

SOFTERA

Pro²- projektitoimitus- malli



Pro²

Softeran Pro² on vuodesta 2003 saakka käytössä ollut projekti-toimitusprosessi, jolla ohjataan ohjelmistotuotantoa. Pro² perustuu Rational Unified Process-järjestelmään (RUP), joka on maailman-laajuisesti käytetty ja tunnustettu prosessimalli. RUP:in tehokkuus perustuu UML-mallinnuksen käyttöön, muutos-hallinnan huomioon ottavaan prosessiin sekä projektiin osallistuvien henkilöiden selkeään vastuunjakoon.

Pro²-prosessimallissa järjestelmä jaetaan osiin, jotka mallinnetaan ja joille tehdään riskianalyysi. Suunnittelu ja toteutus tehdään riskianalyysin osoittamassa järjestyksessä, aloittaen suurimman riskin omaavasta osakokonaisuudesta. Tällöin vaikeimpiin osiin vaadittavat muutokset tehdään jo projektin alkuvaiheessa, eivätkä ne vaikuta hidastavasti enää helpompien osien toteutukseen.

Käytännön työskentelyssä Pro² tarkoittaa selkeitä rooleja ja vastuunjakoa, jokaiseen työvaiheeseen soveltuvia työohjeita ja esimerkkejä sekä valmiita dokumentointi- ja ohjelmointimalleja.

Pro²:n hyödyt:

1. Ohjelmistojen laatu paranee
2. Aikataulut saadaan pitämään
3. Budjetti saadaan pitämään
4. Muutostenhallinta tehostuu
5. Projektin seuraaminen helpottuu

Jokainen Softeran työntekijä on koulutettu käyttämään Pro²:ta sekä siihen liittyviä työkaluja.

RUP, UML ja Pro²

Pro² on Softeran kehittämä, yksinkertaistettu versio Rationalin mittavasta RUP- ohjelmistokehitysprosessista. Prosessin tarkoituksena on antaa ohjeita ja työkaluja tehokkaalle ohjelmiston kehitykselle ja vähentää riskien vaikutuksia. RUP on lähes standardiksi noussut ja tehokkaaksi tunnustettu toimintatapa, joka perustuu UML-mallinnukseen (Unified Modelling Language).

UML ei ole riippuvainen työkaluista, ohjelmointikielestä tai prosessista. Pääasiassa sitä käytetään oliopohjaisten tietojärjestelmien ongelmien ja niiden ratkaisuiden visualisoinnissa.

RUP-prosessista on kevyitä ja raskaampia versioita eri sovellusalojen käyttöön, mutta yhteistä niille kaikille on jäljempänä lyhyesti esiteltävien osaprosessien ja kaavioiden noudattaminen kaikissa projektin vaiheissa.

RUP:sta johdetun Pro²:n kehittämisen lähtökohtana on Softeran oma osaaminen sekä lukuisien projektien läpiviennistä dokumentoitu best practises-”työkalupakki” yhdistettynä RUP:n systemaattiseen toimintatapaan.

Prosessi

Pro²:ssa järjestelmä jaetaan osiin kuten perinteisessä vesiputousmallissakin, jossa järjestelmä kuitenkin suunnitellaan, toteutetaan ja testataan yhtenä kokonaisuutena. Pro²:ssa jokainen tehtäväkokonaisuus vietään läpi peräkkäisillä iteraatioilla, joista jokainen muodostaa oman vesiputouksensa.

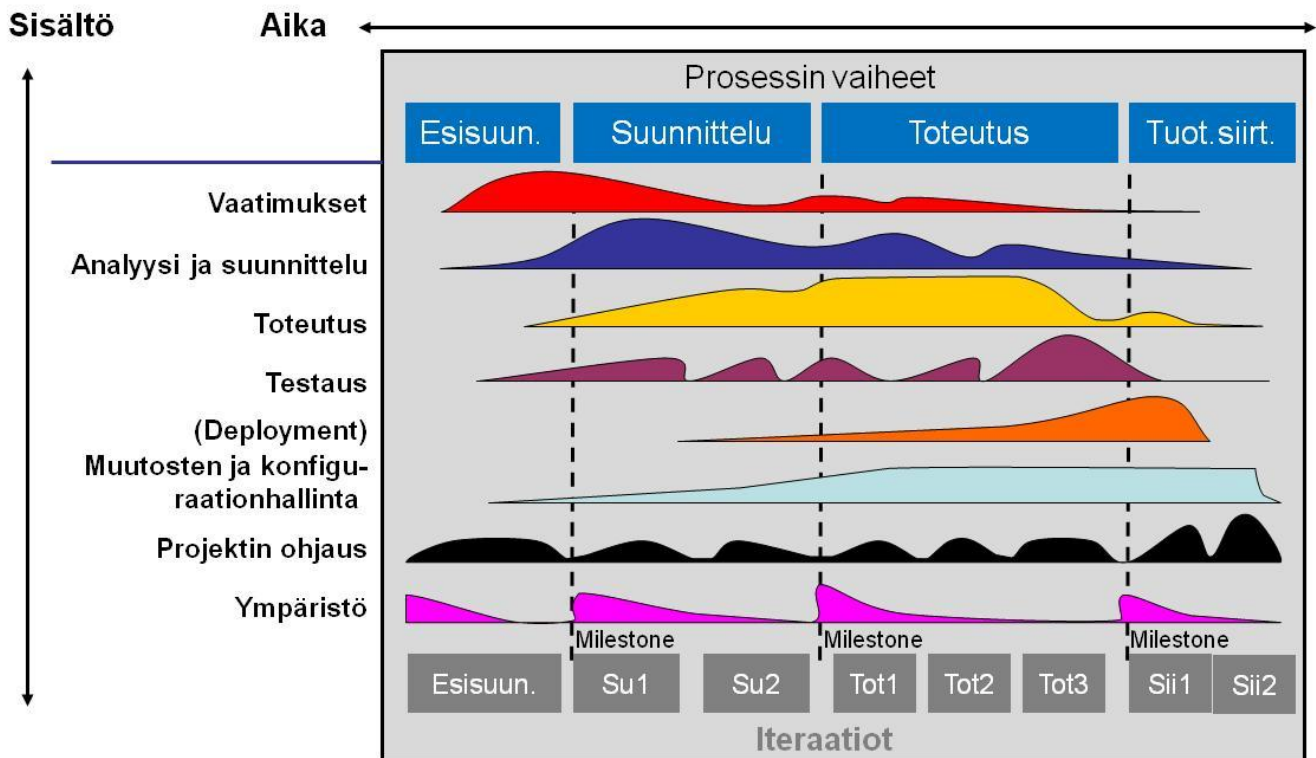
Tehtäväkokonaisuuksien lähtökohtana ovat selkokielisesti dokumentoidut käytötapa-ukset, jotka ohjaavat kehitystyötä läpi prosessin. Käytötapa-ukset kerätään ja kuvataan käyttäjän näkökulmasta ja ne kokoavat yhteen kaikki toiminnot, mitä järjestelmältä vaaditaan.

Projektin elinkaari

Ohjelmistoprojektin jokaisen osakokonaisuuden läpiviennissä on neljä yhteistä päävaihetta: alkuvaihe, suunnittelu-/tarkennusvaihe, toteutus/rakennusvaihe sekä siirtymävaihe eli tuotteen toimitus.

Jokaisesta vaiheesta kuvataan tavoitteet, aktiviteetit, työmäärä, ”virstanpylväät” sekä virstanpylvään evaluointikriteerit. Pro² ohjaa työmenetelmiä loogisesti ottaen huomioon, missä elinkaaren vaiheessa kukin osaprosessi on.

Piikkomalla työvaiheet elinkaarivaiheisiin (kuvattu pystysuoraan) sekä hallinnoimalla kutakin vaihetta (kuvattu vaakasuoraan) iteroimalla vältetään useimmat ohjelmistoprojektien tyypillisimmät sudenkuopat, kuten muutos-tarpeiden havaitsemisen myöhästyminen sekä modulien epäjohdonmukaiset toiminnallisuudet.



Iteratiivisuus

Iteratiivisuuden ideana on, että projektin vaiheet jaetaan pienemmiksi ja helpommin hallittaviksi ”miniprojekteiksi”. Menetelmä toimii erinomaisesti varsinkin silloin, kun kaikkia kehitettävän ohjelman vaatimuksia ei voida selvittää kehityksen alkuvaiheessa.

Iteratiivisella läpiviennillä loppuratkaisuun pyritään iteraatiosuunnitelman mukaisesti toteutettavien julkaisuiden kautta. Jokainen julkaisu sisältää täydentävää ja järjestelmän ominaisuuksia tarkentavaa kehittämistyötä.

Iteratiivisuuden etuina ovat testauksen helpottuminen, virheiden kumuloitumisen estäminen sekä muutosten hallinnan helpottuminen. Iteratiivinen toimintatapa antaa myös paremmat mahdollisuudet hallita kehitystyötä projektin aikana muuttuvien vaatimuksen tai yksityiskohtien osalta – kuten aina kuitenkin käy.