**Vesiputousmalli**

* Suunnittelu ja toteutus menee vaiheesta toiseen.
* Jos suunnitelmat muuttuvat, koko malli menee uusiksi
* Aikaisemmat virheet hankala korjata
* Helposti ymmärrettävä

**Ketterä kehitys**

* Agile Finland edustaa Suomessa
* Ketterä ohjemistokehitys on joukko projekteissa käytettäviä menetelmiä.
* Agile manifestissa määritellään 4 tyypillistä arvoa ja 12 periaatetta.
* Yksilö ja vuorovaikutus, toimiva sovellus, asiakasyhteistyö, muutokseen reagointi.
* Periaatteita: Yksinkertaisuus, toimiva ohjelmisto, kasvokkain käytävä keskustelu.

**Scrum**

* Projektihallinan viitekehys, yleensä käytetään ketterässä ohjelmistokehityksessä.
* Kehitysprosessi suoritetaan alusta loppuun vahvasti lomittuneella vaiheistuksella.
* Ryhmän toiminta: Itseohjautuvia, tilanteisiin sopeutuvia, ja nopeita.
* Suunnittelupalaveri 8h noin kuukauden mittaiselle printille.
* Päiväpalaveri max 15 min, jossa kehitystiimi luovat suunnitelmansa seuraavalle 24 tunnille.

**XP**

* XP (Extreme programming) painottaa muiden ketterien menetelmien tapaan mukautuvuutta.
* Asiakasvaatimukset voidaan käydä läpi välittömästi.
* Lyhyet kehityssyklit ja toistuvat ohjelmistojulkaisut.
* Pienet julkaisut mahdollistavat asiakkaiden näkevän työn edistymisen.
* Julkaisut kerran viikossa.
* Pariohjelmoinnissa 2 henkilöä työskentelee samalla koneella.
* Tarkoituksena saada laadukkaampaa koodia, jossa on vähemmän virheitä.
* Koodin yhteisomistus tarkoittaa sitä, että kaikki ohjelmoijista voi muokata kaikkia koodin osia.
* Yhteisomistus ei toimi ilman hyvää kommunikointia, versionhallintaa ja jatkuvaa integrointia.

**LEAN**

* LEANIN perustana on minimoida ”jätteen” tuotto ja maksimoida prosessin tuotanto.
* LEANIN periaatteena on etsiä kaikki mahdolliset jätteet ja poistaa ne mahdollisimman tehokkaasti
* Kaizen on LEAN-termi joka tarkoittaa hyvää muutosta, jossa eliminoidaan jätteet mahdollisimman pienin kustannuksin.
* MVP, Minium Viable Product, jossa on kaikki tarvittavat ominaisuudet.
* Kanban on visuaalinen työkalu, jolla osoitetaan milloin tuotannon tulisi alkaa ja loppua.

**TDD**

* Ensin luodaan testi ja sitten muokataan ohjelmaa niin että se läpäisee testin.
* Kun testikoodi kirjoitetaan etukäteen, saadaan jatkuvasti kehittyvä testiverkosto.
* Jo olemassa olevia testejä suorittamalla huomataan, jos on tullut tehneeksi uusia virheitä.
* Mocking on yksikkötestausilmiö, joka auttaa testaamaan objekteja erikseen toisistaan korvaamalla riippuvaiset objektit monimutkaisella käyttäytymisellä, testiobjekteilla ja simuloidulla käyttäytymisellä.

**RUP**

* Ohjelmistokehityksen prosessikehys.
* Se ei ole itsenäinen prosessi vaan laajennettava kehys.
* Voimaantulovaiheessa katsotaan, onko järjestelmää järkevä toteuttaa.
* Kehittelyvaiheessa projekti alkaa ottaa muotoaan.
* Rakennusvaiheessa rakennetaan ohjelmistoa.
* Muutosvaiheessa pyritään saamaan järjestelmä tuotantoon.
* Rakennusosat: Roolit, tuotteet ja tehtävät.
* Parhaat käytännöt: suunnittele visuaalisemmin, valvo laatua, hallinnoi muutoksia.