

1. Perinteinen vesiputousmalli

Mallissa edetään vaiheesta toiseen

Kaikkia vaiheita ei voi suunnitella etukäteen. Ongelmien ilmestyessä malli pitää tehdä kokonaan uudestaan. Vaikea korjata virheitä.

Säästää aikaa ja raha. Helposti ymmärrettävä. Helppo seurata.

Käytetään koulussa esim. kurssisuunnitelmissa ja kouluprojekteissa.

2. Agile Manifest Alliance

Pidetään ketterän kehityksen perusmääritelmänä.

Ohjelmistotuotantoprojekteissa käytettäviä menetelmiä.

17 ketterän kehityksen kannattajaa sopivat yhteisestä perustasta.

Agilella on 12 pääperiaatetta.

3. SCRUM

Projektihallinnan viitekehitys jota käytetään ohjelmistokehityksessä. Voi soveltaa myös projektihallinnassa. Vaiheistus joka on vahvasti lomittunut.

Toimintaa jossa ryhmä pyrkii etenemään yksikkönä ja toimimaan tiiviissä yhteistyössä. Malli on tilanteisiin sopeutuvia, itseohjautuvia ja nopeita.

Suunnittelupalaverit kestävät enintään 8h. Päiväpalaverit enintään 15min, joissa tehdään suunnitelmat seuraavalle 42 tunnille.

Scrum vaiheet:

- sprint

- sprint suunnittelupalaveri

- Päiväpalaverit

- Kehitysjonon työstö

- sprinttikatselmus

- retrospekki

Tuoteomistajan tehtävät:

- Määrittelee

- Järjestää virstanpylväät

- Kommunikoi osakkaiden kanssa

- Tiedottaa osakkaille sovelluskehityksestä

Neuvottelee tärkeysjärjestyksestä

Varmistaa että kehitys on selkeää

Scrum masterin tehtävät:

poistaa esteet

ryhmän valmentaminen

pelisääntöjen noudattaminen

Antaa työrauhan ja suojaa uusilta vaatimuksilta

Kehitystiimi vastaa tuotteen kehityspolusta ja julkaisukelvollisuudesta.

Kehitystiimin tehtävät:

analyysi

suunnittelu

kehittäminen

testaus

dokumentointi

4. XP

Painottaa mukautuvuutta enemmän ennustavuutta.

Vaatimukset käydään läpi välittömästi ja otetaan mukaan kehitystyöhön.

XP:ssä on monia lyhyitä kehityssyklejä.

XP perustuu viiteen ydinarvoon.

Pienet julistukset ovat usein julkaistavia ohjelman versioita, joista asiakas näkee työn edistyksen.

Tämän ansiosta asiakkaat voivat antaa palautetta ajoissa, sillä mitä nopeammin tiedät ongelman, sitä enemmän saat aikaa sen korjaamiseen.

Pariohjelmoinnissa kaksi henkilöä työskentelee yhdellä tietokoneella. Tarkoituksena on se, että toinen ilmoittaa koodaajalle hänen virheensä, kun hän tekee sen.

Koodin yhteysomistuksessa kaikki voi muokata koodia ilman alkuperäisen kirjoittajankin lupaa.

Yhteisomistuksessa on tärkeää hyvä kommunikointi, jottei koodissa tule ristiriitoja.

Vertauskuvien avulla on tarkoitus saada asiakas, ohjelmoijat ja esimiehet ymmärtämään miten systeemi toimii.

5. TDD

Ensin luodaan uusi testitapaus ja sen jälkeen testataan, toimiiko se. Tällä pyritään varmistumaan ohjelmiston oikea toiminta.

Kun testikoodi kirjoitetaan etukäteen saadaan kehittyvä testiverkosto.

TDD:tä voi hyödyntää myös kouluprojekteissa.

Mocking on yksikkötestausilmiö, joka auttaa testaamaan jokaista objektia erikseen korvaamalla riippuvaiset objektit ennalta määritellyillä/simuloiduilla käytöksellä.

6. RUP

Se ei ole itsenäinen prosessi vaan laajennettava kehys, jota muokataan tarpeiden mukaan.

Voimaantulovaiheessa katsotaan, onko järjestelmä kannattava tehdä.

Kehittelyvaiheessa vähentää pahimmat havaitut riskit.

Rakennusvaiheessa rakennetaan ohjelmisto

Muutosvaiheessa saadaan järjestelmä tuotantoon ja loppukäyttäjän saataville.

Päärakennusosat:

Roolit: Työntekijälle jaetaan roolit osaamisen mukaan.

Tuotteet: Projektin tulos johon sisältyy prototyypit ja dokumentit.

Tehtävät: Jaetaan roolin mukaan.

RUP käytännöt:

Kehitä iteratiivisesti.

hallinnoi vaatimuksia.

Käytä komponentteja.

Suunnittele visuaalisesti.

Valvo laatua.

Hallinnoi muutoksia.

Hyödyt

Käyttää vesiputousmallin parhaimmat osat

Keskittyy dokumentoinnin tärkeyteen

Ongelmat

Raskasprosessinen

Hidas tiettyihin projekteihin

Riippuu liikaa osakkaiden palautteesta

Monimutkainen ymmärtää

7. ASD

Systeemi, joka sopii jatkuvasti tilanteisiin ja olosuhteisiin.

ASD:ssä on 3 sykliä jotka ovat Spekulaatio, Yhteistyö ja oppiminen.

Spekulaatiossa selvitetään mitä asiakas haluaa ja tarvitsee, jonka avulla rakennetaan projektin aikataulus.

Yhteistyössä määritellään työmäärän tasapainotus, soveltaminen tilanteisiin, delegointi, ja työtetään projekti.

Oppimisessa tavoitellaan virheiden korjausta, designin rakentamista ja testausta. Virheitä ei tarvitse pelätä.

ASD:tä on helppo soveltaa, koska periaate on erittäin joustava.

ASD:ssä ongelmana in ajan hallinta.