

# Muistiinpanot

## Perinteinen vesiputousmalli:

Vesiputousmalli etenee vaiheesta toiseen. Suurien ohjelmistojen vaiheet vaikea suunnitella etukäteen. Vesiputousmalli on helposti ymmärrettävä. Aikaisessa vaiheessa tehdyt virheet ovat vaikea korjata myöhemmin. Suunnitelma menee uusiksi, jos ongelmia ilmenee tai asiakas muuttaa vaatimuksiaan. Kattava suunnitelma alussa säästää myöhemmin rahaa ja aikaa. Tulokset nähdään vasta kun projekti on loppumaisillaan. Toimii hyvin pienissä projekteissa.

## Ketterän kehityksen yleiset piirteet:

Ketterä ohjelmistokehitys on joukko ohjelmistotuotantoprojekteissa käytettäviä menetelmiä. Agile Manifesto julistusta pidetään ketterän kehityksen perusmääritelmänä. Agile Manifestissa määritellään 4 tyypillistä arvoa ja 12 periaatetta.

12 periaatetta ovat: Tyydyttää asiakas, ottaa vastaan muuttuvat vaatimukset, toimittaa versioita toimivasta ohjelmistosta säännöllisesti, työskennellä yhdessä päivittäin, projektit rakennetaan motivoituneiden yksilöiden ympärille, toimivin tapa tiedon välittämiseksi on kasvokkain, toimiva ohjelmisto on edistymisen ensisijainen mittari, kestävä toimintatapa, laadun ja rakenteen huomiointi, yksinkertaisuus, parhaat vaatimukset ja suunnitelmat syntyvät tiimeissä ja tiimi tarkastelee säännöllisesti tehokkuutta.

4 TYPILLISTÄ ARVOA: yksilöitä ja vuorovaikutusta, toimivaa sovellusta, asiakasyhteistyötä ja muutokseen reagoimista.

## Scrum:

Scrum on projektinhallinnan viitekehys, jota käytetään yleisesti ketterässä ohjelmistokehityksessä. Voidaan soveltaa projektinhallinnassa. Koko ryhmä pyrkii etenemään yksikkönä ja toimimaan tiiviissä yhteistyössä.

Palaverit ovat osa Scrumia. Suunnittelupalaveri kestää enintään 8 tuntia noin kuukauden mittaiselle sprintille. Päiväpalaverille ollaan varattu enintään 15 minuuttia, jossa luodaan suunnitelma seuraavalle 24 tunnille ja katsotaan miten työt ovat edistyneet.

Scrumin vaiheet ovat: Sprintti, Sprintin suunnittelupalaveri, Päiväpalaveri, Tuotteen kehitysjonon työstö, Sprinttikatselmus ja Sprintin retrospektiivi.

Tuoteomistajan tehtävät ovat: Määrittää ja tiedottaa julkaisu, järjestää virstanpylvään julkaisusta, kommunikoi osakkaiden kanssa, tiedottaa osakkaille sovelluskehityksestä, neuvottelee tärkeysjärjestyksestä, laajuudesta, rahoituksesta, aikatauluista ja varmistaa, että tuotekehitys on läpinäkyvää ja selkeää.

Scrummasterin tehtävät ovat: Poistaa mahdolliset esteet, ryhmän valmentaminen, päivittäinen työ on tuottavaa, pelisääntöjen noudattaminen ja suojaa ryhmää uusilta vaatimuksilta ja antaa työrauhan sprintin ajaksi.

Kehitystiimi vastaa sprintin valitusta tuotteen kehityspolusta ja julkaisukelvollisesta tuoteversiosta. Kehitystiimin tehtävät ovat: Analyysi, suunnittelu, kehittäminen, testaus ja dokumentointi.

## **XP:**

XP painottaa muiden ketterien menetelmien tapaan mukautuvuutta enemmän kuin ennustettavuutta. Asiakasvaatimukset voidaan käydä läpi välittömästi ja tuoda kehitystyöhön. XP:ssä useiden toistuvien ohjelmistojulkaisujen ja lyhyiden kehityssyklien tarkoitus on parantaa tuottavuutta. XP perustuu viiteen ydinarvoon.

Pienet julistukset ovat usein julkaistuja ohjelma versioita, joita julkaistaan asiakkaille. Jokaisen version täytyy olla testattu ja toimiva ja on myös tärkeää saada asiakkailta ja käyttäjiltä palautetta ajoissa, sillä mitä nopeammin ongelmat saadaan tietoon, niin on enemmän aikaa korjata ne.

Pariohjelmoinnissa 2 henkilöä työskentelee yhdellä koneella ja tarkoituksena on saada laadukkaampaa koodia jossa vähemmän virheitä. Rooleina ovat ohjaaja joka tekee koodia ja navigoija joka auttaa ohjaajaa antamalla vinkkejä ja korjaamalla virheitä. Koodin yhteisomistus eli kaikki ohjelmoijista voi muokata vaan koodilohkoista ja korjata niitä. Yhteisomistus ei toimi ilman hyvää kommunikointia. Pitää myös varmistaa, että muutokset eivät aiheuta ristiriitoja. XP suosii yksinkertaista suunnittelua ja vertauskuvia. Vertauskuvien tarkoitus on saada asiakas ja ohjelmoijat ymmärtämään, miten systeemi toimii.

## **LEAN:**

Koodaamisessa jätteitä ovat esimerkiksi turhat koodinpätkät, koodi joka kirjoitetaan uudestaan lisäyksen takia tai poistamaan kokoaan. LEAN:in perustana on jätteen minimointi ja prosessin tuotannon maksimointi. Periaatteena on etsiä kaikki mahdolliset jätteet ja poistaa ne tehokkaasti.

Kaizen on termi, joka tarkoittaa hyvää muutosta ja siinä eliminoidaan jätteet yksi kerrallaan ja mahdollisimman pienin kustannuksin.

MVP eli Minimum Viable Product tarkoittaa prototyyppiversiota, jossa on kaikki asiakkaan pyytämät ominaisuudet.

Kanban on visuaalinen työkalu, jolla osoitetaan milloin tuotannon tulisi alkaa ja loppua.

Varmistaa että tuotannossa on riittävästi tarvikkeita ja muita työtehtäviä.

## **TDD (Test-driven development):**

Ensin luodaan uusi testitapaus sitten muokataan kehitettävää ohjelmaa niin, että se läpäisee uuden testin. Yksikkötestit kirjoitetaan pienissä osissa ennen varsinaista tuotantokoodia. Tällä pyritään parempaan suunnitteluun ja varmistumaan ohjelmiston oikeasta toiminnasta. Hyötyjä ovat testikoodin kirjoitus etukäteen. Jo olemassa olevia testejä suorittamalla huomataan, jos virheitä korjattaessa tulee tehneeksi uusia virheitä.

Mocking on yksikkötestausilmiö, joka auttaa testaamaan objekteja erikseen toisistaan korvaamalla riippuvaiset objektit monimutkaisella käyttäytymisellä, testiobjekteilla ja ennalta määritetyllä käytöksellä. Näitä testiobjekteja kutsutaan mock objekteiksi.

## **RUP:**

Oma esitys

## **Adaptive Software Development (ASD):**

ASD luotiin RAD:in (Rapid Application Development) ja CAS:in (Complex Adaptive System) pohjalta. Rakennettiin kolmen cyclin perusteella: Spekulaatio, Yhteistyö ja Oppiminen. Spekulaatiossa selvitetään mitä asiakas haluaa ja mitä hän tarvitsee. Yksi oletus on, että asiakas ei tiedä mitä hän haluaa. Kaiken saadun informaation perusteella rakennetaan koko projektin aikataulutus.

Yhteistyön avulla tasapainotetaan työmäärä. Projektin adaptoituminen erilaisiin tilanteisiin ja olosuhteisiin, delegointi ja projektin edistäminen ja työstäminen tapahtuu tässä välissä.

Oppimisessa tavoitellaan virheiden korjausta, designin rakentamista ja testausta. ASD:n kulmakivi on "Do it wrong the first time".

ASD on erittäin taipuva, sen vaatimukset ovat vähäiset ja se soveltuu opiskeluympäristöön.

ASD:n huono puoli on ajankäyttö. Se vie paljon aikaa.