTOITU	TUOTOS: OPINNÄYTETYÖ

Kuva 3. Työn vaiheet

Työ alkaa suunnittelulla Pharmacin kehitysprojektille sekä sen ohessa toteutettavalle insinööriyölle. Tämän jälkeen tulee tiedonkeruuvaihe, joka jakautuu olemassa olevan tiedon keräämiseen ja työpajoilla kerättävään uuteen tietoon. Tiedonkeruun jälkeen seuraa Pharmacille soveltuvan ohjelmistokehitysmallin suunnittelu aiemman vaiheiden tuotosten analysoinnin pohjalta. Ohjelmistokehitysmallin ehdotus esitetään seuraavassa vaiheessa Pharmacin edustajille, jonka pohjalta mallia voidaan vielä hienosäätää. Läpikäynnissä mahdollisesti nousevat jatkokehitystarpeet listataan erilliseksi dokumentiksi seuraavassa vaiheessa. Lopuksi kehitysprojekti dokumentoidaan osaksi insinööriyötä.

## 2.3 Aineisto

Tiedonkeruu työssä jakaantuu kahteen osioon, olemassa olevan tiedon keräämiseen eri lähteistä ja työpajoilla tehtävään tiedonkeruuseen Pharmacilla. Kuvassa 4 on havainnollistettu osiot, niiden vaiheet ja lopputuotokset.



Kuva 4. Tiedonkeruun kuvaus

Olemassa olevan tiedon keräys tapahtuu tutustumalla Pharmacin toimintaan ja terveydenhuollon tietojärjestelmien kehitykseen liittyvään regulaatioon, Pharmacin toiminnallisiin vaatimuksiin käynnissä olevasta järjestelmäprojektista sekä sovelluskehityksen perinteiseen malliin ja ketteriin menetelmiin. Ensimmäinen osio koostuu Pharmacin toimintaan ja terveydenhuollon tietojärjestelmiin liittyvän regulaation kartoituksesta. Tuotoksena tästä osiosta on ylätason dokumentti siitä, mitkä kaikki lait, asetukset, määräykset ja ohjeet vaikuttavat Pharmacin toimintaa ja IT-järjestelmien kehitystä. Pharmacin kehitettävän järjestelmän osalta on olemassa vaatimusmäärittelylista, jota on käytetty aiemmin mm. valmisohjelmien kartoitukseen. Tiedonkeruussa on tarkoitus tutustua vaatimusmäärittelydokumentteihin ja luoda niistä ylätason tarpeet, jotka auttavat

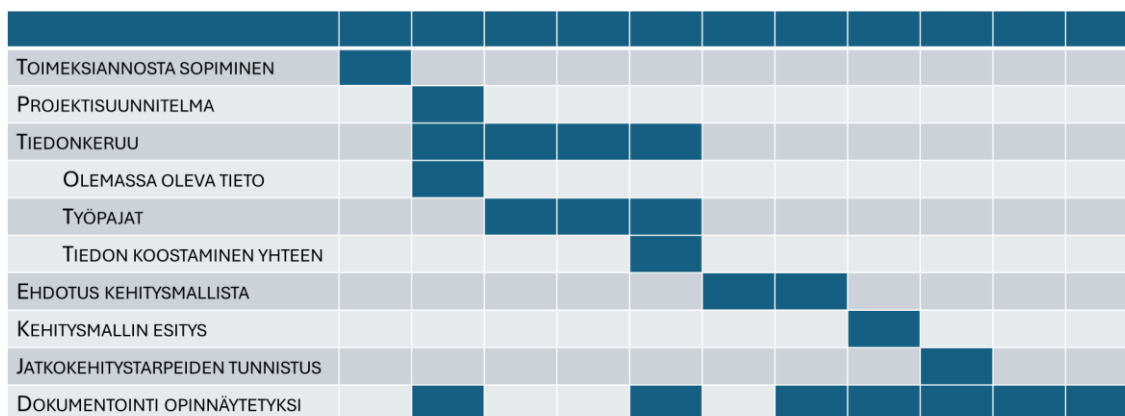
ohjelmistokehitysmallin suunnittelussa. Viimeisenä osiona olemassa olevan tiedon keräyksessä on tutustuminen sovelluskehityksen vesiputousmalliin ja ketteriin menetelmiin. Näistä tuotoksena on ylätason kuvaus molemmista sekä analyysi molempien mallien hyvistä ja huonoista puolista.

Pharmacilta ja Pharmacin alihankkijalta hankittavan tiedon keräämiseen valikoitui kvalitatiivinen tutkimusmenetelmä. Kvalitatiivinen menetelmä valittiin, koska haastateltavia ihmisiä tulisi olemaan suhteellisen pieni määrä ja jokaisella haastateltavalla on hieman oma erikoisalueensa. Käytännössä jokaisella haastateltavalle jouduttaisiin varioimaan hieman oma kyselypohja kvalitatiiviseen menetelmään, jolloin yksittäisten vastausten vertailu keskenään ei olisi järkevää eikä oikein mahdollistakaan. Kvalitatiivinen menetelmä mahdollistaa myös syvällisemmän ymmärryksen luomisen ja tarvittaessa tutkimuskysymysten muokkaamisen joustavammin tiedonkeruun toteutuksen yhteydessä.

Tarkemmaksi tiedonkeruumalliksi valikoitui tietojen kerääminen työpajoissa. Työpajat on teemoitettu eri asiantuntijaroolien mukaan, joita ovat tietosuojavastaava, IT-arkkitehti, Product Owner ja hankepäällikkö. Koska työpajoissa käsitellään tarpeita, vaatimuksia ja ohjelmistokehitystä aina hieman eri näkökulmasta, on työpajoissa aina ko. teemahenkilön lisäksi mukana muitakin asiantuntijoita. Tavoitteena työpajoille on, että ko. alueen erityisasiantuntija tuo työpajassa esiin oman näkemyksensä, mutta sitä voidaan jalostaa, rikastaa ja tarvittaessa haastaa yhdessä muiden asiantuntijoiden kanssa. Työpajat tulevat keskittymään tiedonkeräämiseen Pharmacilla käynnissä olevan järjestelmän näkökulmasta, jotta lopputuotokset olisivat mahdollisimman konkreettisia.

## 2.4 Vaiheistus

Aiemmin kuvassa 3 esitelty projektin ositus määrittelee suurimmalta osin projektin kulun ja sen, missä järjestyksessä kukin osio tulisi suorittaa. Kuvassa 5 on esitelty projektin vaiheistus Gantt-kaavioon sovitettuna.



Kuva 5. Projektisuunnitelman Gantt-kaavio

Projekti alkaa toimeksiannosta sopimisesta ja päättyy valmiiseen insinööriyöhön. Tiedonkeruu alkaa samanaikaisesti projektisuunnitelman tekemisellä, jonka jälkeen seuraa kehitysehdotusmallin työstö. Kehitysmalli esitellään yrityksen edustajille ja kirjataan tämän pohjalta tunnistetut suurimmat jatkokehitystarpeet. Insinööriyön dokumentointi etenee koko projektin muun tekemisen rinnalla keskittyen kuitenkin isolta osin projektin loppuvaiheeseen.

### 3 Tiedonkeruu

Tiedonkeruu jakaantui projektissa olemassa olevaan ja työpajoilla kerättävään uuteen tietoon. Olemassa olevan tiedon kerääminen koostui regulaation kartoittamisesta, kehitteillä olevan järjestelmän vaatimuksiin tutustumisesta ja sovel-luskehitysmalleihin perehtymisestä. Työpajoissa oli neljän eri roolin mukaisesti toteutettavia teemoja, mutta määrällisesti työpajoja pidettiin enemmän.

#### 3.1 Regulaatio

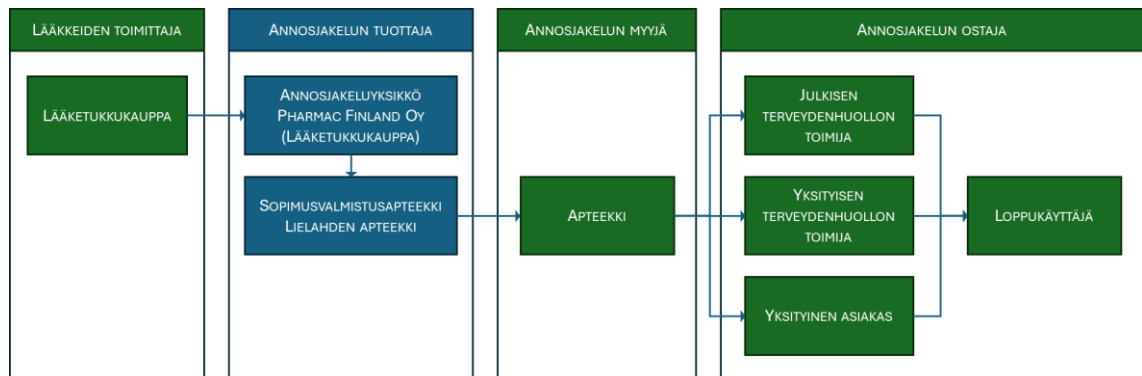
Olemassa olevan tiedonkeruun yhtenä aiheena on regulaatio, jonka osalta tähän lukuun on koostettu ja esitelty Pharmacin toimintaa keskeisimmin ohjaava sääätely. Toiminta on vahvasti reguloitua ja sitä ohjaavat mm. erilaiset lait, asetukset ja määräykset. Lainsäädännön ja sääätelyn näkökulmasta kokonaisuus jakautuu muutamaaan suurempaan kokonaisuuteen. Isointa osaa toimintaa ohjaa



lääkkeiden koneellisen annosjakelun säätely, mutta regulaation piirissä ovat mm. myös lääketukun toiminta, lääkkeiden hinnoittelu ja yksityisen terveydenhuollon toimijan lupa.

### 3.1.1 Lääkkeiden koneellinen annosjakelu

Lääkkeiden koneelliseen annosjakeluun liittyy usea eri taho ja jokaisella taholla on regulaation mukainen roolinsa. Kuvassa 6 on esitelty Pharmacin tuottaman annosjakelupalvelun toteuttamiseen liittyvät toimijat ja lääkkeiden kulku lääkettukukaupasta lääkkeen loppukäyttäjälle saakka.



Kuva 6. Annosjakelupalvelu

Annosjakelupalvelun ostajana voivat toimia julkisen tai yksityisen terveydenhuollon toimijat tai yksityinen henkilöasiakas. Lääkkeen loppukäyttäjä on aina yksityinen henkilö. Julkisen ja yksityisen terveydenhuollon toimijat voivat kilpailuttaa tarjoamansa annosjakelupalvelun paikallisesti tai valtakunnallisesti. Kilpailuun voivat osallistua annosjakelupalvelun myyjät eli apteekit. Myyjä apteekki voi tuottaa annosjakelun itse tai ostaa sen haluamaltaan sopimusvalmistusapteekilta. Sopimusvalmistusapteekki on vastuussa annosjakelupalvelun tuottamisesta, mutta se voi käyttää sen tuottamiseen alihankkijana erillistä annosjakeluyksikköä. [7.] Pharmac tuottaa koneellista annosjakelua Lielahden apteekille toimien sen annosjakeluyksikkönä. Annosjakeluun käytettävät lääkkeet Pharmac ostaa lääketukuilta, jolloin Pharmacilla on oltava tähän toimintaan lääketukukaupan toimilupa.

Lääkkeiden koneellista annosjakelua säätelevät Suomessa Lääkelaki ja siihen liittyvät asetukset sekä Fimean antamat ohjeet. Fimea toimii apteekissa ja sairaala-apteekissa tapahtuvan koneellisen annosjakelun sekä sopimusvalmistuksen luvan myöntäjänä ja valvovana viranomaisena. Lääkelain 12a §:n mukaan annosjakelutoimintaa voi harjoittaa vain apteekissa tai sairaala-apteekissa. [2.]

Apteekissa ja sairaala-apteekissa tapahtuva koneellinen annosjakelu sekä sopimusvalmistus edellyttävät Lääkealan turvallisuus- ja kehittämiskeskuksen lupaa. Lupa on myönnettävä, jos toiminta täyttää 15 §:ssä säädetyt vaatimukset. Lupa voidaan liittää lääkkeen valmistusta, luovutusta ja käyttöä koskevia sekä muita lääketurvallisuuden edellyttämiä ehtoja.

Lääkelain 15 §:ssä säädetään yltäosalla koneellisen annosjakelun edellytykset, mutta lupaan liittyvät tarkemmat säännökset jätetään määrittelemättä, joka mahdollistaa valtioneuvostolle ja Fimealle tarpeiden mukaisen säännösten muokkaamisen. [2.]

Apteekkien, sivuapteekkien, sairaala-apteekkien ja lääkekeskusten lääkevalmistuksen ja koneellisen annosjakelun edellytyksenä on, että henkilökunta on riittävästi perehtynyt lääkkeiden valmistukseen ja että siihen tarvittavat tuotantotilat ja laitteet ovat asianmukaiset. Toiminnassa tulee lisäksi noudattaa soveltuvin osin 11 §:n mukaisia lääkkeiden hyviä tuotantotapoja.

Tarkempia säännöksiä lupaedellytyksistä voidaan antaa valtioneuvoston asetuksella. Lääkealan turvallisuus- ja kehittämiskeskus voi antaa määräyksiä 12 a §:n 1 momentin mukaisen luvan hakemisesta.

Apteekki voi tuottaa annosjakelupalvelun itse tai hankkia sen erilliseltä sopimusvalmistusapteekilta. Mikäli apteekki hankkii annosjakelun sopimusvalmistusapteekilta, tulkitaan se lainsäädännöllisesti aina sopimusvalmistukseksi. Apteekin tulee tässä tapauksessa tehdä ilmoitus Fimealle Lääkelain 12§: toisen kappaaleen mukaisesti. [2.]

Apteekkarin voi kuitenkin valmistuttaa yksittäisiä lääkevalmisteita toisessa apteekissa ja hankkia sieltä yksittäisiä 17 §:n 1 momentin 3 kohdan mukaisesti maahan tuotuja raaka-aineita omaa valmistusta varten. Apteekkarin on tehtävä tästä ilmoitus Lääkealan

turvallisuus- ja kehittämiskeskukseksi. Keskus voi antaa määräyksiä apteekin lääkevalmistuksessa ja sopimusvalmistuksessa noudatettavista menettelytavoista sekä tarkempia määräyksiä ilmoitusmenettelystä.

Fimean määräys 06/2011 määrittelee tarkemmat vaatimukset edellä mainittuun lääkkeiden sopimusvalmistukseen. Määräys koskee apteekkien omien lääkevalmistusten valmistusta, joka ei ole saman ennakkovalvonnan alaista kuin teollisesti valmistetuissa lääkevalmisteissa, mutta jonka tulee täyttää tietyt turvallisuus- ja laatuvaatimukset. [8.]

Määräyksessä yksi keskeinen osa on laadunvarmistus ja -valvonta. Laadunvarmistus kattaa mm. käytetyt tilat ja laitteet, henkilökunnan koulutuksen sekä dokumentaation. Laadunvalvonta taas käsittelee raaka-aineiden, lääkevalmistusten ja pakkausmateriaalien valvontaa ja hyväksyntää. Määräys sisältää myös ohjeistukset tuotevirheiden käsittelyyn ja ilmoittamiseen sekä suhteen muihin säädöksiin, kuten työturvallisuus-, kemikaali- ja jätelakiin. [8.]

Määräys käsittää sopimusvalmistusta koskevat määräykset, mukaan lukien laatuseurannat sekä valmistuttajan ja valmistajan vastuut. Käytännössä siis myös sopimusvalmistusapteekin tulee hakea Fimealta lupa annosjakelutoimintaan. Annosjakelutoiminta toimii juridisesti aina sopimusvalmistusapteekin yhteydessä, mutta eri luvalla ja eri yhtiössä kuin kyseisen apteekin ns. normaali apteekkitoiminta. Mikäli sopimusvalmistusapteekki käyttää annosjakelun tuottamiseen erillistä annosjakeluyksikköä, vastaa sopimusvalmistusapteekki ja kyseisen apteekin apteekkari annosjakeluyksikön toiminnan laadusta. [8.]

Pharmacin tapauksessa sopimusvalmistusapteekkina toimii Lielahden apteekki, joka myy palvelua edelleen yli 400 apteekille. Pharmac toimii Lielahden apteekin annosjakeluyksikkönä kuvan 6 mukaisesti. Pharmacin tuottaman annosjakelupalvelun laadusta ja sitä kautta mm. laatuvarmistuksesta vastaa juridisesti sopimusvalmistusapteekki.

Edellä mainittujen lisäksi Fimean määräys 06/2011 määrää sopimusvalmistuksessa noudatettavan Euroopan komission GMP-ohjeita laajamittaisessa

lääkevalmistuksessa. Euroopan lääkevirasto European Medicines Agency (EMA) määrittelee hyvän valmistustavan Good Manufacturing Practice (GMP) vähimmäisstandardiksi, jonka lääkevalmistajien on täytettävä tuotantoprosesseissaan. EMA koordinoi tarkastuksia varmistamaan näiden standardien noudattamisen ja harmonisoi GMP-toimintoja EU-tasolla. GMP takaa, että lääkkeet ovat johdonmukaisesti korkealaatuisia, tarkoitukseensa sopivia ja täyttävät myyntiluvan vaatimukset [9].

### 3.1.2 Lääketukku

Suomessa on yleisenä käytäntönä, että annosjakeluyksikön yhteydessä toimii lääketukkukauppa, joka ostaa annosjakeluun käytetyt lääkkeet ja myy ne eteenpäin. Pharmacilla on lääketukkukaupan toimilupa ja lääkevalmisteet Pharmac myy tätä kautta Lielahden apteekille, joka myy ne eteenpäin asiakasapteekilleen.

Läkelain 35a §:ssä säädetään, että lääketukkukaupan toiminnassa tulee noudattaa lääkkeiden hyviä jakelutapoja. EU on julkaissut lääkkeiden hyvien jakelutapojen Good Distribution Practise (GDP) -ohjeiston. [2.]

Lääketukkukaupan toiminnassa tulee noudattaa Euroopan yhteisöjen säännöksiin perustuvia lääkkeiden hyviä jakelutapoja. Lääkealan turvallisuus- ja kehittämiskeskus voi antaa tarkempia määräyksiä lääketukkukaupassa noudatettavista lääkkeiden hyvistä jakelutavoista.

EU:n GDP-ohjeisto kattaa lääkkeiden jakeluprosessin laatuvaatimukset, tarkoituksenaan varmistaa lääkkeiden laadun ja eheyden säilyminen koko toimitusketjun ajan. Ohjeisto sisältää määräyksiä varastoinnista, kuljetuksesta, dokumentoinnista ja toimitusten seurannasta sekä korostaa riskienhallinnan merkitystä. Tavoitteena on taata, että loppukäyttäjät saavat turvallisia ja tehokkaita lääkkeitä. [10.]

### 3.1.3 Lääkkeiden hinnoittelu

Lääkkeiden hinnoittelu Suomessa perustuu kahteen tekijään, tukkuhintaan ja apteekin katteeseen (vähittäishintaan). Reseptilääkkeiden tukkuhinnan voi lääkeryitys määrittää vapaasti, mutta Kela-korvattavien lääkkeiden tukkuhinnan on oltava Lääkkeiden hintalautakunnan (Hila) vahvistaman enimmäishinnan rajoissa. Itsehoitolääkkeille on asetettu enimmäis- ja vähimmäishinta, ja apteekit määrittelevät niiden myyntihinnan näiden rajojen sisällä. Kaikille apteekkeille on myytävä lääkkeet samalla tukkuhinnalla hankintamäärästä riippumatta, eikä alennuksia saa antaa. [11.] Tukkuliikkeiden on mahdollista hankkia lääkkeitä alennetuilla hinnoilla toisilta tukuilta tai lääkevalmistajilta.

Hila on suomalainen viranomainen, jonka päätehtävänä on säädellä sairausvakuutuslain piiriin kuuluvien lääkkeiden, kliinisten ravintovalmisteiden ja perusvoiteiden korvattavuutta ja tukkuhintoja. Se pyrkii varmistamaan, että korvattavat lääkkeet ovat potilaille kohtuullisesti hinnoiteltuja samalla kun ylläpidetään lääkekorvausjärjestelmän taloudellista kestävyyttä. Hila tekee päätöksiä myös lääkkeiden korvausluokituksista ja viitehinnoista, mikä on olennaista lääkekorvausjärjestelmän hallinnoinnissa. [12.]

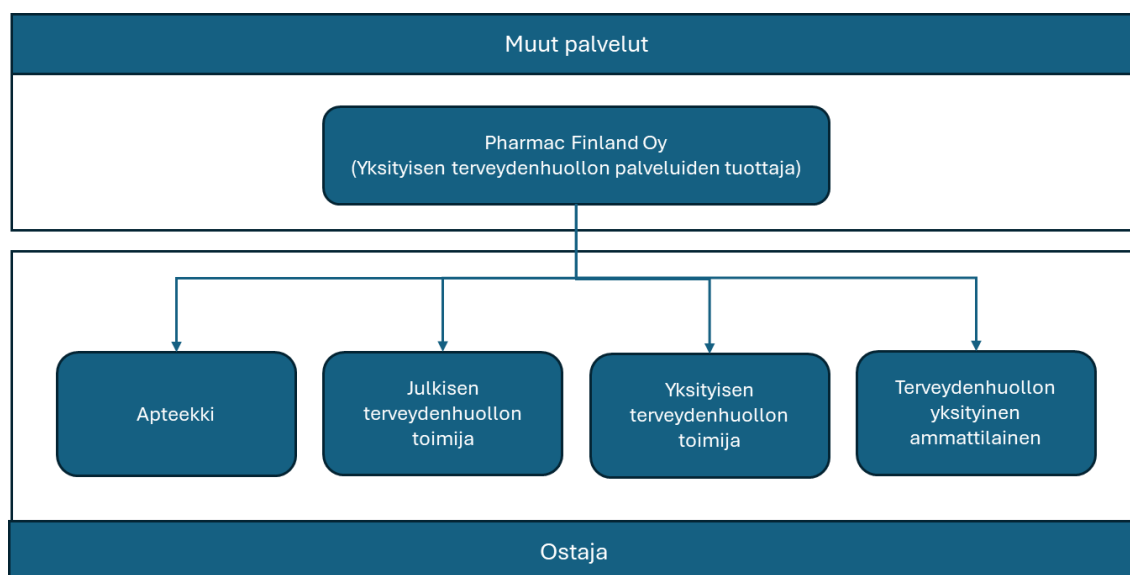
Läaketaksa on valtioneuvoston asettama hinnoittelujärjestelmä, joka määrittelee, kuinka apteekit hinnoittelevat myymänsä lääkkeet Suomessa. Tämä järjestelmä varmistaa, että reseptilääkkeiden hinta on sama kaikissa Suomen apteekkeissa. Läaketaksan mukaan apteekin myyntikate pienenee suhteellisesti, kun lääkkeen tukkuhinta nousee, mikä tekee hinnoittelusta degressiivisen. Reseptilääkkeiden ja itsehoitolääkkeiden hinnoittelussa käytetään erillisiä läaketaksa- taulukoita tai hinnoittelukaavoja. [13.] Läaketaksa päivittyy kahden viikon välein ja Pharmac muuttaa myymiensä lääkevalmisteiden hinnat taksahinnaston mukaiseksi.

Itsehoitolääkkeiden enimmäishinta on säädelty läaketaksalla, vaikka niiden hinnat voivat vaihdella apteekkeittain. Läaketaksan tavoitteena on pitää lääkehinnat kohtuullisina ja läpinäkyvinä kuluttajille. Apteekkien on noudatettava

lääketaksaa myydessään lääkkeitä, ja he eivät saa ottaa vastaan alennuksia lääkeyrityksiltä. Tämä säännöstö tukee lääkkeiden yhdenvertaista saatavuutta ja edistää potilaiden terveyttä. [13.]

### 3.1.4 Muut palvelut

Edellä kuvattujen lisäksi Pharmacilla on yksityisen terveydenhuollon palveluiden tuottajan lupa mm. Lääkehoidon arviointien (LHA) ja lääkehoidon kokonaisarvioiden (LHKA) tuottamiseen. Pharmac tuottaa myös erilaisia koulutuspalveluita terveydenhuollon tarpeisiin. Pharmacin muiden palveluiden ostajana voivat olla kuvan 7 mukaisesti apteekit, julkisen tai yksityisen terveydenhuollon toimijat tai yksittäiset terveydenhuollon ammattilaiset.



Kuva 7. Pharmacin muut palvelut

Yksityisiä terveydenhuollon palveluntarjoajia säännellään useiden lakien ja asetusten avulla, joiden tarkoituksena on, että toiminta on korkealaatuista, turvallista ja asianmukaisesti järjestettyä. Näiden säännösten ja lakien noudattamista valvovat eri viranomaiset, kuten Aluehallintovirastot (avi) ja Valvira.

Terveydenhuoltolaki (1326/2010) [14] määrittelee, että terveydenhuollon toiminnan tulee perustua tieteelliseen näyttöön ja hyviin hoitokäytäntöihin. Lisäksi on

olemassa Asetus yksityisestä terveydenhuollosta (744/1990) [15], joka sisältää säännökset yksityisten terveydenhuollon toimijoiden toiminnasta.

Yksityisiä terveydenhuollon palveluita säännellään myös Potilaan asemaa ja oikeuksia koskevalla lailla (785/1992), joka turvaa muun muassa potilaan oikeuden saada tietoa, itsemääräämisoikeuden sekä hoitoon liittyvät asiakirjat. Potilailla on oikeus laadukkaaseen ja yhdenvertaiseen hoitoon. [16.]

Lisäksi Laki terveydenhuollon ammattihenkilöistä (559/1994) varmistaa henkilöstön pätevyyden vaatimalla luvan saantia, rekisteröintiä ja ammatinharjoittamisen valvontaa. Laki määrittelee mm. sen, että palveluiden tuottajalla tulee olla lupaviranomaisen hyväksymä terveydenhuollon palveluista vastaava johtaja. Vastaavaksi johtajaksi hyväksytään palvelujen tuottajan kirjallisesta hakemuksesta henkilö, joka täyttää vastaavalle johtajalle asetuksessa säädetyt vaatimukset. [17.]

Yksityisen terveydenhuollon palveluntarjoajat ovat myös velvollisia noudattamaan sosiaali- ja terveydenhuollon asiakasmaksuista annettua lakia (734/1992), joka säätelee palveluista perittäviä maksuja, maksujen enimmäismääriä ja määrittää, mitkä palvelut ovat asiakkaille maksuttomia. [18.]

### 3.1.5 Sosiaali- ja terveydenhuollon tietojärjestelmät

Valvira määrittelee, hallinnoi ja valvoo sosiaali- ja terveydenhuollon tietojärjestelmiä asiakastietolain mukaisesti. Tietojärjestelmät luokitellaan niiden käyttötarkoituksen ja tietoturvan tarpeen perusteella eri luokkiin, jotka määrittävät myös niiden valvontaprosessit. Ylätasolla järjestelmät luokitellaan A- ja B-luokkiin. [3.]

Luokka A sisältää tietojärjestelmiä, jotka ovat yhteydessä Kanta-palveluihin. Näitä ovat esimerkiksi apteekkijärjestelmät, reseptien hallintajärjestelmät ja potilastietojärjestelmät, jotka tallentavat tietoja kansallisesti käytettäviin tietokantoihin. Luokka A jaetaan edelleen kolmeen alaluokkaan (A1, A2, A3) sen mukaan, kuinka kriittisiä järjestelmät ovat potilasturvallisuuden ja -tiedon kannalta.

Korkeamman luokan järjestelmät vaativat yleensä tiukempia tietoturvatavoimia ja läpäisevät Kansaneläkelaitoksen (Kela) järjestämän yhteistestauksen, mikä varmistaa niiden yhteentoimivuuden muiden järjestelmien kanssa. [19.]

Luokka B kattaa järjestelmät, jotka käsittelevät asiakas- ja potilastietoja mutta eivät liity suoraan Kanta-palveluihin eivätkä tuota Kanta-palveluihin tallennettavia asiakirjoja. Tällaiset järjestelmät eivät vaadi yhteistestausta, mutta niiden on täytettävä THL:n määrittelemät olennaiset tietoturva-vaatimukset. [19.]

Lisäksi on tietojärjestelmiä ja sovelluksia, jotka eivät täytä asiakastietolain määritelmiä, mutta jotka silti käsittelevät potilas- tai asiakastietoja hallinnollisissa tai tukitoiminnoissa. Nämä järjestelmät luokitellaan usein luokittelemattomiksi ohjelmistoiksi ja sovelluksiksi. [19.]

Asiakas- ja potilastietojen käsittelyyn tarkoitettun tietojärjestelmän pitää täyttää käyttötarkoituksensa mukaiset olennaiset vaatimukset. Olennaisten vaatimusten täyttämistä ja niiden ylläpidosta on vastuussa tietojärjestelmäpalvelun tuottaja. Olennaiset vaatimukset jakautuvat kolmeen osa-alueeseen: toiminnallisiin vaatimuksiin, yhteentoimivuuteen sekä tietoturvaan ja tietosuojaan. [3.]

Toiminnalliset vaatimukset sosiaali- ja terveydenhuollon tietojärjestelmissä viittaavat järjestelmän toimintoihin ja tietosisältöön, jotka perustuvat alaa säätelevään lainsäädäntöön, kuten lääke- ja potilaslakiin. Nämä vaatimukset on määriteltä tarkemmin Terveiden ja hyvinvoinnin laitoksen (THL) dokumentissa [20]. Tietojärjestelmän käyttötarkoitus määrittää, millaisia toimintoja ja tietoja siinä tulee olla. Tietojärjestelmäpalvelun tuottajan on kuvattava nämä toiminnot ja tietosisällöt järjestelmälomakkeella, joka toimitetaan eri tahojen (mm. Kela ja Valvira) käyttöön riippuen siitä, kuuluuko järjestelmä luokkaan A tai B. Järjestelmälomakkeen tiedot on pidettävä ajantasaisina ja tarkkoina. [3.]

Yhteentoimivuus tarkoittaa, että Kanta-palveluihin liitetyt tietojärjestelmät pystyvät tallentamaan ja jakamaan tietoa niin, että se on haettavissa ja näytettävissä myös muilla yhteensopivilla järjestelmissä. Kanta-palvelujen kautta tapahtuva tietojen vaihto edellyttää, että tietojärjestelmät ovat keskenään yhteentoimivia ja



toteutettu kansallisten määritysten mukaisesti. Yhteentoimivuus varmistetaan Kelan järjestämässä yhteistestauksessa, joka on pakollinen luokan A2 ja A3 tietojärjestelmille. Kela myös tarjoaa yhteistestauksesta lausunnon ja raportin ja itse testauspalvelu on maksuton. [3.]

Tietoturva sosiaali- ja terveydenhuollon tietojärjestelmissä keskittyy varmistamaan asiakas- ja potilastietojen luottamuksellisuuden, eheyden ja käytettävyyden. Nämä tietoturva-vaatimukset käsittävät kolme pääkohtaa, luottamuksellisuuden, eheyden ja käytettävyyden. Luottamuksellisuudella varmistetaan, että vain oikeutetut henkilöt pääsevät käsiksi tietoihin, esimerkiksi hoitosuhteen kautta. Eheyden osalta tietojen on oltava muuttumattomia ja vain valtuutetut voivat tehdä muutoksia varmistamalla esimerkiksi ammattihenkilön allekirjoituksella, että tiedot pysyvät ajantasaisina ja yhdenmukaisina eri järjestelmissä. Käytettävyys tarkoittaa, että tietojen täytyy olla helposti saatavilla terveydenhuollon tarpeisiin, esimerkiksi potilastietojen on oltava haettavissa Kanta-palveluista. [3.]

Valvira ylläpitää tietojärjestelmärekisteriä ja kaikkien tietojärjestelmien tuottajien tulee ilmoittaa järjestelmänsä viranomaiselle rekisteröitäväksi. Rekisteröinti on maksullista, ja kustannukset vaihtelevat järjestelmän luokan mukaan. Luokan A järjestelmän ensirekisteröinti maksaa 1200 euroa, kun taas luokan B rekisteröinti on 600 euroa. Rekisteröinnin yhteydessä Valvira tarkastaa, että tietojärjestelmä täyttää kaikki olennaiset vaatimukset, jotka on määritelty järjestelmän käyttötarkoitusta varten. [21.]

Pharmacin uusi järjestelmä ei kuulu nykyisessä luokittelussa Valviran sosiaali- ja terveydenhuollon tietojärjestelmiin, mutta on hyvin mahdollista, ellei jopa todennäköistä, että annosjakeluun sovellettavaa regulaatiota tullaan muuttamaan lähivuosien aikana. Fimean ylijohtaja Sinikka Rajaniemen tekemässä Lääkehoidon ohjauksen kehittäminen -selvityksessä [22] mainitaan mm. seuraavaa.

Lisäksi lääkehuollon tietojärjestelmiä (kuten lääkelogistiikan, varastoinnin, valmistuksen, koneellisen annosjakelun tietojärjestelmiä)

tulisi ohjata vastaavalla tavalla kuin kansallisesti ohjataan sosiaali- ja terveydenhuollon asiakas- ja potilastietojärjestelmiä.

Näin ollen Pharmacin järjestelmäprojektissa pyritään ottamaan mahdollisimman hyvin huomioon sekä nykyinen regulaatio, että mahdolliset muutokset tulevaisuudessa. Näiden vaatimusten huomioiminen ei kuitenkaan tarkoita, että kaikki toiminnallisuudet rakennettaisiin järjestelmään tässä vaiheessa vaan että järjestelmän rakennettaisiin niin, että mahdolliset muutokset pystyttäisiin toteuttamaan mahdollisimman vähäisellä ylimääräisellä työllä.

### 3.1.6 Regulaatio tulevaisuudessa

Lääkkeiden koneellinen annosjakelu on vahvasti reguloitua, mutta monilta osin regulaatio on tehty alun perin apteekkien pienimuotoista annosjakelua varten. Lakeja on myös tarkennettu erilaisilla säännöillä ja suosituksilla, joiden muuttaminen viranomaistahoilta voi tapahtua huomattavasti helpommin kuin itse lain muuttaminen.

On myös hyvin mahdollista, että itse lainsäädäntöönkin tulee muutoksia lähivuosina. Tästä esimerkkinä aiemmin mainittu ehdotus koneellisen annosjakelun tietojärjestelmien luokittelusta Valviran sosiaali- ja terveydenhuollon tietojärjestelmien mukaisesti. Sosiaali- ja terveysministeriön (STM) asettamissa työryhmissä on tuotu esiin lääkkeiden koneelliseen annosjakeluun liittyviä kehittämistarpeita jo useiden vuosien ajan. Suurimpia regulaation vaikuttavia tarpeita ovat:

- Koneellisen annosjakelupalveluyksikön perustaminen tulisi mahdollistaa myös muille toimijoille kuin apteekille ja sairaala-apteekille [23].
- Julkisen terveydenhuollon koneellisen annosjakelun toimivuutta parannetaan tietojärjestelmiä kehittämällä ja hoidon jatkuvuutta parantamalla [23].
- Annosjakelun valmistustoimintaa koskevaa lainsäädäntöä tulisi tarkentaa ja antaa siitä Fimean määräys (lääkelakia sen vuoksi uudistettava) [24].

- Pohdittava, voivatko yksityiset annosjakeluyksiköt tarjota palvelua julkisen puolen toimijoille [24].
- Koneellisen annosjakelupalvelun sisältö ja toimijoiden tehtävät tulisi määritellä lainsäädännössä, jotta eri toimijat saadaan sitoutumaan annosjakelun toimintamalliin [7].

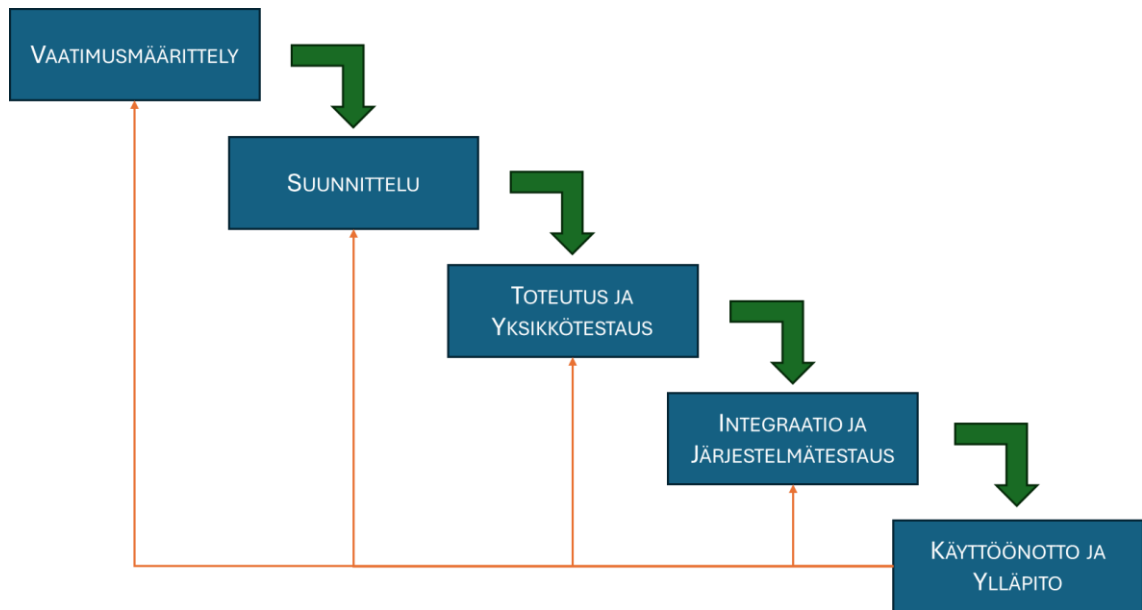
Edellä kuvattujen valossa on siis todennäköistä, että Pharmaciin vaikuttava regulaatio tulee jollain tavalla muuttumaan. Näin ollen nyt rakennettavassa järjestelmässä on hyvä ottaa huomioon se, että järjestelmää on pystyttävä muokkaamaan mahdollisimman ketterästi lähivuosien aikana.

## 3.2 Tietojärjestelmien kehittäminen

Toinen osio olemassa olevan tiedonkeruusta oli tutustua kahteen sovelluskehitysmalliin, perinteisempään vesiputousmalliin ja uudempiin ketteriin menetelmiin. Tässä kappaleessa on esitelty molemmat näistä. Tavoitteena oli kartoittaa yleistiedot molemmista malleista, tunnistaa niiden vahvuudet ja heikkoudet, eroavaisuudet toisistaan.

### 3.2.1 Vesiputousmalli

Vesiputousmalli on yksi vanhimmista ja perinteisimmistä ohjelmistokehityksen malleista, joka noudattaa järjestäytyneitä ja lineaarista lähestymistapaa projektihallintaan. Sen juuret ulottuvat 1970-luvulle ja se on saanut nimensä siitä, kuinka projektin vaiheet valuvat alas kuin vesiputous, yksi kerrallaan, ilman että aiempiin vaiheisiin palataan. Malli on jaettu selkeisiin vaiheisiin, jotka alkavat vaatimusmäärittelystä ja etenevät suunnittelun, toteutuksen ja yksikkötestauksen, integraation ja järjestelmätestauksen kautta käyttöönottoon ja ylläpitoon. [25]. Kuvassa 8 on kuvattu vesiputousmallin vaiheet ja niiden eteneminen lineaarisesti.



Kuva 8. Vesiputousmalli ja sen vaiheet.

Vesiputousmalli on suunniteltu niin, että jokaisen vaiheen tulokset on dokumentoitava ja hyväksyttävä ennen seuraavan vaiheen aloittamista. Vesiputousmalli edustaa suunnitelmavetoista prosessia, jossa kaikki prosessivaiheet on suunniteltava ja aikataulutettava ennen töiden aloittamista. [25.]

Vaatusmäärittelyssä määritellään kaikki järjestelmän ominaisuudet, rajoitteet/reunaehdot ja tavoitteet yhteistyössä järjestelmän käyttäjien kanssa. Vaatimukset kirjataan yksityiskohtaisesti, ja ne muodostavat koostettuna vaatimusmäärittelydokumentin. Järjestelmän suunnitteluvaiheessa tehdään käytännössä tekninen arkkitehtuurisuunnitelma sisältäen tietomallit ja mietitään käytettävät tekniikat. Toteutus ja yksikkötestaus -vaihe koostuu edellisissä vaiheissa määritettyjen ja suunniteltujen toiminnallisuuksien/ohjelmien toteuttamisesta ja näiden testaamisesta. Integrointi ja järjestelmätestaus kattaa toteutettujen toiminnallisuuksien ja ohjelmien liittämisen yhteen toimivaksi järjestelmäksi ja koko järjestelmän testaamisen. Käyttö ja ylläpito -vaiheessa järjestelmä asennetaan ja otetaan käyttöön. Ylläpito käsittää virheiden korjaamisen, joita ei havaittu elinkaaren aikaisemmissa vaiheissa, järjestelmäyksiköiden toteutuksen parantamisen ja järjestelmän palveluiden kehittämisen uusien vaatimusten ilmetessä. [25.]

Ajan myötä vesiputousmalli on kehittynyt ja mukautunut vastaamaan modernimman ohjelmistokehityksen haasteita. Esimerkiksi "modifioitu vesiputousmalli" tai "vesiputousmalli palautuksilla" sallii paluun aiempiin vaiheisiin, mikä tarjoaa lisää joustavuutta projektinhallintaan. Tämä evoluutio osoittaa, että vaikka perinteiset lähestymistavat voivat olla rajoittavia, niitä voidaan sopeuttaa ja yhdistää uudempiin menetelmiin vastaamaan nykypäivän ohjelmistokehityksen tarpeita. [26.] Kuvassa 8 on havainnollistettu ylätasolla mahdollisuus palata mallin aiempiin vaiheisiin.

Vesiputousmallin suurimpiin etuihin kuuluu sen yksinkertaisuus ja selkeys. Jokaisella vaiheella on selkeät toimitettavat tuotokset ja tarkastusprosessit, mikä tekee mallista helppotajuista ja hallittavaa. Mallin jäykkyys on kuitenkin myös sen haittapuoli. Se ei sovi hyvin projekteihin, joissa vaatimukset todennäköisesti muuttuvat tai joissa kehityksen aikana tarvitaan joustavuutta. Lisäksi, koska se perustuu yhden vaiheen valmistumiseen ennen seuraavan aloittamista, se voi olla tehoton ja riskialtis, jos ongelmia havaitaan prosessin myöhemmissä vaiheissa.

### 3.2.2 Ketterät menetelmät

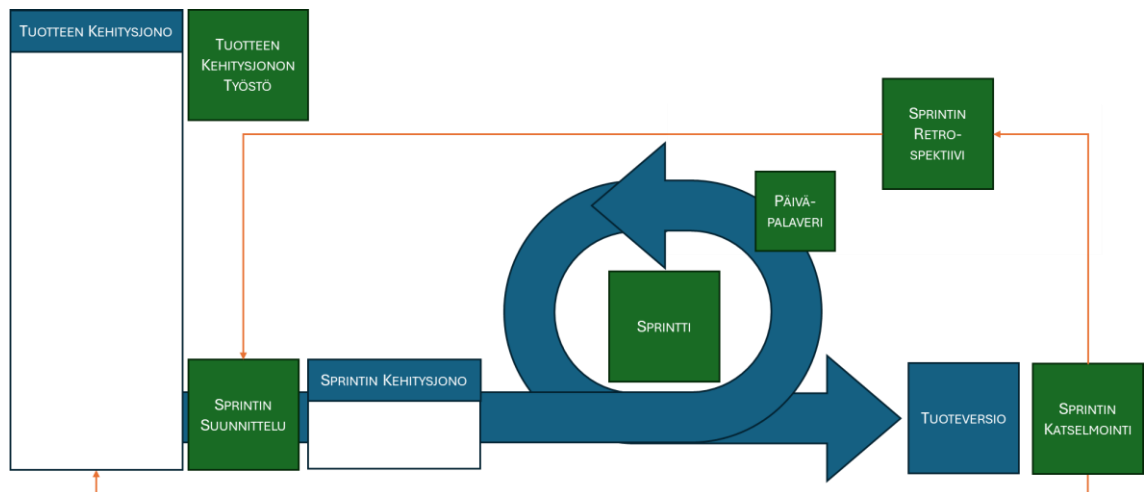
Ketterät menetelmät tunnetaan iteratiivisesta kehityksestä ja jatkuvasta asiakaspalautteen hyödyntämisestä. Niitä käytetään parhaiten sovelluskehityksessä, missä järjestelmävaatimukset voivat muuttua nopeasti. Tavoitteena on toimittaa käyttökelpoista ohjelmistoa asiakkaille nopeasti, mikä mahdollistaa jatkuvan palautteen ja vaatimusten päivityksen kehitysprosessin aikana. Ketterät menetelmät pyrkivät vähentämään byrokratiaa, kuten turhaa dokumentointia, jolla on epäselvä pitkän aikavälin arvo. [25.]

Ketterien menetelmien filosofia heijastuu agile-manifestossa, joka korostaa yksilöiden ja vuorovaikutusten merkitystä prosessien ja työkalujen yli, toimivan ohjelmiston dokumentaatiota enemmän, asiakasyhteistyötä sopimusneuvottelujen yli, ja muutoksiin reagoimista suunnitelmien noudattamisen sijaan. Tämä

manifesto on luotu tunnettujen ketterien menetelmien kehittäjien toimesta ja se ohjaa heidän lähestymistapojensa kehitystä. [26.]

Tunnetuimpia ketteriä menetelmiä ovat esimerkiksi Extreme Programming ja Scrum, jotka ovat integroineet useita perinteisiä kehitysmenetelmiä järjestelmällisyyden pohjalta. Ketterien menetelmien onnistuminen on johtanut niiden soveltamiseen monenlaisissa kehitysprojekteissa, mutta niiden soveltaminen suuriin järjestelmiin vaatii usein merkittäviä muutoksia, erityisesti kriittisten järjestelmien kehittämisessä, missä tarvitaan turvallisuus-, suojaus- ja luotettavuusanalyyssejä. [25.]

Pharmacilla oli jonkin verran kokemusta ketterien menetelmien Scrum-mallista, joten se valikoitui tarkemman tarkastelun kohteeksi tässä kartoituksessa. Scrumissa tiimi työskentelee sprintsissä, jotka sisältävät useita tapahtumia tai toimintoja. Jokaisen sprintin tulisi edistää tuoteversiota lähemmäs sen tavoitetta. Yhden sprintin tyyppinen kesto on 1–4 viikkoa. [27.] Kuvassa 9 näkyy ylätasoon malli Scrumista.



Kuva 9. Scrum-malli

Tuotteen kehitysajono on ajan tasalla oleva priorisoitu luettelo tuotteen toiminnallisuuksista ja edellisiltä sprintsiltä nousseista kehitysideoista. Jono on dynaaminen ja sitä päivitetään jatkuvasti uusien tarpeiden ja prioriteettien mukaan.

Sprintin kehitysajon on joukko tuotteen kehitysajon tehtäviä, jotka kehittäjätiimi on valinnut toteutettavaksi sprintin aikana. Tuoteversio päivittyy sitä mukaa, kun kukin sprintti valmistuu. [27.]

Sprintin suunnittelussa koko scrum-tiimi määrittelee sprintin tavoitteen tuotteen kehitysajon priorisoiduista tehtävistä. Suunnittelulle varataan enintään kahdeksan tuntia kuukauden mittaiselle sprintille. Lyhyemmillä sprinteillä aikaraja on lyhyempi. Päiväpalaverissa Scrum-tiimi tarkastelee edistymistä sprintin tavoitetta kohti ja mukauttaa tarvittaessa sprintin työajon. Päiväpalaverin tulisi kestää enintään 15 minuuttia päivässä. Sprintin katselmoinnissa koko Scrum-tiimi tarkastelee sprintin tuloksia sidosryhmien kanssa ja päättää tulevista mukautuksista. Sidosryhmät kutsutaan antamaan palautetta toteutetusta tuoteversiosta. Sprintin retrospektiivissä Scrum-tiimi arvioi, miten viime sprintti sujui henkilöiden, vuorovaikutusten, prosessien, työkalujen ja tuotosten suhteen. Tiimi tunnistaa parannuksia, joiden avulla seuraava sprintti voidaan tehdä tehokkaammaksi tai muutoin paremmin toimivaksi. [27.]

### 3.2.3 Vesiputousmalli vs. ketterät menetelmät

Tiedonkeruussa kartoitettiin sovelluskehityksen perinteistä vesiputousmallia sekä uudempia ketteriä menetelmiä. Puhtaasti kumpikaan yksinään ei vaikuttaisi soveltuvan Pharmacin tarpeisiin, joten molempien mallien hyviä ja huonoja puolia on seuraavassa verrattu erityisesti ottaen huomioon mallien mukautuminen projektinaikaisiin muutoksiin, riskienhallintaan, aikataulutukseen ja budjetointiin.

Vesiputousmallissa projektin vaiheet seuraavat toisiaan tiukassa lineaarisessa järjestyksessä: vaatimusten määrittely, suunnittelu, toteutus, testaus ja ylläpito. Ennakoitavuus ja selkeys ovat vesiputousmallin parhaita puolia ja malli voi olla erittäin hyödyllinen projekteissa, joissa vaatimukset ovat hyvin ymmärrettyjä ja vakaita läpi projektin. Mallin luonteen takia se suhtautuu muutoksiin jäykästi. Käytännössä kerran määritellyistä vaiheista voi olla vaikeaa tai jopa mahdotonta palata takaisin muuttamaan päätöksiä ilman merkittäviä seurauksia.

Ketterät menetelmät ovat sopeutuvaisia muutoksiin projektin vaatimuksissa, mikä tekee niistä sopivia projekteille, joissa vaatimukset saattavat kehittyä tai eivät ole alussa täysin selkeitä. Käyttäjien osallistaminen kehitysprosessiin säännöllisten arviointien ja iteraatioiden kautta voi johtaa korkeampaan asiakastytyväisyyteen, koska lopputuote vastaa paremmin käyttäjien tarpeita. Ketterät menetelmät jakavat projektin hallittaviin osiin, mikä voi auttaa minimoimaan riskejä, jotka liittyvät projektin monimutkaisuuteen ja muuttuviin vaatimuksiin. Tiimien itseohjautuvuus ja päätöksenteko ovat oleellisia ketterissä menetelmissä, mikä voi lisätä tiimin motivaatiota ja sitoutumista. Koska samoja ominaisuuksia kehitetään koko ajan, toistuvat niiden testaukset, arvioinnit iteraatiosta toiseen. Näin ollen ketterissä menetelmissä on mahdollista saada laadukkaampia tuotoksia.

Ketterät menetelmät vaativat jatkuvaa uudelleenarviointia, mikä voi johtaa vaikeuksiin ennustaa työmäärää ja resursseja. Tämä tekee kokonaiskustannusten ja projektin keston arvioinnin vaikeaksi projektin alussa. Projektien onnistuminen on vahvasti riippuvainen jatkuvasta käyttäjien ja liiketoimintaomistajien osallistumisesta, mikä voi olla haasteellista, jos kyseiset tahot eivät ole koko ajan saatavilla. Ilman huolellista hallintaa projektin laajuus voi kasvaa iteraatioiden ja usein toistuvien vaatimusten tarkennusten myötä, mikä voi johtaa korkeampiin kustannuksiin ja viivästyksiin.

Vesiputousmallissa riskit on tunnistettava ja hallittava varhain, sillä muutokset myöhemmissä vaiheissa voivat olla kalliita ja työläitä. Tämä vaatii perusteellista suunnittelua ja sitoutumista määriteltyyn suunnitelmaan, mikä voi vähentää joustavuutta vastata odottamattomiin haasteisiin.

Ketterässä kehityksessä riskienhallinta on jatkuvaa ja dynaamista. Tiimit arvioivat riskejä säännöllisesti sprinttien tai iteraatioiden aikana, mikä mahdollistaa nopeat korjausliikkeet ja riskien minimoimisen reaaliajassa. Tämä voi vähentää yllätyksiä ja suuria ongelmia projektin myöhemmissä vaiheissa. Käytännössä tämä kuitenkin tarkoittaa, että riskienhallintaan voi mennä huomattavasti enemmän työaika vesiputousmalliin verrattuna.



Vesiputousmallin aikataulut ja budjetit ovat ennalta määriteltäviä, ja niiden muuttaminen projektin aikana on monimutkaista. Malli edellyttää, että kaikki projektin näkökohdat määritellään tarkasti ennen toteutuksen aloittamista, mikä voi edesauttaa tarkan ja ennustettavan aikataulun ja budjetin luomiseen.

Ketterissä menetelmissä aikataulu ja budjetti ovat joustavia. Projektin vaatimuksia ja tavoitteita tarkastellaan ja päivitetään säännöllisesti, mikä mahdollistaa resurssien tehokkaan ja tarkoituksenmukaisen käytön. Vaikka tämä lisää muutosten hallinnan vaativuutta, se mahdollistaa myös paremman sopeutumisen asiakkaan tarpeisiin ja markkinoiden muutoksiin. Käytännössä kuitenkin on melko harvinaista, että yrityksessä on joustavaksi rakennettu budjetti, vaan projektille on yleensä budjetoitu tietty rahamäärä.

### 3.3 Toiminnalliset vaatimukset

Pharmacin kartoittaessa eri valmisohjelmistojen sopivuutta uudeksi järjestelmäksi luotiin vaatimusmäärittely sekä toiminnallisista että ei-toiminnallisista vaatimuksista. Toiminnallisia vaatimuksia oli yhteensä noin 250, joista osa oli hyvin ylätasoa vaatimuksia ja osa todella tarkalle tasolle määriteltäviä. Toiminnallisiin vaatimuksiin tutustuminen oli osa olemassa olevan tiedon hankintaa ja ei-toiminnalliset vaatimukset tulitaisiin taasen käymään läpi tietosuojavastaavan, IT-arkkitehdin, Product Ownerin ja hankepäällikön kanssa pidettävässä työpajassa. Tämä siksi, että muun muassa regulaatioon mahdollisiin muutoksiin ei ollut vielä tässä vaiheessa kirjattu ei-toiminnallisia vaatimuksia.

Osa toiminnallisista vaatimuksista koskivat koko järjestelmää ja/tai sen arkkitehtuuria. Kuvassa 10 on ote tällaisista vaatimuksista järjestelmän selainpohjaisuuden ja modulaarisuuden osalta.

A2.6	<b>Selainpohjainen ratkaisu</b> Toimitettavan ratkaisun käyttöliittymä on oltava selainpohjainen.
A2.9	<b>Modulaarisuus</b> Järjestelmä toteutetaan modulaarisesti.

Kuva 10. Ote järjestelmätason toiminnallisista vaatimuksista

Selainpohjaisen ratkaisun vaatimus pohjautuu siihen, että tuleva järjestelmä olisi mahdollisimman laiteriippumaton ja poistaisi mm. yksittäisten työasemien ylläpidollista työtä. Modulaarisuusvaatimukseen on taasen päädytty siitä syystä, että järjestelmän kehitys nyt ja tulevaisuudessa olisi helpompaa. Modulaarisuus mahdollistaa myös helpommin kokonaan uusien moduulien luomisen järjestelmään MVP-version jälkeen. Modulaarinen rakenne tulee helpottamaan myös järjestelmän testausta, koska yksittäisten toiminnallisuuksien ja moduulien testaus on helpompi testata ja tunnistaa tiettyyn moduuliin liittyvät bugit.

Järjestelmän pitää pystyä ottamaan vastaan apteekkirajapinnoista lähetettäviä tilauksia ja ottaa huomioon apteekkilainsäädäntö. Kuvassa 11 on esitetty esimerkkejä näistä vaatimuksista.

A2.97	<p><b>Apteekkien tilausliittymä</b></p> <p>Toteutettavan ratkaisun tulee sisältää rajapinta, jonka kautta vastaanotetaan apteekkien apteekkijärjestelmillään Pharmac Finlandille tekemät tilaukset sekä toimitetaan apteekkijärjestelmiin valikoimatiedot. Kokonaisuudessaan apteekkien tilausliittymä muodostuu nykytilassa yhteensä kuudesta eri tietovirrasta.</p>
A2.162	<p><b>Pääapteekki - sivuapteekki liitännäisyyksien esittäminen ja hallinta Tilaaja- eli asiakastiedoissa</b></p> <p>Toteutettavan ratkaisun tulee esittää tilaajien tiedot niin, että pääapteekki - sivuapteekki liitännäisyydet ovat helposti havaittavissa. Liitännäisyydet tulee esittää sekä niin, että pääapteekista ilmenee mahdolliset tämän sivuapteekit että niin, että sivuapteekista ilmenee tämän pääapteekki.</p>

Kuva 11. Ote annosjakeluun liittyvistä toiminnallisista vaatimuksista

Apteekkien tilausrajapinnan tulee pystyä ottamaan vastaan apteekteista Pharmacille lähetettävät tilaukset. Tilausrajapinnan pitää pystyä myös muun muassa hylkäämään tilaus, mikäli se sisältää Pharmacin valikoiman ulkopuolella olevia valmisteita. Apteekkarilla voi olla Fimealta lupa useammalle apteekille niin, että pääapteekin rinnalla toimii erillinen sivuapteekki. Juridisesti apteekit toimivat saman y-tunnuksen alla, mutta apteekkivero lasketaan tietyissä tapauksissa erikseen pää- ja sivuapteekille [28]. Tämä tulee ottaa huomioon erityisesti laskutusmoduulia suunniteltaessa.

Pharmac toimii Lielahden apteekin annosjakeluyksikkönä, jolloin esim. laskutus annosjakeluvalmisteista tapahtuu Lielahden apteekin nimissä. Kuvassa 12 on esimerkki tämän vaatimuksen toteuttamisesta. Kuvassa on esitelty myös ylätasoa vaatimus lääkkeiden hintatietojen hakemisesta Taksa-tietokannasta.

A2.211	<p><b>Pharmacin annosjakeluyksikkö-aseman huomiominen ratkaisun perusrakenteissa</b> Toteutettavan ratkaisun perusrakenteiden (vrt. tietomallin, johon ratkaisu pohjautuu) tulee olla suunniteltu ja toteutettu niin, että perusrakenteissa on huomioitu Pharmac Finland Oy:n annosjakeluyksikkö-status. Tästä seuraa esimerkiksi, että:</p> <p><b>Valmisteita myytäessä:</b></p> <ul style="list-style-type: none"> <li>- Myyjä on Lielahden Apteekki (=Pharmac Finland Oy:n sopimusvalmistusapteekki)</li> <li>- Toimittaja on Pharmac Finland Oy (=Myyjän Alihankkija)</li> <li>- Vaikka myyjä on Lielahden Apteekki, laskulla on Toimittajan eli Pharmac Finland Oy:n tilinumero (tämän mahdollistaa Lielahden Apteekin ja Pharmac Finlandin välinen sopimus saatavien siirrosta).</li> </ul> <p><b>Materiaaleja myytäessä ja erillislaskutuksessa (esim. kuljetusmaksut joissain tapauksissa):</b></p> <ul style="list-style-type: none"> <li>- Myyjä on Pharmac Finland Oy</li> <li>- Toimittaja on Pharmac Finland Oy</li> </ul> <p><b>Ei-lääkevalmisteita (eli ravintolisiä) muille kuin apteekeille myytäessä:</b></p> <ul style="list-style-type: none"> <li>- Myyjä on Pharmac Finland Oy</li> <li>- Toimittaja on Pharmac Finland Oy</li> </ul>
A2.114	<p><b>Taksa-liittymä</b> Toteutettavan ratkaisun tulee sisältää rajapinta Taksa-hintatietojen vastaanottamiseen (Hintalautakunnan kahden viikon välein määrittämät lääkevalmisteiden Tukku- ja Vähittäismyyntihinnat).</p>

Kuva 12. Ote annosjakelun regulaatioon liittyvistä vaatimuksista

Koska Pharmac myy ja laskuttaa myös muita kuin annosjakeluvalmisteita, tulee järjestelmän pystyä kuvan esimerkin mukaisesti laskuttamaan muita tuotteita niin, että niissä on toimittajana ja laskuttajana Pharmac. Näiden lisäksi tulee ottaa huomioon muun muassa se, että sopimusvalmistusapteekki voi muuttua, jolloin kyseinen muutos pitää olla mahdollista toteuttaa järjestelmään.

### 3.4 Työpajat

Tiedonkeruun toinen osio koostui neljästä eri työpajasta. Koska työpajojen teemat olivat laajoja, oli käytännön työpajoja lukumääräisesti huomattavasti useampi. Tässä työssä työpajojen sisällöt on yhdistetty teemojensa mukaisesti selkeyden vuoksi. Tiedonkeruun tämän vaiheen tarkoituksena oli kattaa eri asiantuntijoiden sekä sidostyhmien näkökulmat ja varmistaa, että suunniteltava sovelluskehitysmalli sekä työstettävän järjestelmän suunnittelu ja toteutus vastaisivat monipuolisesti Pharmacin tarpeita.

Työpajoihin oli valmisteltu liitteissä 1–4 kuvatut kysymyslomakkeet, mutta tavoitteena oli, että työpajoissa kerättäisiin myös muita keskustelun kautta nousevia asioita. Kysymyslomakkeet oli muokattu kehitettävään järjestelmään liittyen, koska sitä kautta olisi todennäköisempää löytää tarpeet myös yleiseen sovelluskehitykseen Pharmacilla. Työpajoihin osallistui teemahenkilön lisäksi muitakin asiantuntijoita ja mm. Pharmacin Product Ownerit ja hankepäällikkö olivat lähes kaikissa työpajoissa mukana. Työpajateemat keskittyivät seuraaviin alueisiin:

- Tietosuojavastaavan työpaja: Tämän työpajan tarkoituksena oli varmistaa, että järjestelmä noudattaa GDPR:n ja muiden relevanttien tietosuojalakien vaatimuksia. Työpajassa keskusteltiin henkilötietojen käsittelystä, tietoturvasta ja rekisteröityjen oikeuksien toteuttamisesta järjestelmässä. Työpajan avulla haluttiin varmistaa, että järjestelmä suunnitellaan alusta alkaen tietosuoja ja tietoturva huomioon ottaen.
- IT-arkkitehdin työpaja: Tässä työpajassa keskityttiin järjestelmän tekniseen suunnitteluun ja arkkitehtuuriin. Tavoitteena oli varmistaa, että järjestelmä on teknisesti toteutettavissa, skaalautuva ja yhteensopiva olemassa olevien IT-järjestelmien kanssa. Keskustelut käsitelivät modulaarisuutta, integraatioita, suorituskykyä, tietoturvaa ja käytettäviä tekniikoita, jotka vaikuttavat järjestelmän toimivuuteen ja kestävyteen.
- Product Ownerin työpaja: Tässä sessiossa keskityttiin liiketoiminnan vaatimusten ymmärtämiseen ja priorisointiin. Työpajan avulla haluttiin varmistaa, että järjestelmä tukee mahdollisimman hyvin Pharmacin liiketoimintaprosesseja ja että se vastaa käyttäjien tarpeisiin. Tässä työpajassa määriteltiin järjestelmän keskeiset toiminnallisuudet ja liiketoiminnan tavoitteet.

- Hankepäällikön työpaja: Tämä työpaja keskittyi projektinhallinnan strategiaan ja suunnitteluun, riskienhallintaan sekä projektin aikataulutukseen ja budjetointiin. Tavoitteena oli luoda selkeä suunnitelma järjestelmän toteutukseksi, mukaan lukien resurssien hallinta, aikataulujen määrittely ja kommunikaatiosuunnitelmat. Tässä vaiheessa varmistettiin, että projektin kaikki osa-alueet ovat hallinnassa ja että sidosryhmät ovat ajan tasalla projektin etenemisestä.

Näiden neljän työpajan järjestäminen auttoi varmistamaan, että järjestelmä suunniteltaisiin ja toteutettaisiin kokonaisvaltaisesti ottaen huomioon kaikki keskeiset näkökulmat: tekniset vaatimukset, liiketoiminnan tarpeet, tietosuoja- ja turvallisuusnäkökohdat sekä projektinhallinnan vaatimukset. Tavoitteena oli, että lopputulos olisi sekä teknisesti että liiketoiminnallisesti optimaalinen. Toteutettavan järjestelmän näkökulmasta tehty tiedonkeruu auttaisi rakentamaan sovelluskehitysmallin juuri Pharmacille tärkeiden asioiden ja tarpeiden näkökulmasta.

### 3.4.1 Tietosuojavastaava

Tietosuojavastaavan työpajan kysymysrunko on esitelty liitteessä 1. Kysymysten avulla pyrittiin saamaan kattava ymmärrys siitä, miten tietosuoja vaikuttaa järjestelmän suunnitteluun ja miten tietosuojavastaava voi tukea sen kehitystä GDPR:n ja muun regulaatioiden mukaisesti. Tavoitteena oli varmistaa, että järjestelmä tukee liiketoimintaprosesseja tietosuojan näkökulmasta. Tietosuojavastaavan työpajassa käsiteltiin tietoturvaan ja tietosuojaan liittyviä aiheita keskittyen erityisesti Pharmaciin nyt tai tulevaisuudessa vaikuttavaan regulaatioon. Työpajaan osallistuneet keskustelivat erityisesti seuraavista teemoista:

Koska terveydenhuollon tietojärjestelmissä käsitellään arkaluonteisia henkilötietoja, on tärkeää, että järjestelmä on suunniteltu suojaamaan näitä tietoja tehokkaasti. Tietoturva- ja tietosuojatoimet keskittyvät estämään tietomurtoja ja tietojen väärinkäyttöä. Työpajassa korostettiin, että tietosuojatoimien on oltava järjestelmällisiä ja kattavia, ja niiden on ulotuttava kaikkiin järjestelmän osiin.

Pharmaciin vaikuttava regulaatio, mukaan lukien tietosuojalaki ja sosiaali- ja terveydenhuollon tietojen käsittelyä säätelevät normit, asettavat tiukat vaatimukset

tietojen käsittelylle. GDPR:n vaatimukset ovat erityisen tärkeitä, sillä ne määrittelevät, miten henkilötietoja tulee suojata ja käsitellä. Työpajassa tarkasteltiin, kuinka nämä vaatimukset voidaan integroida järjestelmän kehitysprosessin kaikkiin vaiheisiin, varmistamaan että järjestelmä on suunniteltu alusta alkaen noudattamaan tietosuojalainsäädäntöä.

Työpajassa käsiteltiin myös lokitusvaatimuksia, jotka ovat olennainen osa tietosuojaa ja tietoturvastrategiaa. Lokien avulla voidaan seurata ja tarkastaa järjestelmän käyttöä sekä tunnistaa ja reagoida mahdollisiin tietoturvaloukkauksiin nopeasti.

Tietojen säilytyksen osalta keskusteltiin erilaisista säilytysajoista, jotka vaihtelevat riippuen tiedon luonteesta ja käyttötarkoituksesta. Erilaiset skenaariot vaativat erilaisia säilytysaikoja, ja työpajassa pohdittiin, miten järjestelmä voi tukea eri säilytysvaatimuksia ja -aikoja tehokkaasti ja lainsäädännön mukaisesti.

Tietosuojavastaavan työpajassa käytiin läpi myös Valviran vaatimuksia sosiaali- ja terveydenhuollon tietojärjestelmille, vaikka niitä ei tällä hetkellä Pharmacin järjestelmiin sovelletakaan. Lisäksi työpajassa katselmoitiin aiemmat listatut vaatimukset tietosuojan ja tietoturvan osalta.

### 3.4.2 IT-arkkitehti

IT-arkkitehdin työpajan kysymysrunko on esitelty liitteessä 2. Kysymysten tarkoituksena oli selvittää, miten IT-arkkitehti ja tekniset asiantuntijat ovat suunnitelleet tai tulisivat suunnittelemaan järjestelmän teknisen toteutuksen tukemaan liiketoimintaa ja käyttäjiä, sekä miten erilaisia vaatimuksia – kuten tietoturvaa, skaalautuvuutta ja integraatioita – on otettu huomioon. Lisäksi kysymykset auttaisivat ymmärtämään, miten arkkitehtuuri tukee järjestelmän elinkaaren hallintaa ja jatkuvaa kehitystä.

IT-arkkitehdille suunnatussa työpajassa käsiteltiin uuden järjestelmän arkkitehtuurin keskeisiä asioita, keskittyen erityisesti yksinkertaistamiseen,

modulaarisuuteen, skaalautuvuuteen, joustavuuteen sekä dokumentaation merkitykseen. Työpajassa nousi osallistujilta näkemyksiä siitä, miten IT-arkkitehtuuria tulisi lähestyä ketterän kehityksen puitteissa.

Työpajassa korostettiin tietomallin yksinkertaistamisen tärkeyttä, joka helpottaa muutosten hallintaa ja nopeuttaa iterointiprosessia. Samanaikaisesti mallin on oltava tarpeeksi kattava vastaamaan liiketoiminnan vaatimuksiin. Skaalautuvuus on keskeistä, sillä se takaa, että järjestelmä voi kasvaa ja kehittyä liiketoiminnan tarpeiden mukana. Tämä tarkoittaa, että tietomallin on pystyttävä käsittelemään kasvavaa transaktioiden, käyttäjien ja datan määrää.

Lisäksi keskusteltiin puhtaasti teknisistä ratkaisuista, kuten integraatiokerroksesta, lokituksesta, käyttöoikeuksista ja käyttöoikeusrooleista. Ketterien menetelmien osalta todettiin, että automaation kehittäminen olisi todella tärkeää muun muassa testauksen osalta, mikä mahdollistaisi iteratiivisesti kehitettävien toiminnallisuuksien testaamisen toistuvasti niiden kehittyessä.

Vaikka ketterässä kehityksessä pyritään välttämään turhaa dokumentaatiota, korostettiin, että järjestelmän keskeisten osien, arkkitehtuurin, käytettävien tekniikoiden ja periaatteiden dokumentointi on välttämätöntä. Hyvin laadittu dokumentaatio auttaa uusia tiimin jäseniä ymmärtämään nopeasti järjestelmän rakenteen ja toiminnan, mikä nopeuttaa perehdytysprosessia ja vähentää virheiden riskiä.

### 3.4.3 Product Owner

Product Owner -työpajan kysymysrunko on esitelty liitteessä 3. Näiden kysymysten tarkoituksena oli auttaa tunnistamaan, miten Pharmacin Product Ownerit näkevät järjestelmän roolin ja arvon yrityksen toiminnassa sekä miten he suunnittelevat ohjaavansa projektia sen koko elinkaaren ajan. Työpajassa korostuivat neljä pääteemaa: järjestelmän integroitavuus, käyttäjälähtöisyys, skaalautuvuus ja joustavuus sekä laadunvarmistus ja testaus.

Uuden järjestelmän integroitavuus olemassa oleviin tietojärjestelmiin nousi esiin keskeisenä haasteena. Tämä tarkoittaa, että uuden järjestelmän on kyettävä toimimaan saumattomasti mm. Pharmacin nykyisten tuotanto-, hallinto- ja apteekkijärjestelmien kanssa sekä pystyttävä hakemaan tietoa ulkoisista lähteistä muun muassa lääkkeiden hintojen osalta. Työpajassa keskusteltiin tarpeesta suunnitella tehokkaat ja toimivat rajapinnat, jotka takaavat tiedon sujuvan liikku-  
misen järjestelmien välillä. Tämän saavuttaminen vaatii syvää ymmärrystä nykyisten järjestelmien arkkitehtuureista, tietomalleista ja toiminnallisuuksista. On myös todennäköistä, että on tarpeen kehittää näiden järjestelmien rajapintoja ulkoisten toimijoiden kanssa.

Toinen keskeinen teema työpajassa oli käyttäjälähtöisyys. Tämän syynä oli varmasti osaltaan se, että kaikissa nykyisin käytössä olevissa järjestelmissä ei ole tätä huomioitu. Korostettiin, että sekä Pharmacin sisäisen henkilöstön, että apteekkien loppukäyttäjien, kuten farmaseuttien ja teknisten työntekijöiden, tulee osallistua kehitysprosessiin varhaisessa vaiheessa. Ketterät menetelmät tarjoavat tässä hyvän pohjan, sillä ne mahdollistavat joustavan palautteen keräämisen ja tuotteen jatkuvan iteroinnin käyttäjäpalautteen perusteella. Tämä auttaa varmistamaan, että lopputuote vastaa käyttäjien todellisia tarpeita ja parantaa tuotteen käyttöönoton sujuvuutta.

Järjestelmien skaalautuvuuden ja joustavuuden osalta keskusteltiin siitä, kuinka tärkeää on suunnitella järjestelmä siten, että se voi mukautua tulevaisuuden tarpeisiin, kuten käyttäjämäärän kasvuun tai lainsäädännön muutoksiin. Ketterät menetelmät tukevat tätä tavoitetta, sillä ne mahdollistavat joustavat ja nopeat muutokset kehitystyössä muuttuvien vaatimusten mukaisesti.

Laadunvarmistuksen ja testauksen merkitystä korostettiin erityisesti Pharmacin tuottaman annosjakelun osalta, jossa virheettömän toiminnan varmistaminen on kriittistä. Keskusteltiin siitä, kuinka tärkeää on investoida jatkuvaan laadunvarmistukseen ja kattavaan testaukseen, mukaan lukien automatisoidut testit ja käyttäjätestaukset. Tämä auttaa tunnistamaan ja korjaamaan mahdolliset



ongelmat varhaisessa vaiheessa, mikä parantaa tuotteen luotettavuutta ja vähentää kustannuksia pitkällä tähtäimellä.

#### 3.4.4 Hankepääällikkö

Hankepääällikön työpajan kysymysrunko on esitelty liitteessä 4. Kysymysten avulla oli tarkoitus auttaa hahmottamaan projektin kokonaiskuvaa hankepääällikön näkökulmasta ja antamaan ymmärrystä siitä, miten projektia tulisi hallita tehokkaasti. Keskeiset teemat käsittelivät budjetin ja aikataulun hallintaa, riskejä sekä MVP:n määrittämistä ja sen laajuuden hallintaa. Tässä yhteydessä käytiin läpi menetelmiä, työkaluja ja käytäntöjä, joita eri asiantuntijat olivat aiemmissa töissään käyttäneet.

Keskusteluissa korostettiin, että pysyminen budjetissa ja aikataulussa vaatii jatkuvaa seurantaa ja nopeaa reagointia muuttuviin olosuhteisiin. Hankepääällikön on tärkeä kehittää ja ylläpitää selkeitä viestintäkanavia kaikkien projektin sidosryhmien kanssa, jotta kaikki pysyvät ajan tasalla projektin edistymisestä ja mahdollisista haasteista. Strategiaksi ehdotettiin säännöllisiä tarkistuspisteitä ja iteraatiivista suunnittelua, jossa projektin tavoitteita ja resursseja arvioidaan säännöllisesti uudelleen. Tämä mahdollistaa joustavan resurssien kohdentamisen ja aikataulun säätämisen reaaliajassa, mikä on erityisen tärkeää ketterässä kehityksessä.

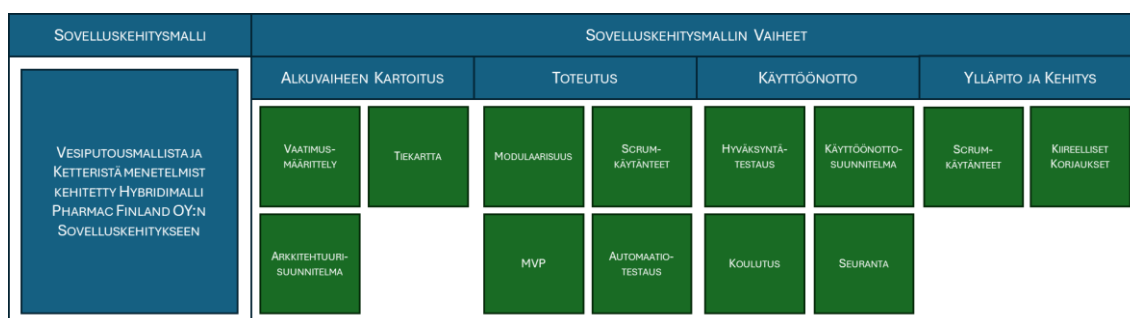
Työpajassa käsiteltiin, miten MVP:tä voidaan käyttää tehokkaana työkaluna projektin laajuuden hallitsemiseksi ja varmistamaan, että kehitystiimi keskittyy olennaiseen. Tässä yhteydessä painotettiin MVP:n jatkuvaa tarkastelua ja sen määrittelyn selkeyttä. MVP:n sisällön tarkka määrittely auttaa projektitiimiä ymmärtämään, mitkä projektin osat ovat kriittisiä ja mitkä voidaan jättää myöhempisiin kehitysvaiheisiin. Keskusteltiin myös siitä, miten MVP:tä voidaan käyttää asiakaspalautteen keräämiseen varhaisessa vaiheessa, mikä mahdollistaa tuotteen jatkuvan parantamisen perustuen todelliseen käyttäjäkokemukseen.

Lisäksi esiin nousi vahvasti riskeihin varautumisen tärkeys: riskien tunnistaminen, priorisointi ja niiden hallintasuunnitelmien kehittäminen auttavat minimoimaan yllätyksiä, jotka voivat vaikuttaa aikatauluun ja kustannuksiin.

## 4 Ehdotus kehitysmallista

Kuten aiemmin on todettu, ketterää kehitystä ei kannata suoraan soveltaa Pharmacin toimintamalliksi, eikä kumpikaan ohjelmistokehitysmalleista yksinään ole Pharmacille soveltuvin ratkaisu. Näin ollen olisi järkevintä luoda molempia malleja hyödyntävä hybridimalli. Tavoitteena tässä on yhdistää perinteisen vesiputousmallin suunnitelmallisuus ja ketterien menetelmien joustavuus. Tällainen malli voisi toimia tehokkaasti erityisesti regulaatiota ja monimutkaisia vaatimuksia sisältävissä projekteissa, kuten Pharmacin käynnissä olevassa järjestelmäprojektissa.

Hybridimalli mahdollistaisi järjestelmällisen lähestymistavan suurten ja kompleksisten vaatimusten määrittelyyn sekä joustavuuden ja reagoitakyvyn nopeasti muuttuvissa kehitys- ja liiketoimintaympäristöissä. Sovelluskehitysmalli koostuu kuvan 13 mukaisesti yleisestä kehyksestä, joka jakautuu alkuvaiheen suunnitteluun, toteutukseen, käyttöönottoon sekä ylläpitoon ja kehitykseen.



Kuva 13. Sovelluskehityksen hybridimalli

Jokaiseen vaiheeseen on kehitetty erilaisia työkalupohjia, joiden avulla vaiheiden toteutus ja seuranta on mahdollista. Mallissa on pyritty ottamaan mahdollisimman kattavasti Pharmacin kriittisimmät tarpeet huomioon. Mallin on tarkoitus

toimia ensimmäisenä versiona Pharmacille ja sitä on tarkoitus jatkokehittää muun muassa käynnissä olevan järjestelmäprojektin puitteissa. Seuraavissa luvuissa on esitelty vaiheet ja niihin tässä työssä luodut työkalut.

#### 4.1 Alkuvaiheen suunnittelu

Projektin alkuvaihe suunnitellaan vahvemmin vesiputousmallin mukaisesti, jossa yksityiskohtainen vaatimusmäärittely on keskeistä. Tässä vaiheessa tehdään analyysi liiketoimintaprosesseista, regulaatiovaatimuksista ja käyttäjien tarpeista. Tuotoksena tästä vaiheesta on projektille selkeä vaatimusmäärittely ja perusta, jolle koko järjestelmän kehittäminen rakentuu. Liiketoimintavaatimusten osalta lopputuotoksena on ylätason tiekartta, joka linkittyy järjestelmään toteutettaviin moduuleihin. Tässä vaiheessa on myös syytä tuottaa järjestelmälle ylätason arkkitehtuurisuunnitelma.

- Liiketoimintaprosessien analyysi: Ymmärretään ja kuvataan nykyiset prosessit, joita järjestelmän on tarkoitus tukea. Analyysissa tunnistetaan nykyisten prosessien hyviä puolia, pullonkauloja ja parannusmahdollisuuksia. Tarkoituksena ei ole siis siirtää nykyisiä prosesseja yksi-yhteen vain uuden teknisen alustan päälle.
- Regulaatiovaatimusten määrittely: Määritellään lääkealalle ja terveydenhuollon toimialalle tyypilliset regulaatiovaatimukset, jotka vaikuttavat järjestelmän toimintaan. Erityistä huomiota kiinnitetään säädöksiin, jotka liittyvät tietosuojaan ja tietoturvaan, lääkitysturvallisuuteen ja lääkkeiden seurantaan ja jäljitettävyyteen.
- Käyttäjien tarpeiden kartoitus: Käyttäjät, mukaan lukien sisäiset työntekijät, laskutus, hallinto, myynti ja mahdolliset ulkoiset asiakkaat sekä muut sidosryhmät otetaan mukaan kartoitukseen. Heidän palautteensa ja tarpeidensa ymmärtäminen on kriittistä järjestelmän käyttökelpoisuuden ja hyväksynnän kannalta.
- Vaatimusmäärittely: Kaikki kerätty tieto dokumentoidaan huolellisesti. Tämä sisältää yksityiskohtaisen vaatimusmäärittelydokumentin, jossa kuvataan sekä toiminnalliset että ei-toiminnalliset vaatimukset ja se toimii projektin lähtökohtana. Nämä vaatimukset ovat pääosin regulaatiosta johdettuja vaatimuksia, sekä toiminnallisia ja ei-toiminnallisia vaatimuksia, jotka kohdistuvat koko järjestelmään.
- Tiekartta: Liiketoimintatarpeista ja muista vaatimuksista luodaan ylätason tiekartta koko projektille. Liiketoimintavaatimuksia ei jalosteta vielä tässä vaiheessa kaikilta osin käyttäjätarinoiksi, vaan

tavoitteena on tunnistaa ja kuvata järjestelmän eri moduulit ja niiden keskeisimmät toiminnallisuudet ylätasolla.

- Järjestelmän arkkitehtuurin suunnittelu: Luodaan korkean tason suunnitelma siitä, miten järjestelmä toteutetaan arkkitehtuurisesti. Tämä kattaa teknisen infrastruktuurin, tietokantojen suunnittelun, kolmansien osapuolien integraatiot ja järjestelmän yleiset turvallisuusvaatimukset.

#### 4.1.1 Vaatimusmäärittely

Arkkitehtuuriset ja regulaatiosta johdetut vaatimukset suunnitellaan ja toteutetaan perinteisen vesiputousmallin mukaisina vaatimuksina. Tämä tehdään siitä syystä, että nämä ovat tyypillisesti vaatimuksia, jotka muuttuvat harvoin. Vaatimukset sisältävät järjestelmän perusrakenteeseen ja toiminnallisuuteen liittyviä asioita sekä regulaatioista johdettuja vaatimuksia. Nämä määritellään yksityiskohtaisesti projektin alussa ja niiden muutoksiin suhtaudutaan varoen, sillä ne vaikuttavat usein laajasti koko järjestelmään.

Pharmacin tapauksessa em. kaltaisia vaatimuksia on listattu jo aiemmassa vaatimusmäärittelyvaiheessa. Kuvassa 14 on esitelty GDPR:ään liittyviä vaatimuksia rekisteröidyn pyyntöön oikeudesta saada pääsy tietoihin, henkilötietoa sisältävien tietovarantojen dokumentaatiosta ja lokitiedoista.

B2.39	<p><b>GDPR:n mukaiseen pyyntöön oikeudesta saada pääsy tietoihin varautuminen</b>  Rekisteröidyllä (esim. loppukäyttäjä tai työntekijä) on oikeus saada rekisterinpitäjältä vahvistus siitä, käsitteekö tämä häntä koskevia henkilötietoja. Näin rekisteröidyllä on mahdollisuus arvioida ja varmistaa käsittelyn lainmukaisuus.  Jos rekisteröidyn tietoja käsitellään, rekisterinpitäjän on toimitettava hänelle jäljennös käsiteltävistä henkilötiedoista. Jos rekisteröity esittää pyynnön sähköisesti, tiedot on toimitettava yleisesti käytetyssä sähköisessä muodossa, paitsi jos rekisteröity toisin pyytää.</p>
B2.44	<p><b>Henkilötietoa sisältävien tietovarantojen määrittelemine dokumentaatiassa</b>  Tietovarantojen, jotka sisältävät henkilötietoa tulee olla selkeästi kuvattuna järjestelmän dokumentaatiassa (kyseisten tietovarantojen sijainti ja näkyvyys).</p>
B2.61	<p><b>Lokitietojen luokittelu</b>  Toteutettavassa ratkaisuissa lokitietojen tuottaminen tulee toteuttaa niin, että ratkaisu luokittelee lokitiedot niitä tuottaesaan loogisella tasolla automaattisesti esim. käyttölokiteoihin (=tieto tietojen hausta, katselusta ja käytöstä) ja teknisiin lokiteoihin (tai muihin yksiselitteisesti määriteltäviin luokkiin).</p>

Kuva 14. Esimerkki GDPR:ään liittyvistä vaatimuksista

Käynnissä olevaa projektia varten osa aiemmin työstetyistä vaatimuksista oli tehty riittävälle tasolle, mutta osa vaatii vielä tarkentamista. Esimerkissä GDPR:n osalta lokitietojen luokittelu tulisi vielä tarkentaa selkeämmäksi ja yksiselitteisemmäksi vaatimukseksi.

#### 4.1.2 Tiekartta

Toiminnalliset vaatimukset ovat usein liiketoiminnan tarpeista johdettuja ja voivat olla alttiimpia muutoksille projektin edetessä. Vaikka nämä vaatimukset kirjataan myös projektin alkuvaiheessa, niille on tyypillistä suurempi muutosalttius, ja niiden kehittäminen voi edetä ketterän mallin mukaisesti iteratiivisten ja inkrementaalisten toimitusten kautta.

Kuvassa 15 on esitelty luonnos tiekartasta Pharmacin käynnissä olevasta järjestelmäprojektistä. Tiekarttaan on merkattu sekä alussa vesiputousmallin mukaisesti tehtävät asiat, että liiketoimintatarpeista moduuleiksi muodostetut toiminnallisuuskokonaisuudet. Tiekartan tavoitteena on visualisoida koko projektin sisältö ja ylätasen aikataulut.

ROADMAP															
VAATIMUSMÄÄRITTELY	■														
ARKKITEHTUURISUUNNITELMA	■	■													
TEKNINEN YMPÄRISTÖ			■												
INTEGRAATIOALUSTA			■												
STORYBOOK				■											
ASIAKKAAT & ASIAKUUDET				■											
TUOTTEET				■	■										
TILAUSTEN VASTAANOTTO						■									
TILAUSTEN TUOTTAMINEN						■	■	■							
LASKUTUS & HINNOITTELU								■	■						
TOIMITUS									■						
UAT										■					
KÄYTTÖÖNOTTO												■	■		
YLLÄPITOON SIIRTYMINEN													■	■	

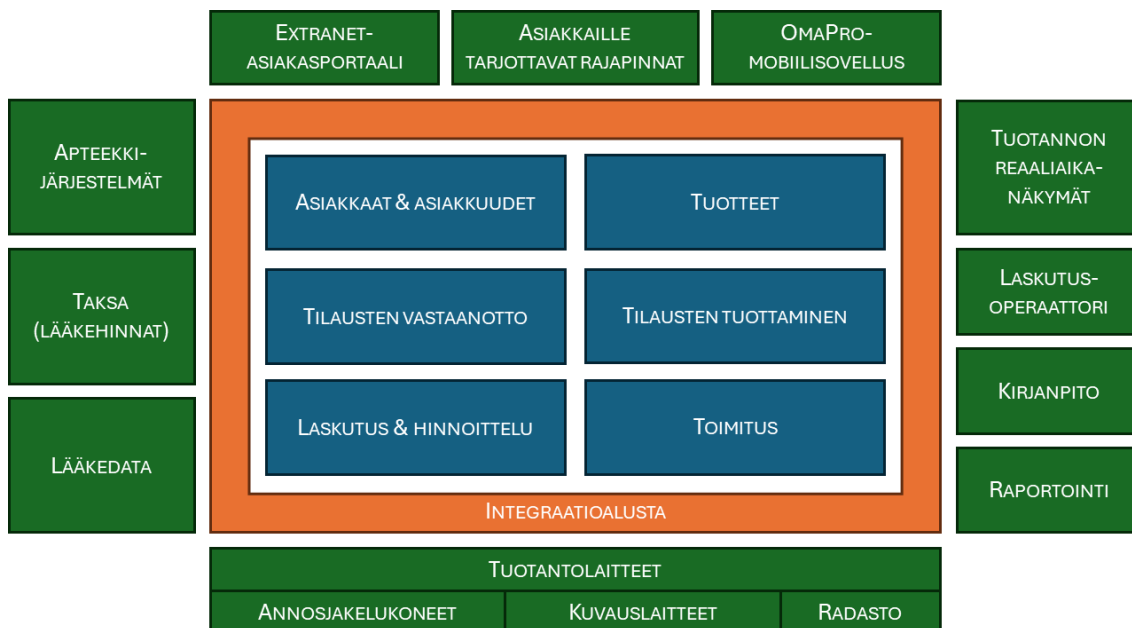
Kuva 15. Luonnos tiekartasta

Projektin alussa tehtävä vaatimusmäärittely, arkkitehtuurisuunnitelma, tekninen ympäristö ja integraatioalusta ovat perinteisellä vesiputousmallilla tehtäviä kokonaisuuksia, jolloin niiden sisältö ja niihin kuluva aika on tarkemmin selvillä. Liiketoimintatarpeista muodostetut moduulit, kuten esim. asiakkaat ja asiakkuudet, tuotteet, laskutus ja hinnoittelu on määritelty vasta ylätasolla, jolloin niiden työ määrään ei ole niin selkeää näkyvyyttä.

Käynnissä olevassa järjestelmäprojektissa on kuitenkin toteutettava kaikki moduulit, ja projektille on varattu tietty budjetti. Tämä vaatii tiekartan jatkuvaa ja säännöllistä päivittämistä, jotta projektin toteutumista voidaan peilata tavoitteisiin. Budjetissa ja aikataulussa pysyminen vaatii jokaisen moduulin osalta vaatimusten priorisointia moduulin sisällä, jotta yksittäisten moduulien tekemiset eivät vie liikaa kokonaisprojektin resursseja. Tiekartan säännöllisen päivittämisen olisi hyvä tapahtua Product Ownereiden ja hankepäällikön yhteistyönä. Päivittämisen olisi hyvä tapahtua muutaman sprintin välein, jotta projektin kokonaiskuva on vastuullisilla ajantasaisena.

#### 4.1.3 Arkkitehtuurisuunnitelma

Arkkitehtuurisuunnitelma on keskeinen dokumentti ohjelmistokehitysprojekteissa, ja se tarjoaa selkeän kuvauksen ohjelmiston rakenteesta, käytetyistä teknologioista, järjestelmän osista ja niiden välisistä suhteista. Kuvassa 16 on esitelty hahmotelma Pharmacille rakennettavan järjestelmän liiketoiminta-arkkitehtuurista, jossa näkyy järjestelmän päätoiminnallisuudet jaoteltuna moduuleihin ja suhteet ulkoisiin järjestelmiin.



Kuva 16. Liiketoiminta-arkkitehtuuri

Arkkitehtuurisuunnitelmassa mietitään järjestelmän keskeiset moduulit ja se, miten ne kommunikoivat keskenään. Järjestelmän kommunikointi ja integrointi ulkoisiin järjestelmiin ja datalähteisiin tulee myös miettiä. Suunnitelmassa tulee miettiä projektissa käytettävät ohjelmointikielet ja -kehikset, infrastruktuuri, sekä käytetäänkö pilvipalveluita vai paikallisia palvelimia. Liiketoiminnan puolelta on hyvä pohtia vaatimukset järjestelmän suorituskyvylle ja skaalautuvuudelle, mukaan lukien odotetut kuormitustasot ja järjestelmän kyky kasvaa käyttäjämäärien tai datan määrän kasvaessa.

Järjestelmän käyttämät ja sisältämät tietomallit ja tietokantarakenteet tulisi miettiä ylätasolla tässä vaiheessa. Näiden muuttaminen on mahdollista ja todennäköistä projektin edetessä liiketoimintavaatimusten selkeytyessä, mutta esim. tietomallin muuttaminen kokonaan toisenlaiseksi voi lisätä kehitystyötä merkittävästi ja vaatia muutoksia useisiin moduuleihin. Kuvassa 17 on esitelty ote mallinnuspohjasta tuotteet-moduulille.

TUOTEDATA			TUOTEPERHE								
KATEGORIA	NIMI	FIELD TYPE	AJ-TUOTTEET (TAKSA)	AJ-TUOTTEET	AJ-PALKKIOT	PALVELUTUOTTEET	FYSISET TUOTTEET	TALLENNETTUOTTEET	WEBINAARIT	KUVATUOTTEET	DOKUMENTITUOTTEET
PERUSTIEDOT	NIMI	TEXT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	TUOTEPERHE	SIMPLE SELECT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	TUOTERYHMÄ	SIMPLE SELECT									
PERUSTIEDOT	ID	NUMBER	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	ID TUOTEPERHEESSÄ	NUMBER	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	KUVAUS	LONG TEXT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	KUVA	IMAGE	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	VALMISTAJA	SIMPLE SELECT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	TOIMITAJA	SIMPLE SELECT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	VALMISTUSMAA	SIMPLE SELECT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	MYYNITILI	SIMPLE SELECT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	PROJEKTINUMERO	SIMPLE SELECT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	KUSTANNUSPAIKKA	SIMPLE SELECT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	TUOTEOMISTAJA	SIMPLE SELECT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	KERTA-/JATKUVAVELOITTEINEN	BOOLEAN	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	AKTIIVINEN/PASSIIVINEN	BOOLEAN	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	STATUS (LUONNOS YMS)	SIMPLE SELECT	X	X	X	X	X	X	X	X	X
PERUSTIEDOT	YKSIKKÖ	SIMPLE SELECT	X	X	X	X	X	X	X	X	X

Kuva 17. Ote tuotteet-mallinnuspohjasta

Pohjan tarkoituksena on auttaa tunnistamaan kaikki tiedot, jotka tuotteet-moduuliin liittyvät ja myös miettiä sitä, millainen tietotyyppi ko. tieto on. Koska eri moduuleissa on todella suuri määrä tietoja, on niiden listaus taulukkomaiseen muotoon järkevämpää, kuin yrittää kuvata niitä esim. ketterien menetelmien mukaisen käyttäjätarinan avulla.

Kuvassa 18 on esitelty laskutus ja hinnoittelu -moduulin osalta ote laskutusoperaattorille lähetettävästä tiedosta. Taulukkoon on kirjattu Pharmacin järjestelmän tietomallissa olevat tiedot ja kyseisten tietojen välitys laskutusoperaattorille lähetettävässä tiedostossa.



GUI	CUSTOMER	INVOICER	FINVOICE ID
Y-TUNNUS		SELLERPARTYIDENTIFIER	19
VERKKOLASKUTUNNUS		SELLERORGANISATIONUNITNUMBER	5
VERKKOLASKUOPERAATTORI		INTERMEDIATOR	7
Y-TUNNUS	INVOICERECIEPIENTPARTYIDENTIFIER		83
VERKKOLASKUTUNNUS	INVOICERECIEPIENTORGANISATIONUNITNUMBER		9, 99
VERKKOLASKUOPERAATTORI		INTERMEDIATOR	11

Kuva 18. Ote laskutusoperaattorille lähetettävistä tiedoista

Käytännössä kuva toimii integraatiokuvauksena tiedosta, joka välitetään Pharmacin järjestelmästä eteenpäin ulkoiseen järjestelmään, tässä tapauksessa laskutusoperaattorin rajapintaan. Nämä tiedot ja rakenteet on hyvä pyrkiä pitämään rakenteellisesti ja tarkasti kuvattuina, koska ilman tällaista dokumentaatiota on vaikea tunnistaa esimerkiksi integraatioihin kohdistuvia vaikutuksia, kun järjestelmän tietomalliin tehdään muutoksia.

Edellä mainittujen lisäksi arkkitehtuurikuvauksessa tulisi suunnitella järjestelmän tietoturva ja tietosuoja ylätasolla. Käytännössä tämä tarkoittaa muun muassa autentikointi- ja autorisointimekanismeja, lokitusta, käyttöoikeus ja käyttövaltuushallintaa, tietoturva-aukkojen hallintaa sekä tietosuojan noudattamista.

## 4.2 Toteutus

Jokainen järjestelmän moduuli suunniteltaisiin ja toteutettaisiin tarkemmalla tasolla käyttäen Scrumia. Tämä tarkoittaisi iteratiivista kehitystä, jossa lyhyet sprintit keskittyvät tietyille moduulille asetettujen priorisoitujen ominaisuuksien tuottamiseen. Toteutuksista saatu palaute ja uudet ideat integroitaisiin jatkuvasti kehitykseen. Scrumia käytettäessä on kuitenkin hyvä sopia tietyistä vakioiduista asioista teknisen, visuaalisen ja toiminnallisen toteutuksen suhteen. Myös itse Scrumin käytänteet on hyvä mallintaa selkeiksi kaikille. Seuraavissa luvuissa on esitelty toteutusvaiheen keskeisimmät työkalupohjat.

#### 4.2.1 Modulaarinen ja uudelleenkäytettävä toteutus

Modulaarisuus tarkoittaa järjestelmän rakentamista itsenäisistä, erillisistä moduuleista, joita voidaan kehittää, testata, käyttää ja päivittää itsenäisesti. Tämä lähestymistapa mahdollistaa joustavuuden muutoksille, kun muutoksia voidaan tehdä yksittäisiin moduuleihin. Myös vianetsintä, virheenkorojaukset ja päivitykset voidaan tehdä nopeammin ja kohdennetummin yksittäisiin moduuleihin.

Moduulit ja niiden sisältämät komponentit tulee suunnitella hyvin, jolloin niiden uudelleenkäyttäminen on mahdollista järjestelmän muissa osissa. Uudelleenkäytettävyys ei rajoitu pelkästään teknisiin komponentteihin, vaan myös käyttöliittymäelementtien visuaalisuus kannattaa tehdä yhtenäiseksi ja johdonmukaiseksi. Käyttöliittymän elementtien, kuten värien, fonttien ja asettelun, yhtenäisyys koko järjestelmässä parantaa käytettävyttä ja auttaa käyttäjiä intuitiivisempaan toimintaan järjestelmäkokonaisuuden osalta, kun asiat toimivat samalla tavalla ja näyttävät samoilta kautta linjan. Kuvassa 19 on esimerkki käyttäjän lisääminen -näköymästä Figma-työkalulla tehtynä.

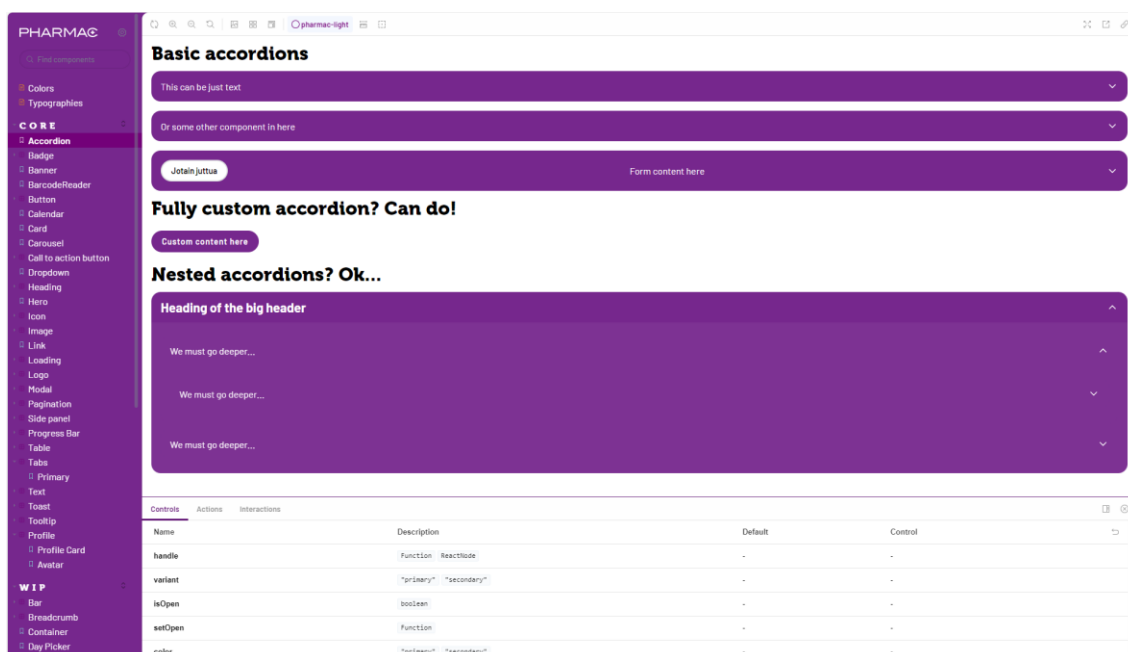
The image shows a user creation form with the following fields and elements:

- Logo** (top left)
- PERUUTA Lisää käyttäjä** (top navigation)
- Käyttäjä** (form title)
- Rooli** (dropdown menu): Apteekkari
- Sukunimi** (dropdown menu): Apteekkari
- Etunimi** (dropdown menu): Assi
- Sähköpostiosoite** (text input): assi.a.apteekkari@apteekki.com
- Puhelinnumero** (text input): 044-123 456
- Käyttäjätunnus (miten tämä muodostuu ja toimitetaan?)** (section title)
- Käyttäjätunnus** (password input): \*\*\*\*\*
- Generoi uusi salasana** (button)
- Uusi käyttäjä +** (button)
- Sidebar menu:** Logo, Etusivu, Asiakaspalvelu, Tuotehallinta, Tuotanto, Pakkaamo, Laskutus, Asiakkaat

Kuva 19. UI-luonnos käyttäjän luonnista

Käyttöliittymäsuunnitteluun valikoitui Pharmacilla Figma-työkalu, jonka avulla projektissa toteutetaan sekä ylätasoa konsepteja, tarkempia prototyyppisiä ja yksityiskohtaisia käyttöliittymäleiskoja. Sen avulla suunnittelijat ja kehittäjät voivat luoda interaktiivisia prototyyppisiä, jotka he voivat testata käyttäjien kanssa reaaliajassa. Figmalla on tarkoitus luoda edellä mainittujen lisäksi vakioitu tyyliopas ja komponenttikirjasto, jotta suunnitellut komponentit olisivat eri moduuleissa yhtenäisiä ja uudelleenkäytettäviä.

Käyttöliittymäsuunnittelun vakioinnin lisäksi myös tekniset komponentit on hyvä vakioida. Kuvassa 20 on ote Storybook-työkalusta, jonka avulla luodaan uudelleenkäytettäviä teknisiä komponentteja. Storybook on avoimen lähdekoodin työkalu, joka on suunniteltu auttamaan kehittäjiä rakentamaan ja järjestämään käyttöliittymän komponentteja eristetyssä ympäristössä. Tämä mahdollistaa komponenttien suunnittelun, kehityksen ja testauksen erillään pääsovelluksesta, mikä tekee kehitysprosessista nopeampaa ja joustavampaa.



Kuva 20. Ote Storybook-työkaluun mallinnetuista teknisistä komponenteista

Storybook toimii itsenäisenä kehitysympäristönä, jossa kehittäjät ja suunnittelijat voivat luoda ja hienosäätää käyttöliittymäkomponentteja. Se tukee useita kehitysalustoja, kuten Reactia, Vuea ja Angularia. Storybook toimii teknisenä komponenttikirjastona, jossa kaikki järjestelmän ja moduulien komponentit ovat yhdessä paikassa. Tämä tarkoittaa, että kun jokin komponentin osa muutetaan, kuten esimerkiksi brändiväri, muutos päivittyy automaattisesti kaikkiin sovelluksen osiin, joissa kyseistä komponenttia käytetään. Tämä poistaa tarpeen muokata jokaista komponenttia erikseen ja varmistaa visuaalisen yhdenmukaisuuden koko sovelluksessa.

#### 4.2.2 Scrum-käytännöt

Kappaleessa 3.2.2 esitelty Scrum valittiin kehykseksi toteutusosioon. Pharmacilla työkaluksi Scrumin toteuttamiseen ja seuraamiseen valikoitui Microsoftin DevOps. Tämä tehdään pääosin siitä syystä, että Pharmacin toimistosovellukseen käytetään Microsoftin tuotteita, joten työkalun käyttöönotto ja käyttäminen olisi nykyisille käyttäjille melko intuitiivista. MS DevOps -työkalussa vaatimukset rakentuvat neljästä eri suurusluokaltaan olevasta vaatimuksesta/tavoitteesta. Näitä ovat Epic, Feature, User Story ja Task.

Epic on laaja, korkean tason vaatimus tai tavoite, joka koostuu useista pienemmistä osista ja joka yleensä kestää pidempään toteuttaa. Se on niin suuri, että se vaatii jakamista pienempiin osiin (Featureihin ja User Storyihin) ollakseen hallittavissa. Epicit auttavat kuvaamaan laajoja liiketoiminnallisia tavoitteita ja niitä käytetään ryhmittelemään yhteen liittyviä featureita ja tarinoita. Käytännössä Pharmacilla toteutettavassa järjestelmäprojektit kannattaa mallintaa Epic-tasoisiksi.

Feature on Epicin alainen käsite, joka koostuu useasta User Storysta ja kuvaa tietyn toiminnallisuuden tai toiminnallisuusryhmän, joka tarjoaa arvoa käyttäjälle. Feature on yksityiskohtaisempi kuin Epic, mutta silti riittävän laaja kattaakseen yhden kokonaisvaltaisen toiminnallisuuden tai liiketoimintaprosessin osa-alueen.

User Story on ketterän kehityksen yksi peruselementti, joka kuvaa ominaisuutta tai toiminnallisuutta loppukäyttäjän näkökulmasta. Se on pienin yksikkö, joka voidaan toteuttaa, testata ja toimittaa itsenäisesti. User Story sisältää yleensä kolme osaa: kuka (käyttäjä), mitä (toiminto) ja miksi (hyöty). Kuvassa 22 on esimerkki User Storystä.

The screenshot shows a Jira User Story page for ID 10775, titled 'Laskun lähettäminen yritysasiakkaalle'. The story is in the 'New' state, located in the 'PharmacDev\Development' area and 'PharmacDev\Sprint 32-2022' iteration. The description includes a role (Talouhallinnon työntekijä), a need (to send invoices to business customers), and a goal (fast and error-free processing). The acceptance criteria are detailed in five numbered sections: 1. Invoice creation, 2. Invoice review and approval, 3. Invoice sending, 4. Invoice payment confirmation, and 5. Error handling. The right sidebar shows planning details (Story Points, Priority 2, Risk), classification (Business), and board information (Development/Stories, Column 17/19).

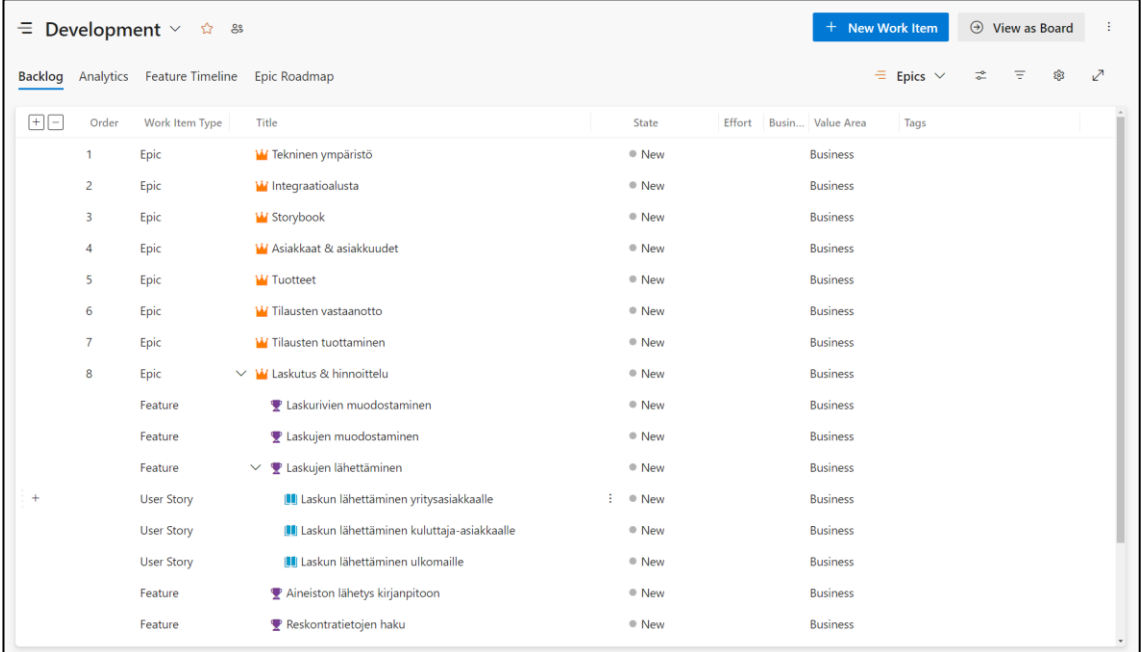
## Kuva 21. Esimerkki User Storystä

Esimerkissä on kuvattu laskun lähettäminen yritysasiakkaalle -käyttäjätarina. Tarina on kuvattu taloushallinnon työntekijän näkökulmasta. Acceptance Criteria -osioon on kuvattu käyttäjätarinan hyväksymiskriteerit. Nämä ovat monesti tärkeitä vaatimukselle, jonka pohjalta kehittäjälle tulee selkeämpi kuva

haluttavasta toteutuksesta. Hyväksymiskriteerit toimivat myös hyvänä pohjana käyttäjätarinan testaukselle.

User Story muodostuu monesti useammasta kappaleesta Taskeja, eli tehtäviä. Storyn pilkkominen Taskeihin tapahtuu yleensä sovelluskehittäjän toimesta ja Taskit ovat monesti luonteeltaan teknisiä. Tämä auttaa sovelluskehittäjää pilkkomaan käyttäjätarinan, joka on tehty liiketoimintanäkökulmasta, loogisiin teknisiin palasiin.

Tuotteen kehitysjonon hallintanäkymä MS DevOps -työkalussa näkyy kuvassa 23. Aiemmin esitellyt eri laajuusluokan vaatimukset näkyvät työkalussa hierarkisena rakenteena.

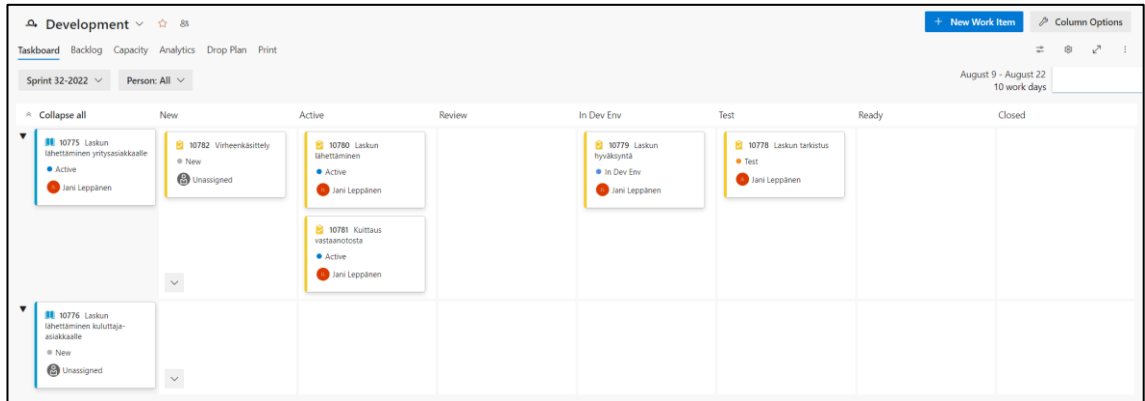


Order	Work Item Type	Title	State	Effort	Busin...	Value Area	Tags
1	Epic	Tekninen ympäristö	New			Business	
2	Epic	Integraatioalusta	New			Business	
3	Epic	Storybook	New			Business	
4	Epic	Asiakkaat & asiakkuudet	New			Business	
5	Epic	Tuotteet	New			Business	
6	Epic	Tilausten vastaanotto	New			Business	
7	Epic	Tilausten tuottaminen	New			Business	
8	Epic	Laskutus & hinnoittelu	New			Business	
	Feature	Laskurivien muodostaminen	New			Business	
	Feature	Laskujen muodostaminen	New			Business	
	Feature	Laskujen lähettäminen	New			Business	
	User Story	Laskun lähettäminen yritysasiakkaalle	New			Business	
	User Story	Laskun lähettäminen kuluttaja-asiakkaalle	New			Business	
	User Story	Laskun lähettäminen ulkomaille	New			Business	
	Feature	Aineiston lähetyksen kirjanpitoon	New			Business	
	Feature	Reskontratietojen haku	New			Business	

Kuva 22. Tuotteen kehitysjojo

Tuotteen kehitysjonolla priorisointia voi tehdä raahaamalla vaatimuksia listauksessa, jolloin priorisointi näkyy suoraan näkymässä niin, että suuremman prioriteetin vaatimukset ovat listassa ylimpänä.

Sprintin kehitysjonolle valikoidaan tiimin kesken sprintin suunnittelussa seuraavalle sprintille otettavat User Storyt ja ne jalostetaan tämän jälkeen Taskeiksi. Sprintin kehitysjonoa seurataan tarkemmassa näkymässä kuvan 24 mukaisesti.



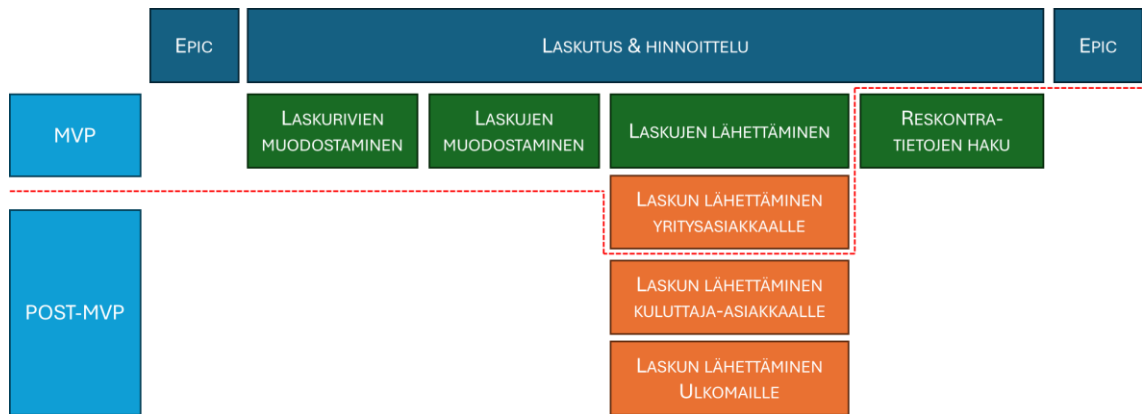
Kuva 23. Sprint Backlog

Sprintin kehitysjonoa käytetään Scrum-tiimin päivittäisenä työkaluna sprintin aikana. Se sisältää kaikki sprintin aikana suoritettavat tehtävät ja on jatkuvasti näkyvissä koko tiimille. Tiimi päivittää sprintin työjonoa säännöllisesti heijastamaan tehtävien edistymistä ja jäljellä olevaa työtä. Tämä lisää läpinäkyvyyttä ja mahdollistaa joustavuuden sprintin aikana, kun tiimi voi reagoida muutoksiin ja uusiin tietoihin.

#### 4.2.3 MVP

Tuotteen kehitysjonoa hallitaan edellä kuvatun mukaisesti MS DevOps -työkalussa. Listamuotoinen kehitysjono ei kuitenkaan ole paras, kun seurataan kokonaiskuvaa ja mietitään, mitä toiminnallisuuksia järjestelmän MVP-versioon ollaan toteuttamassa.

Kuvassa 25 on MVP:n hallintaan ja visualisointiin luotu pohja. Projektin hallinta, erityisesti ketterässä ympäristössä, edellyttää jatkuvaa seuranta ja mukauttamista, jotta lopputulos pysyy sekä aikataulussa että budjetissa.



Kuva 24. MVP:n hallinta

MVP:n lisäksi on olennaista ylläpitää ja säännöllisesti päivittää projektin tiekarttaa, joka esiteltiin luvussa 4.1.2. MVP-pohja on käytännössä tarkemman tason versio tiekartasta, ja vastaavasti MS DevOps -työkalu mahdollistaa MVP:n jakamisen pienemmiksi tehtäviksi. MS DevOps -työkalu on aiemmin kuvatun mukaisesti osana päivittäistä työtä ja tiekarttaa sekä MVP-pohjaa tulee päivittää mielellään parin sprintin välein.

#### 4.2.4 Testaus

Testaus on olennainen osa ketteriä menetelmiä, ja sen tärkeyttä ei voi korostaa liikaa. Ketterässä kehityksessä muutokset ja iteratiivinen kehitys ovat jatkuvia, ja testaus varmistaa, että nämä muutokset eivät heikennä tuotteen laatua tai suorituskykyä. Pharmacin tapauksessa testausta tehdään kehittäjien toimesta yksikkö- ja integraatiotesteillä sekä käyttäjien voimin tehtävällä loppukäyttäjätestauksella.

Koska ketterässä kehityksessä vaatimukset voivat muuttua projektin edetessä, jatkuva testaus mahdollistaa nopeat muutokset ohjelmistoon ilman, että laatu kärsii. Testausprosessien avulla voidaan varmistaa, että muutokset integroituvat sujuvasti olemassa olevaan järjestelmään. Käytännössä tämä kuitenkin tarkoittaa, että testaukseen käytetty aika on monesti selkeästi suurempi kuin perinteisessä vesiputousmallissa.



Käyttäjätestauksen osalta hyvin määritellyt käyttäjätarinat edesauttavat testitapausten luomista, koska esimerkiksi hyväksymiskriteerien pohjalta pystytään luomaan testitapausten runko. Yleisimmät testitapaukset ovat etenkin suuremmissa projekteissa hyvä pyrkiä automatisoimaan, koska testit tulisi ajaa jokaisessa sprintissä varmistaen, ettei kehityksessä ole rikkoutunut mitään aiempaa toiminnallisuutta. Automaatiotestaustyökaluksi valikoitui Robot Framework, jonka esimerkkitestiraportti näkyy kuvassa 26.

## ERP Report

Generated  
20240421 03:04:54 UTC+03:00  
11 hours 28 minutes ago

### Summary Information

**Status:** 7 tests failed

**Start Time:** 20240421 03:02:33.976

**End Time:** 20240421 03:04:53.936

**Elapsed Time:** 00:02:19.960

**Log File:** [log.html](#)

### Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	21	14	7	0	00:01:58	<div style="width: 100%; height: 10px; background: linear-gradient(to right, green 66.7%, red 66.7%, red 100%);"></div>

Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
4384	9	7	2	0	00:00:33	<div style="width: 100%; height: 10px; background: linear-gradient(to right, green 77.8%, red 77.8%, red 100%);"></div>
6532	1	1	0	0	00:00:02	<div style="width: 100%; height: 10px; background: linear-gradient(to right, green 100%, red 100%, red 100%);"></div>
6533	5	5	0	0	00:00:10	<div style="width: 100%; height: 10px; background: linear-gradient(to right, green 100%, red 100%, red 100%);"></div>
9186	3	0	3	0	00:00:44	<div style="width: 100%; height: 10px; background: linear-gradient(to right, red 100%, red 100%, red 100%);"></div>
9241	2	1	1	0	00:00:16	<div style="width: 100%; height: 10px; background: linear-gradient(to right, green 50%, red 50%, red 100%);"></div>
regression_erp	21	14	7	0	00:01:58	<div style="width: 100%; height: 10px; background: linear-gradient(to right, green 66.7%, red 66.7%, red 100%);"></div>

Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
ERP	21	14	7	0	00:02:20	<div style="width: 100%; height: 10px; background: linear-gradient(to right, green 66.7%, red 66.7%, red 100%);"></div>
ERP.4384 Testitapaukset Tuotevalikoima Taksa	11	8	3	0	00:01:05	<div style="width: 100%; height: 10px; background: linear-gradient(to right, green 72.7%, red 72.7%, red 100%);"></div>
ERP.6530 Testitapaukset ERPn autentikointi	6	6	0	0	00:00:14	<div style="width: 100%; height: 10px; background: linear-gradient(to right, green 100%, red 100%, red 100%);"></div>
ERP.9505 Testitapaukset Tilausten Validointi	3	0	3	0	00:00:48	<div style="width: 100%; height: 10px; background: linear-gradient(to right, red 100%, red 100%, red 100%);"></div>
ERP.ERP	1	0	1	0	00:00:13	<div style="width: 100%; height: 10px; background: linear-gradient(to right, red 100%, red 100%, red 100%);"></div>

### Test Details

All Tags Suites Search

Suite:

Test:

Include:

Exclude:

[Help](#)

Kuva 25. Testiraportti automaatiotesteistä

Automaatiotestit on rakennettu perustuen liiketoimintavaatimusten käyttäjätarinoihin. Näin ollen raportista on suoraan nähtävissä, mitä käyttäjätarinoita

kyseisissä testeissä on testattu sekä mitkä testit ovat menneet läpi onnistuneesti ja mitkä testit taasen epäonnistuneet.

### 4.3 Käyttöönotto

Loppuvaiheessa, kun järjestelmä on valmis käyttöönottoon, tulee käyttöön enemmän perinteisen vesiputousmallin mukaisia työkaluja. Tämä siitä syystä, että käyttöönoton sisältö ja aikataulut on järkevämpää tehdä perinteisen projektin muotoisesti, koska lopputuotos on melko tarkasti määritelty eikä siihen lähtökohtaisesti tule enää käyttöönoton edetessä isoja muutoksia. Käyttöönotto-vaihe muodostuu Pharmacilla ensimmäisessä vaiheessa seuraavanlaisista osioista.

- Tekniset valmisteluissa tulisi varmistaa, että kaikki järjestelmän osat ovat valmiina käyttöönottoa varten. Tämä voi sisältää lopulliset testaukset, kuten integraatio- ja hyväksymistestaukset, sekä järjestelmän asennuksen tuotantoympäristöön.
- Käyttöönottosuunnitelmaa laadittaessa tulisi tuottaa yksityiskohtainen suunnitelma käyttöönotolle, mukaan lukien aikataulu, resurssit, vastuuhenkilöt ja muut toimenpiteet, joita tarvitaan onnistuneeseen siirtymiseen.
- Tuleville järjestelmän loppu- ja pääkäyttäjille tulisi valmistella riittävä koulutus ja ohjaus. Koulutuksen tavoitteena on varmistaa, että kaikki käyttäjät ymmärtävät, kuinka uutta järjestelmää käytetään ja miten se tukee heidän päivittäistä työtään. Muutosvastarinnan vähentämiseen kannattaa miettiä esim. muutosagenttien käyttämistä.
- Tietojen migrointi vanhoista järjestelmistä ja integroinnit muihin järjestelmiin tulisi varmistaa ja testata tämä erityisellä huolella pyrkien kattamaan myös yllättävät käyttötilanteet.
- Uuden järjestelmän tekninen käyttöönotto kannattaa aikatauluttaa tarkalla tasolla ja listata kaikki toimenpiteet sekä niihin liittyvä viestintä ja tiedottaminen. Tämä etenkin, jos tekninen käyttöönotto tehdään kerralla, eikä sitä ole mahdollista tai järkevää paloitella pienempiin osiin.

Käyttöönoton jälkeen seuraisi jakso, jossa järjestelmän suorituskykyä, toimintaa ja toimivuutta seurattaisiin erityisellä huolella. Jakson aikana varattaisiin sekä

teknisiä asiantuntijoita että tukihenkilöitä hoitamaan mahdollisia ongelmatilanteita tehokkaasti.

#### 4.4 Ylläpito ja kehitys

Järjestelmän käyttöönoton jälkeen siirtyisi tekeminen jatkuvaan kehitys- ja ylläpitovaiheeseen. Tämä osa toteutettaisiin luvun 4.2. mukaisesti ketterillä menetelmillä käyttäen samoja työkaluja ja malleja. Sprinteillä toteutettaisiin uusia ominaisuuksia, mutta toisaalta myös korjattaisiin tuotantoympäristössä havaittuja bugeja. Uudet ominaisuudet paketoitaisiin järkevän kokoisiin ja Pharmacille soveltuvien välein julkaistaviksi paketeiksi. Kriittisten bugikorjausten osalta toiminnallisuudet tulisi pystyä viemään tuotantoon kiireellisinä ohi varsinaisten uusien julkaisujen. Tässä vaiheessa automaatiotestausten merkitys kasvaa entisestään, koska aina uusia ominaisuuksia julkaistaessa olisi hyvä testata koko järjestelmän perusominaisuudet, ettei julkaisussa ole rikkoutunut mitään aiempia toiminnallisuuksia.

## 5 Loppusanat

Insinööriyön tavoitteena oli miettiä Pharmacin ohjelmistokehitykseen soveltuva malli ja luoda sen toteuttamiseen työkalupohjia tilaus-, toimitus- ja laskutusjärjestelmän toteutukseen, mutta toisaalta myös Pharmacin tulevaan sovelluskehitykseen. Malli muodostui hybridiratkaisu, joka hyödynsi toisaalta niin perinteistä vesiputousmallia kuin ketterää Scrum-mallia.

Työstä muodostui suhteellisen laaja kokonaisuus ja etenkin Pharmacin toimintaan vaikuttavan regulaation laajuus ja kompleksisuus oli melko haastavaa kuvata tiiviisti ja ymmärrettävästi. Parhaiden käytäntöjen ja Pharmacille sopivien ohjelmistokehitysmallien ja työkalujen luonti taasen oli selkeämpää. Tässä auttoi paljon projektityöryhmän osallistuminen eri tarpeiden kartoitukseen tiedonkeruuvaiheessa.

Insinööriyössä toteutetun mallin jatkokehityssaihioksi tunnistettiin, että vaatimusmäärittelyssä kuvatut vaatimukset tulisi linkittää niihin regulaatioihin, joiden pohjalta vaatimukset on luotu. Näin ollen, jos regulaatio muuttuu, tunnistettaisiin myös toiminnallisuudet, joita muutos koskee. Yksi ratkaisu tähän voisi olla ns. RegOps-malli.

RegOps, lyhennys sanoista "regulation" ja "operations", viittaa vaatimustenmukaisuuden täyttämiseen käyttäen apuna muun muassa data-analytiikkaa, suunnittelulähtöistä ajattelua ja tekoälyä. RegOps hyödyntää automaatioita ja sen tavoitteena on tarjota lähes reaaliaikaisia ratkaisuja säädösten noudattamiseen. [29.] Malli on suhteellisen tuore, eikä sitä ole vielä työstetty konkretiaan saakka, vaan se vaikuttaa vielä enemmän konseptilta. Data-analytiikan ja tekoälyn kehityksessä, voi mallista tulla suuri apu vaatimuksenmukaisuuden täyttämiseen.

Haluan esittää suuret kiitokset Pharmac Finland Oy:lle ja sen toimitusjohtaja Karoliina Kaijasilta-Järvenpäälle mahdollisuudesta toteuttaa insinööriyö Pharmacille ja mahdollisuudesta kokeilla ja soveltaa eri mallien sekä työkalujen toimivuutta oikeassa käytössä. Erityiskiitokset haluan esittää Pharmacin tuotantopäällikölle ja tukkuliikkeen vastuunalaiselle johtajalle Johanna Vesteriselle, joka toimi myös käynnissä olevan projektin toisena Product Ownerina. Sain häneltä kokonaisvaltaisesti paljon palautetta ja kehitysideoita insinööriyöhön.

## Lähteet

- 1 Pharmac Finland Oy:n sisäiset materiaalit.
- 2 10.4.1987/395. Lääkelaki. 1987. <<https://www.finlex.fi/fi/laki/ajantasa/1987/19870395>>.
- 3 Sosiaali- ja terveydenhuollon tietojärjestelmät. Verkkoaineisto. Sosiaali- ja terveysalan lupa- ja valvontavirasto. <<https://valvira.fi/sosiaali-ja-terveydenhuollon-tietojarjestelmat>>. Luettu 10.02.2024.
- 4 Tiedonhallinta sosiaali- ja terveysalalla. Verkkoaineisto. Terveyden ja hyvinvoinnin laitos. <<https://thl.fi/aiheet/tiedonhallinta-sosiaali-ja-terveysalalla>>. Luettu 11.02.2024.
- 5 784/2021. Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä. 2021. <<https://www.finlex.fi/fi/laki/alkup/2021/20210784>>.
- 6 Asetus (EU) 2016/679 Yleinen tietosuojasetus (GDPR). 2016. <<https://eur-lex.europa.eu/Fl/legal-content/summary/general-data-protection-regulation-gdpr.html>>.
- 7 Mikkola H, Sinnemäki J, Hämeen-Anttila K, Laukkanen E, Reinikainen L. Lääkealan turvallisuus- ja kehittämiskeskus Fimea. 2022. Lääkkeiden koineellisen annosjakelun nykytila ja kehittämistarpeet <<https://urn.fi/URN:ISBN:978-952-7299-37-1>>.
- 8 Määräys 6/2011. Apteekkien lääkevalmistus. Lääkealan turvallisuus- ja kehittämiskeskus Fimea. 2011. <<https://www.finlex.fi/fi/viranomaiset/normi/558001/38373>>.
- 9 Good manufacturing practice. Verkkoaineisto. European Medicines Agency EMA. <<https://www.ema.europa.eu/en/human-regulatory-overview/research-development/compliance-research-development/good-manufacturing-practice>>. Luettu 17.02.2024.
- 10 2013/C 68/01 (EU). Ohjeet ihmisille tarkoitettujen lääkkeiden hyvistä jake-lutavoista. 2013. <<https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:C:2013:068:0001:0014:FI:PDF>>.
- 11 Lääkkeiden hinnat ja korvaukset. Suomen apteekkariliitto. Verkkoaineisto. <<https://www.apteekki.fi/laakkeiden-hinnat-ja-korvaukset.html>>. Luettu 18.02.2024.

- 12 Lääkkeiden hintalautakunta. Verkkoaineisto. Lääkkeiden hintalautakunta Hila. <<https://www.hila.fi>>. Luettu 18.02.2024.
- 13 Lääketaksa. Verkkoaineisto. Suomen apteekkariliitto <<https://www.apteekkariliitto.fi/apteekkitieto/laaketaksa.html>>. Luettu 18.02.2024.
- 14 30.12.2010/1326. Terveystuotolaki. 2010. <<https://finlex.fi/fi/laki/ajantasa/2010/20101326>>.
- 15 744/1990. Asetus yksityisestä terveydenhuollosta. 1990. <<https://finlex.fi/fi/laki/alkup/1990/19900744>>.
- 16 17.8.1992/785. Laki potilaan asemasta ja oikeuksista 1992. <<https://www.finlex.fi/fi/laki/ajantasa/1992/19920785>>.
- 17 28.6.1994/559. Laki terveydenhuollon ammattihenkilöistä. 1994. <<https://www.finlex.fi/fi/laki/ajantasa/1994/19940559>>.
- 18 3.8.1992/734. Laki sosiaali- ja terveydenhuollon asiakasmaksuista. 1992. <<https://www.finlex.fi/fi/laki/ajantasa/1992/19920734>>.
- 19 Sosiaali- ja terveydenhuollon tietojärjestelmien luokittelu. Verkkoaineisto. Sosiaali- ja terveysalan lupa- ja valvontavirasto. <<https://valvira.fi/sosiaali-ja-terveydenhuolto/tietojarjestelmien-luokittelu>>. Luettu 10.02.2024.
- 20 Olennaiset vaatimukset ja sertifiointi. Verkkoaineisto. Terveystuot ja hyvinvoinnin laitos. <<https://thl.fi/aiheet/tiedonhallinta-sosiaali-ja-terveysalalla/tiedonhallinnan-ohjaus/olennaiset-vaatimukset-ja-sertifiointi>>. Luettu 11.02.2024.
- 21 Sosiaali- ja terveydenhuollon tietojärjestelmärekisteri. Verkkoaineisto. Sosiaali- ja terveysalan lupa- ja valvontavirasto. <<https://valvira.fi/sosiaali-ja-terveydenhuolto/tietojarjestelmarekisteri>>. Luettu 10.02.2024.
- 22 Rajaniemi S. Lääkehoidon ohjauksen kehittäminen. Sosiaali- ja terveysministeriö. 2023. <[https://stm.fi/documents/1271139/2013549/Selvitys+SR+15.1.23\\_final.pdf/000f9c8b-159b-5c32-5874-4c204a9c610f/Selvitys+SR+15.1.23\\_final.pdf?t=1675761745353](https://stm.fi/documents/1271139/2013549/Selvitys+SR+15.1.23_final.pdf/000f9c8b-159b-5c32-5874-4c204a9c610f/Selvitys+SR+15.1.23_final.pdf?t=1675761745353)>.
- 23 Avohuollon apteekkitoiminnan kehittämistarpeet. Sosiaali- ja terveysministeriö. 2011. <<http://urn.fi/URN:ISBN:978-952-00-3152-7>>.
- 24 Apteekkitoiminnan ja muun lääkehuollon kehittäminen: Työryhmän loppuraportti. Sosiaali- ja terveysministeriö. 2014. <<http://urn.fi/URN:ISBN:978-952-00-3554-9>>.

- 25 Sommerville I. 9<sup>th</sup> edition. Software Engineering. Addison-Wesley. 2011. <<https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-lan-Sommerville.pdf>>.
- 26 Manifesto for Agile Software Development. Verkkoaineisto. <<https://agilemanifesto.org>>. Luettu 17.02.2024.
- 27 What is Scrum? Verkkoaineisto. Scrum.org. <<https://www.scrum.org/learning-series/what-is-scrum>>. Luettu 17.02.2024.
- 28 VH/483/00.01.00/2023. Apteekkiveron syventävät ohjeet. Verohallinto. 2023. <<https://www.vero.fi/syventavat-vero-ohjeet/ohje-haku-sivu/48607/apteekkivero>>.
- 29 Revolutionizing Compliance with RegOps. Verkkoaineisto. The Compliance Evangelist <<https://compliancepodcastnetwork.net/revolutionizing-compliance-with-regops>>. Luettu 13.04.2024.

## Työpajan runko – Tietosuojavastaava

Tietosuoja-roolin työpajaan valmisteltu runko läpikäytävälle asiaille.

1. Miten järjestelmä vaikuttaa henkilötietojen käsittelyyn?
  - a. Onko järjestelmään tallennettavien henkilötietojen laajuus ja luonne muuttumassa?
  - b. Millä tavoin järjestelmä varmistaa henkilötietojen luokittelun ja käsittelyn GDPR:n mukaisesti?
2. Mitä GDPR:n vaatimuksia meidän on erityisesti huomioitava järjestelmää suunniteltaessa?
  - a. Onko olemassa erityisiä tietosuojakäytäntöjä, jotka vaikuttavat järjestelmän suunnitteluun tai toiminnallisuuteen?
3. Mitkä ovat suurimmat tietosuojan riskit järjestelmän osalta ja kuinka me voimme niitä hallita?
  - a. Onko riskien arviointi tehty, ja miten se vaikuttaa vaatimusmäärittelyyn?
4. Miten tietojen säilyttämisen ja poistamisen politiikat toteutetaan uudessa järjestelmässä?
  - a. Kuinka tietojen minimointia ja elinkaaren hallintaa suunnitellaan toteutettavan?
5. Miten rekisteröidyn oikeudet toteutetaan järjestelmässä?
  - a. Miten rekisteröidyn oikeus tarkastaa, korjata ja poistaa omat tietonsa toteutetaan?
6. Millaisia tietoturva-vaatimuksia ja tietosuojan toimenpiteitä järjestelmän tulisi sisältää?
  - a. Miten varmistamme, että tietoturva on integroitu järjestelmän suunnitteluun alusta alkaen (Privacy by Design)?
7. Kuinka järjestelmä mahdollistaa tietojen siirrettävyyden ja miten se vastaa tiedonhallinnan vaatimuksiin?
  - a. Onko järjestelmässä toimintoja, jotka tukevat tiedon helppoa siirrettävyyttä toisille järjestelmille tai palveluntarjoajille?
8. Kuinka järjestelmä dokumentoi henkilötietojen käsittelyn ja käyttäjien toiminnot?
  - a. Millaisia lokitietoja ja auditointimahdollisuuksia järjestelmä tarjoaa?
9. Millainen on järjestelmän käyttäjien koulutus- ja tietoisuusohjelma tietosuoja-asioissa?
  - a. Miten varmistetaan, että henkilökunta ymmärtää heidän roolinsa henkilötietojen käsittelyssä ja tietosuoja-asioissa?
10. Miten tietosuojavastaavan rooli on määritelty järjestelmän ylläpidossa ja kehityksessä?
  - a. Miten tietosuojavastaavan neuvoja ja ohjeita hyödynnetään jatkuvasti järjestelmän kehityksessä?
11. Onko olemassa erityisiä toimialakohtaisia vaatimuksia tai standardointeja, jotka meidän tulee ottaa huomioon?
  - a. Ovatko terveydenhuollon alan tietosuojavaatimukset vaikuttaneet järjestelmän toimintalogiikkaan?



## Työpajan runko – IT-arkkitehti

IT-arkkitehti-roolin työpajaan valmisteltu runko läpikäytävälle asiaille.

1. Miten järjestelmän arkkitehtuuri tukee liiketoiminnan prosesseja ja tarpeita?
  - a. Millaisia pääkomponentteja ja integraatioita uuteen järjestelmään suunnitellaan?
2. Millaisia teknisiä haasteita järjestelmän toteuttamisessa on ennakoitu?
  - a. Mitä ratkaisuja näihin haasteisiin on harkittu?
3. Kuinka järjestelmän modulaarisuus ja skaalautuvuus on otettu huomioon suunnittelussa?
  - a. Miten järjestelmä mukautuu tulevaisuuden laajennuksiin tai toiminnallisuuksiin?
4. Kuinka järjestelmä varmistaa tietojen yhteentoimivuuden olemassa olevien järjestelmien kanssa?
  - a. Millaisia rajapintoja suunnitellaan käytettäväksi?
5. Miten järjestelmän suorituskykyä ja vasteaikoja seurataan ja optimoidaan?
  - a. Tullaanko suorituskykyä ja vasteaikoja seuraamaan millaisilla kuormitustesteillä?
6. Kuinka järjestelmä tukee tietoturvan ja tietosuojan vaatimuksia?
  - a. Mitä teknisiä ratkaisuja tietoturvaan on otettu käyttöön?
7. Millaisia toipumissuunnitelmia ja katastrofipalautusta varten on suunniteltu?
  - a. Miten järjestelmän toimintavarmuus ja jatkuvuus varmistetaan häiriötilanteissa?
8. Kuinka järjestelmän tiedonhallinta ja datan laatu varmistetaan?
  - a. Millaisia menetelmiä ja työkaluja käytetään datan laadunvarmistukseen?
9. Millaisia käyttäjän hallintaratkaisuja on suunniteltu?
  - a. Kuinka järjestelmä tukee eri käyttäjäroolien hallintaa ja oikeuksien määrittelyä?
10. Kuinka järjestelmän ylläpitotoimenpiteitä on suunniteltu hallittavaksi?
  - a. Millainen on ylläpidon ja päivitysten hallinnan prosessi?
11. Mitä vaatimuksia on asetettu järjestelmän dokumentaatiolle ja miten sen ajantasaisuus varmistetaan?
  - a. Onko olemassa standardoitu prosessi dokumentaation päivittämiselle?
12. Kuinka pilvipalveluiden ja ulkoisten palveluntarjoajien käyttö integroidaan arkkitehtuuriin?
  - a. Miten pilvipalvelujen tietoturva ja järjestelmän suorituskyky varmistetaan?
13. Miten järjestelmän käyttöönottoon ja käyttöönoton jälkeiseen vaiheeseen valmistaudutaan arkkitehtuurin näkökulmasta?
  - a. Onko käyttöönoton jälkeistä monitorointia ja optimointia varten suunnitelmia?

## Työpajan runko – Product Owner

Product Owner-roolin työpajaan valmisteltu runko läpikäytävillä asioille.

1. Mikä on järjestelmän päämäärä ja liiketoiminnalliset tavoitteet?
  - a. Miten tämä järjestelmä tukee yrityksen strategisia tavoitteita ja liiketoiminnan kasvua?
2. Mitkä ovat kriittisimmät toiminnot, jotka järjestelmän tulee tukea?
  - a. Onko erityisiä prosesseja, jotka vaativat erityishuomiota järjestelmäsuunnittelussa?
3. Miten määrittelet menestyksen järjestelmän käyttöönotossa?
  - a. Mitä mittareita käytetään arvioimaan järjestelmän onnistumista?
4. Mitkä ovat tärkeimmät sidosryhmät ja heidän tarpeensa, jotka tulee ottaa huomioon järjestelmässä?
  - a. Kuinka heidän vaatimuksensa on kerätty ja priorisoitu?
5. Mitä riskejä näet järjestelmän kehityksessä ja miten niitä hallitaan?
  - a. Onko olemassa suunnitelmia riskien minimoimiseksi projektin aikana?
6. Miten järjestelmän kehitys ja käyttöönotto aikataulutetaan?
  - a. Mikä on projektin aikataulu ja miten muutokset hallitaan projektin aikana?
7. Millainen on käyttäjien koulutus- ja tukisuunnitelma järjestelmän osalta?
  - a. Miten varmistetaan, että käyttäjät ovat valmiita käyttämään järjestelmää tehokkaasti sen käyttöönoton jälkeen?
8. Miten järjestelmän kehitysprosessi on organisoitu?
  - a. Kuka kuuluu projektitiimiin ja miten päätöksenteko on järjestetty?
9. Miten käsittelet järjestelmän jatkuvaa parantamista ja päivityksiä?
  - a. Onko olemassa prosessi järjestelmän jatkuvan kehittämisen ja käyttäjäpalautteen integroimiseksi?
10. Miten järjestelmän ROI (return on investment) arvioidaan?
  - a. Miten varmistetaan, että investointi järjestelmään tuottaa odotetut hyödyt?
11. Onko järjestelmässä ominaisuuksia, jotka tukevat skaalautuvuutta ja kansainvälistä kasvua?
  - a. Miten järjestelmä sopeutuu mahdolliseen laajentumiseen ja monimutkaistumiseen tulevaisuudessa?
12. Miten tietosuojavaatimukset, kuten GDPR, on otettu huomioon järjestelmän suunnittelussa?
  - a. Mitä toimenpiteitä on suunniteltu henkilötietojen suojaamiseksi?

## Työpajan runko – Hankepääällikkö

Hankepääällikkö-roolin työpajaan valmisteltu runko läpikäytävillä asioille.

1. Mikä on projektin laajuus ja mitkä ovat sen päätavoitteet?
  - a. Voisitko määritellä projektin keskeiset toimitettavat osat ja niiden odotetut hyödyt?
2. Millainen on projektin aikataulu ja merkittävät virstanpylväät?
  - a. Miten olet aikatauluttanut projektin kriittiset vaiheet ja miten varmistat, että aikataulu pitää?
3. Millaisia resursseja ja budjettia projektissa käytetään?
  - a. Miten budjetti on jaettu ja millaisia resursseja tarvitaan projektin eri vaiheissa?
4. Mitkä ovat projektin suurimmat riskit ja miten niitä hallitaan?
  - a. Onko riskienhallintasuunnitelma valmis ja miten se päivitetään projektin edetessä?
5. Miten muutokset projektin laajuudessa ja suunnitelmassa käsitellään?
  - a. Onko olemassa selkeä muutoshallintaprosessi ja kuka on vastuussa muutosten hyväksymisestä?
6. Miten sidosryhmien odotuksia hallitaan ja viestintä järjestetään?
  - a. Mitkä ovat keskeiset sidosryhmät ja miten heidän kanssaan viestitään projektin edistymisestä?
7. Miten projektin laatu varmistetaan ja mitä laadunvarmistusmenetelmiä käytetään?
  - a. Mitkä ovat projektin laadunvarmistuksen keskeiset indikaattorit ja menetelmät?
8. Miten projektitiimi on organisoitu ja miten tiimin suorituskykyä seurataan?
  - a. Millaisia rooleja tiimissä on ja miten varmistetaan, että jokainen tiimin jäsen tietää vastuualueensa?
9. Mitkä ovat projektin tärkeimmät säännölliset raportointikäytännöt?
  - a. Kuinka usein projektin tilanne raportoidaan ja kenelle?
10. Millainen on projektin päätösprosessi ja miten onnistumista arvioidaan?
  - a. Millaisia kriteereitä käytetään arvioitaessa projektin onnistumista sen päättyessä?
11. Miten käyttöönotto suunnitellaan ja toteutetaan?
  - a. Millaiset toimenpiteet on suunniteltu järjestelmän käyttöönottoa varten ja miten käyttäjien koulutus järjestetään?
12. Onko olemassa suunnitelmaa projektin jälkeiselle tuelle ja ylläpidolle?
  - a. Miten varmistetaan, että järjestelmä toimii optimaalisesti käyttöönoton jälkeen ja miten jatkokehitys on suunniteltu?