

TEEMA 1: Miten laatu nähdään ketterissä prosesseissa?

IDEOLOGIA

- * Virheitä tulee aina
- * Niiden kaikkien estäminen on mahdotonta
- * Tehdään virheiden **havaitseminen** ja **korjaus** mahdollisimman helpoksi

ASIAKKAAN NÄKÖKULMA

- * Ei näe toteutusta, muutoin kun käyttöliittymän kautta
- * Määrittelee laadun häiriöiden ja toiminnallisuuden tehokkuuden ja tarpeellisuuden mukaan
- * Ketterissä otetaan asiakas mukaan, ja pidetään mukana kehityksessä. Asiakkaan näkökulmaa korostetaan ja saadaan aiemmin palaute.

KEHITTÄJÄN NÄKÖKULMA

- * Ohjelmakoodin ja prosessin selkeys (suunnittelu, toteutus, ylläpidettävyyys)
- * Lyhyet syklit

TOIMIVA LAADUN MÄÄRITELMÄ PROSESSILLE

- * Ilmenee asiakkaan mukana ololla
- * Testit tehdään samanaikaisesti ohjelman kanssa
- * Laatu lisääntyy kun virhetyyppien realisointi vähenee
 - * Ketterä prosessi soveltuu nimenomaiseen projektiin, ei tarkoitettu kaikille projekteille. (myös muut työkalut)
 - * Kun kyseessä on pieni/keskisuuri projekti ketterällä prosessilla on mahdollista saada laadukas tuote.
 - * Dokumentaatio
 - Minimaalinen, mutta kompakti kokonaisuus, josta ei karsita tärkeitä asioita.
 - korjaamattomat virheet
 - * Kehittäjät
 - Vaihtuvuus pieni
 - Yksilöt huomioidaan paremmin
 - Kommunikaatio-ongelmat vähenee kehittäjien välillä

VIRHETYYPIT JA NIIHIN VARAUTUMINEN

- * Väärin määritellyt vaatimukset
 - Asiakas läsnä: epämuodollista katselmointia
 - Varautuu jo väärin vaatimuksiin ja antaa mahdollisuuden (heitetään pois ja huomataan ajoissa)
 - Asiakas saa useasti version ohjelmasta testattavaksi, jolloin aikaisin huomataan puutteet / väärät toiminnallisuudet
- * Kommunikaatiokatkokset
 - Asiakas tiivistä mukana prosessissa
 - * Tunnistetaan virheet ja korjataan aiemmin
 - * Päästään nopeammin asiakkaan kanssa samalle viivalle ja ymmärretään paremmin (noise)
 - * Enemmän materiaalia, jota käsitellä vaatimuksiksi

- * Tahalliset poikkeamat vaatimuksista
 - Agile Manifesto (ei tehdä)
 - * IDEOLOGIA - epäammattimaista
 - * Asiakkaan mukana olo poistaa vaatimusten veikkaamisen tarpeen
- * Ohjelmiston logiikan suunnitteluvirheet ja koodausvirheet
 - Testausta tehdään kokoajan samanaikaisesti ohjelman kehityksen lomassa.
 - Virheet tunnistetaan ja korjataan nopeasti (testaus ja asiakkaan mukana olo).
- * Dokumentointi- ja koodausohjeiden noudattamatta jättäminen
 - Ohjeita on hyvin vähän ja ne ovat selkeitä ja järkeviä (luonnollisesti seurattavia)
 - CONVENTION OVER CONFIGURATION
 - Annetaan tilaa luovuudelle - korostetaan yksilöiden hyviä puolia (ei määritellä liian tarkasti)
- * Testausprosessissa oikomisot
 - Ketterässä prosessissa kokoajan mukana oleva testaus
 - EI saa oikoa, jos löydetään virhe, korjataan. Jos ei ehdi/onnistu - dokumentoidaan.
 - Jokainen sprint loppuu toimivaan kokonaisuuteen (koodia tai ei).
- * Toimintatapavirheet
 - Asiakkaan kanssa tiivis kommunikaatio
 - Asiakas mukana versiokehityksessä, jonka yhteydessä huomataan ajoissa väärin luodut käyttötapaukset
- * Dokumentointivirheet
 - Keskitytään laadukkaaseen ohjelmakoodiin ja testeihin, jota minimalistiset dokumentit tukevat.
 - Tehdään vähän mutta laadukkaasti (LEAN - WASTE)

KÄYTÖN LAATUTEKIJÄT

- * Oikeellisuus
- * Luotettavuus
- * Tehokkuus
- * Yhtenäisyys
- * Käytettävyys
- * Joustavuus
- * Testattavuus
- * Siirrettävyys
- * Uudelleenkäytettävyys

CLEANROOM
KANBAN