Eero Laineen kandidaattityö [Johtaminen perinteisissä ja ketterissä ohjelmistotuotantoprojekteissa](https://www.cs.helsinki.fi/u/mluukkai/ohtu/laine-kandi.pdf)

Laineen kandissa käydään läpi erilaisia johtamistapoja perinteisissä ja ketterissä menetelmissä. Laine käy läpi johtamistyylien eroja näiden kahden menelelmän välillä. Perinteisissä menetelmissä projektilla on oma projektipäällikkö jolla on iso rooli ja vastuu projektista. Huomattiin kuitenkin että päällikölle asetetut vaatimukset eivät yleensä toteutuneet ja huomattiin että tämä hidasti projektien valmistumista.

Ketterissä menetelmissä valta on jaettu eri roolien kesken, ja roolit määräytyvät usein asiantuntijuuden mukaan. Tiimit toimivat mahdollisimman autonomisesti ja ne tekevät päätökset yhdessä.

Kenny Heinosen kandidaattityö [Ohjelmistoala ja ryhmätyöskentely](https://www.cs.helsinki.fi/u/mluukkai/ohtu/heinononen-kandi.pdf)

Heinonen käsittelee kandissaan ryhmätyöskentelyä ohjelmistoprojekteissa. Ohjelmistoprojektit ovat laajoja kokonaisuuksia joissa tarvitaan asiantuntemusta lukuisista ohjelmointimenetelmistä. Tämän takia on luontevaa tehdä projektit ryhmätyönä jolloiin yksittäisen kehIttäjän ei tarvitse tuntea kaikkia projektissa käytettäviä menetelmiä. Ryhmätyöskentely vaatii hyviä vuorovaikutustaitoja jotta ryhmän kommunikointi olisi mahdollisimman tehokasta. Tämä on huomattu ja koulutusohjelmissa harjoitellaankin ryhmätyöskentelyä.

Esa Kortelaisen kandidaattityö [Jatkuva eksperimentointi ohjelmistokehityksen tukena](https://www.cs.helsinki.fi/u/mluukkai/ohtu/kortelainen-kandi.pdf)

Kortelainen käsittelee kandissaan eksperimointia ohjelmistotuotannossa. Eksperimointi on menetelmä jossa etukäteissuunnittelua tehdään hyvin vähän. Sovelluskehityksen edetessä todetaan sitten mitkä käytänteet ovat hyviä ja mitkä huonoja ja työskentelytapaa muutetaan opitun kokemuksen pohjalta. Esimerkiksi uuden ominaisuuden käyttöönotossa voidaan valita osa sovelluksen käyttäjistä testaamaan uutta toimintoa ja saadun palautteen pohjalta tehdään päätöksiä. Tämä nopeuttaa päätöksentekoa kun heti saadaan arvokasta tietoa ja riski sille että rakennetaan jotain mitä asiakas ei ole tilannut pienenee merkittävästi.

Lauri Suomalaisen kandidaattityö [Ohjelmistotuotantomenetelmien kehittyminen 1950-luvulta nykypäivään](https://www.cs.helsinki.fi/u/mluukkai/ohtu/suomalainen-kandi.pdf)

Suomalainen käsittelee kandissaan ohjelmistotuotannon kehitystä 1950-luvulta tähän päivään. Ensin tarkastellaan vesiputousmallia, sen rakennetta ja historiaa. SItten käydään läpi mallin ongelmia. Näitä ovat mm. mallin jäykkyys joka johtuu liiallisesta suunnitelmallisuuden korostamisesta. Asiakkaan vaatimukset voivat muuttua vaikka päivittäin sillä mikä tänään on asiakkaalle arvokasta ei enää huomenna välttämättä ole. Lisäksi kaikkia vaatimuksia on mahdoton kartoittaa täydellisesti heti alussa. Tämän jälkeen Suomalainen siirtyy ketteriin menelmiin ja aloittaa tarkastelun spiraalimallista. Spiraalimallissa ominaista on kehittämisen ohjautuminen riskien ja niiden hallinan mukaan. Spiraalimallin heikkouksia ovat mm. mallin dokumentoinnin heikkous, osa mallissa ja sen vaiheissa käytettävistä tekniikoista ja toimintatavoista eivät ole kovin tarkkaan määritelty. Tämän jälkeen Scrumia ja sen etuja ja heikkouksia.

Kasper Hirvikosken kandidaattityö [Metriikat käytänteiden tukena ohjelmiston laadun arvioimisessa](https://www.cs.helsinki.fi/u/mluukkai/ohtu/hirvikoski-kandi.pdf)

Hirvikoski käsittelee kandissaan metriikoita ojhelmiston laadun arvioinnissa. Metriikoilla tarkoitetaan erilaisia tapoja joiden tehtävänä on arvioida ohjelmiston komponentteja sekä pystyä havaitsemaan ohjelmiston kriittiset osa-alueet. Metriikoita on useita erilaisia mm. koodikirnu, verkkoanalyysi, testikattavuus ja mutaatiotestaus.

Koodikirnulla mitataan ohjelmiston muutoksien vaikutusta ohjelmiston virheherkkyyteen. Verkkoanalyysillä mitataan ohjelmiston riippuvuuksien vaikutusta ohjelmiston virhealttiuteen. Testikattavuudella mitataan kuinka suuriosa ohjleman riveistä suoritetaan testien aikana. Mutaatiotestauksella tutkitaan ohjelmiston testien laadukkuutta. Metriikoita on tutkittu pitkään mutta tutkimustuloksia ei ole saatu vielä tehokkasti hyödynnettyä. Ne kaipaavat vielä kehitystä jotta ne voidaan ottaa osaksi ohjelmistokehitystä.

Kalle Ilveksen kandidaattityö [Scrumban-menetelmän käyttö ketterässä ohjelmistokehityksessä](https://www.cs.helsinki.fi/u/mluukkai/ohtu/ilves-kandi.pdf)

Ilves käy kandissaan läpi ketterät menelmät, Scrum, Kanban ja niiden yhdstelmän Scrumbanin. Nykypäivän nopeasti vaihtuvat vaatimukset tuottavat vaikeuksia aikarajoitteisten iteraatioiden noudattamisessa. Toisaalta Kanban antaa liikaa vapauksia kehitysryhmälle. Tämän takia on yhdistelty molempien meneltemien parhaat puolet Scrumbaniksi. Molemmilla meneltelmillä yhtenäisiä ajattelutapoja on Lean menelmästä tuttu jätteen minimointi. Scrumban sopii erityisesti projekteihin joilla on taipumusta muuttua hyvinkin nopeasti iteraatioiden välissä.