

Matti Vuori

# Testaus ja ongelmanratkaisu – eli miten Leanin oppiva organisaatio pääsee eroon bugeista

Yksi nykyaikaisen laadunvarmistuksen piirteitä on asioiden uusi näkeminen vuosien rutiinin läpi. Lean ja riskiperusteinen ja tutkiva testaus ovat esimerkkejä pyrkimyksestä löytää pihvi ja pureutua siihen. Ongelmanratkaisun näkökulma on myös sellainen, jota kannattaa tarkastella. Bugien löytyminenhän on merkki kahdenlaisesta ongelmasta – tekninen ongelma tuotteen kelpaavuudessa ja toiminnan ongelma, jonka vuoksi bugi on syntynyt tai se löydettiin kenties liian myöhään. Oppivan organisaation periaatteilla voidaan bugien syntyyn vaikuttavia ongelmia poistaa kestävästä parannusta tuottavilla tavoilla.

## Sisällysluettelo

|  |   |
|--|---|
| Johdannoksi yksinkertainen virheiden käsittelyn malli .....      | 2 |
| Ongelman tunnistaminen .....                                     | 3 |
| Ongelmien jako kahtia – tekniikka ja organisaation toiminta..... | 4 |
| Ongelman selvitys.....   | 4 |
| Toiminta-vaihe .....   | 4 |
| Eroon virheistä yhden kerran jälkeen! .....                      | 5 |
| Väliyhteenveto: laajennettu kaavio .....                         | 6 |
| Oppiva organisaatio ei intoile dokumentoinnissa.....             | 8 |
| Ongelmanratkaisua parantavat toiminnan piirteet .....            | 8 |
| Kyse on yhteisestä asiasta.....                                  | 9 |
| Muistutuksia... ..   | 9 |

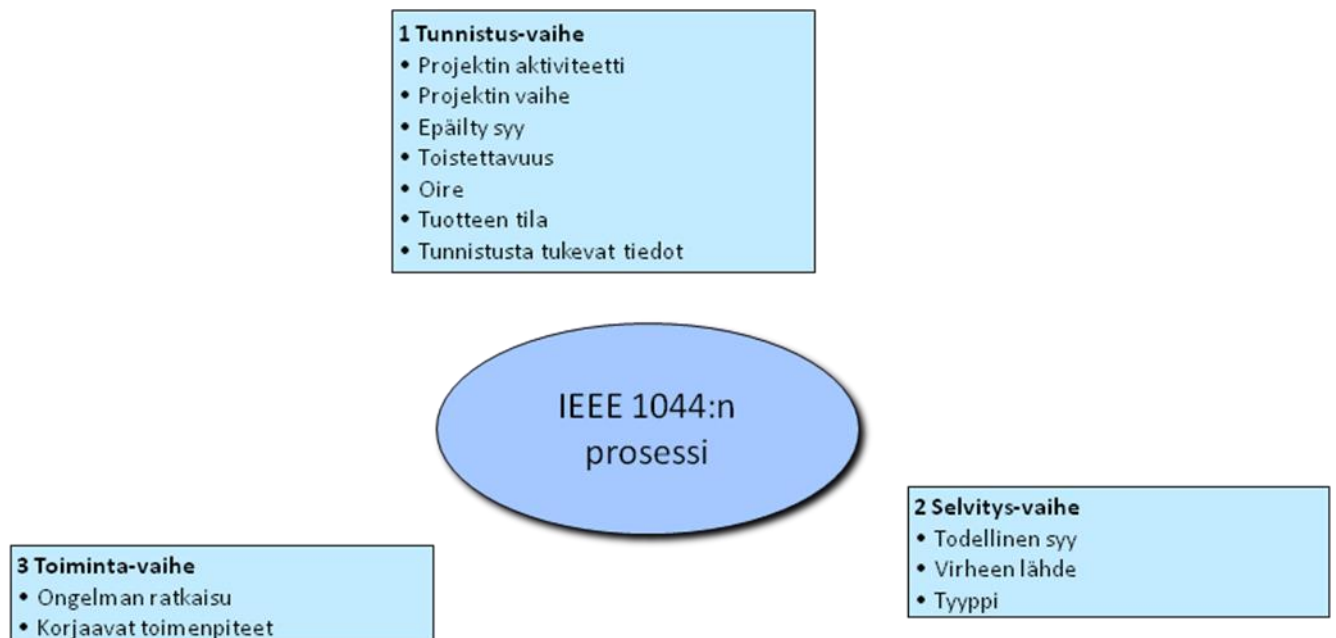
## Johdannoksi yksinkertainen virheiden käsittelyn malli

IEEE Std 1044-1993(R200) Standard Classification for Software Anomalies esittelee standardin ohjelmiston testauksessa löydettyjen poikkeavuuksien luokittelulle.

Standardi käyttää termiä "anomaly", anomalia, mikä viittaa kaikkiin sellaisiin asioihin, jotka poikkeavat odotetusta. Testauksen yhteydessä tämä tarkoittaa yleensä vikoja ja muita poikkeavuuksia, mutta standardin mukaan myös kehittämissuunnitelmia.

Standardin näkökulma on erittäin laaja verrattuna tyypilliseen vianhallintaprosessiin ja siksi se voi käytännössä useimmiten toimia vain omien luokittelujärjestelmien suunnittelun tarkistuslistana, varsinkin sellaisissa uusissa sovellusalueissa, joissa ei ole vielä syntynyt alan käytäntöjä luokitusjärjestelmille.

Standardi 1044-1993(R200) Standard Classification for Software Anomalies on kiinnostavampi kuin miltä näyttääkään, sillä pitkien ongelmia koskevien luokittelujen ohella se tarjoaa myös kivan generien prosessin niiden käsittelyyn



IEEE 1044:n ongelmien käsittelyprosessi ja kussakin vaiheessa käsiteltävät ja luokiteltavat asiat.

Prosessissa on kolme loogista vaihetta: ongelmat pitää tunnistaa, ne pitää tutkia huolella ja sitten tehdä niille jotain.

Yksi tärkeä asia ongelmia käsitellessä on niiden vaikutusten arviointi. Sitä tapahtuu prosessin jokaisessa vaiheessa. Siinä pohditaan tällaisia asioita:

- Vakavuus
- Prioriteetti
- Korjauksen arvo asiakkaalle
- Liiketoimintaoperaation turvallisuus
- Projektin aikataulu
- Projektin kustannukset
- Projektin riskitaso
- Projektin laatu/luotettavuus
- Yhteiskunnalliset vaikutukset

## Ongelman tunnistaminen

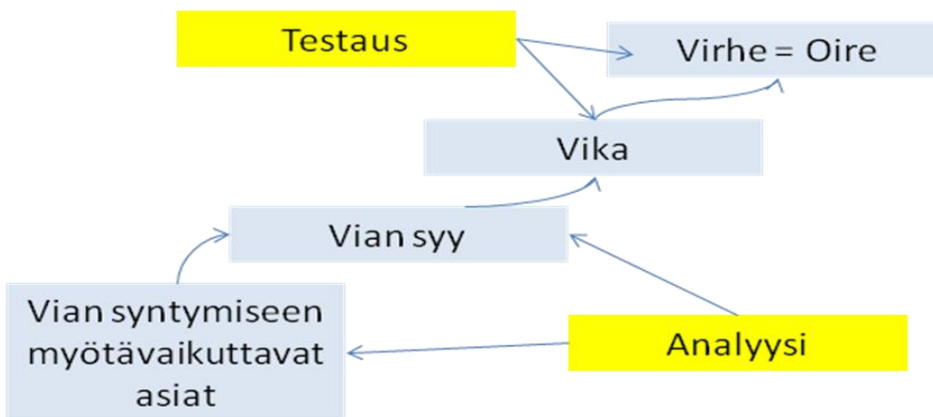
|   |   |
|---|---|
| <p>Tuotteen ongelmia:</p> <ul style="list-style-type: none"> <li>• Asiakkaamme toiminta kärsii</li> <li>• Tuotetta ei voida luovuttaa asiakkaalle</li> <li>• Asiakkaamme tarvitsema tärkeä toiminto ei toimi</li> <li>• Tuotteessa on vika</li> </ul> | <p>Toiminnan ongelmia</p> <ul style="list-style-type: none"> <li>• Vika on päässyt useiden testausvaiheiden läpi, miksi?</li> <li>• Miksi määrittelyihin on päätynyt aivan väärin asiakkaan tarvetta tulkitseva tieto?</li> <li>• Miksei testaajia koulutettu uuden testausvälineeseen logien tutkimiseen?</li> </ul> |
|---|---|

Testauksen tärkeä rooli tässä maailmassa on tunnistaa ongelmia – kun bugit ja puutteet on löydetty, niille voidaan tehdä jotain. Ongelmanratkaisun näkökulmasta tunnistusvaiheelle on useita tavoitteita:

- Varmistuminen siitä, että on ongelma. Tämän vuoksi testiä toistetaan ja pyritään saamaan tietoa ongelman toistumisesta ... satunnaisia häiriöitä ei kannata paikkaa ainakaan kehittämisen alkumetreillä.
- Selvittäminen, miten vakava ongelma on kyseessä. Testauksessa tallennetaan tieto siitä, miten pahasti järjestelmä toimii virheellisesti. Kun tämä yhdistetään testattavan asian tärkeyteen – palkanmaksutoiminto on tärkeämpi kuin matopelin käynnistys – tiedetään, missä mennään.

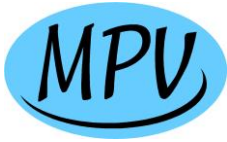
Jos testausperiaatteena on riskiperusteisuus, jo testauksen lähtökohtana on tieto asioiden vakavuudesta, mutta jos testauksen lähtökohdaksi ei mietitä tuotteen toimintojen tärkeyttä, joudutaan palaamaan kysymykseen: mihin kaikkeen ongelma vaikuttaa?

Mutta tunnistamisen on tärkeää olla kevyt prosessi, jolla vain kirjataan, mitä on havaittu ja luodaan sopiva ”esiymmärrys” ja orientaatio asian jatkokäsittelyä varten. IEEE:n mallissa on kivat luokittelut ongelman oireelle – miten se ilmenee. Olennaista on muistaa, että ensin tunnistetaan vain oireita ja myöhemmin tehdään diagnoosi, jotta **voidaan poistaa syyt**.



Ongelmista syihin testauksen ja analyysin avulla.

Yksi tärkeä asia on myös selvittää, kenelle asia ohjataan jatkoa varten.



## Ongelmien jako kahtia – tekniikka ja organisaation toiminta

Jo tunnistusvaiheessa tapahtuu havaintojen jakaminen kahteen eri ongelmaan. Meillä on tekninen ongelma, bugi, jolle pitää tehdä jotain nopeasti. Mutta sitten meillä on myös ihmisten toiminnan ongelma. Riippuen testausvaiheesta, ei bugeja pitäisi tulla kovin suurta määrää vastaan, ja vaikka tulisikin, voidaan ajatella perfektionistisesti, että **tällaisia bugeja ei mielellään saisi tulla enää toista**. Miksi tulisikaan, kun nyt selvitetään, mistä se johtuu... Tämä olisi leanin yksi kantava ajatus. Palataan siihen jonkin ajan päästä.

## Ongelman selvitys

Selvitys-vaiheessa selvitetään tärkeänä asiana, mistä ongelma johtuu. Asia voidaan ensimmäisenä jäljittää tekniikkaan – jossain on jokin toteutettu eri tavalla kuin pitäisi. Testaus tunnistaa vain käyttäytymisen, mutta jonkun pitää jäljittää taustalla oleva tekninen vika tai virhe. Tähän vaiheeseen tai seuraavaan voi kuulua ohjelmallinen tai manuaalinen analyysi virheellisen kohteen riippuvuuksista kokonaisuuteen, jonka perusteella tiedetään, miten herkän asian kanssa ollaan tekemisissä ja mihin kaikkeen korjaaminenkin voi vaikuttaa.

**Oleellista on jäsentää kokonaisuuksia eri abstraktiotasoilla, jotta korjauksetkin tehdään oikealla mentaliteetilla:**

- Koodi.
- Arkkitehtuuri.
- Järjestelmä.
- Järjestelmien järjestelmä.
- Rakenne ja toiminta

Testauksella voi olla tärkeä rooli, kun tässä vaiheessa tutkitaan tarkemmin ohjelmiston käyttäytymistä ja jäljitetään ongelman taustalla olevia tekijöitä kokonaisjärjestelmästä.

Kun syyt tiedetään, voidaan tehdä korjauksia...

Mutta pitää myös selvittää, **mistä syistä virhe syntyi** – tehtiinkö vaatimusmäärittelyssä tulkintavirhe vai saivatko suunnittelijat jostain väärää tietoa.

Varsinkin tämän vaiheen tekniikkaan kohdistuva osuus perustuu ideaalisesti jo aiemmin, projektin määrittelyvaiheessa tehtäviin analyysiin. Kun esimerkiksi vika- ja vaikutusanalyysillä (FMEA) tai vikapuuanalyysillä on analysoitu kohteen käyttäytymistä, voidaan löydetty virheen vaikutukset tai syyt tunnistaa nopeasti – ja samalla täydentää ja validoida malleja. Testaajat ovat olennaisia osallistujia analyysisessioissa.

Riippuen toiminnan systemaattisuudesta ja dokumentointimielisyydestä, virhettä voidaan luokitella tässäkin vaiheessa erilaisilla tavoilla. ”Analyysi-positivistinen” ajattelumalli on, että jos esimerkiksi kirjanpitomme kertoo, että algoritmivirheitä esiintyy paljon, sitten tehdään kampanja, jolla asiaan kiinnitetään erityistä huomiota.

Tähän vaiheeseen palataan myöhemmin uudestaan.

## Toiminta-vaihe

Toiminta-vaiheessa asialle tehdään jotain. Aiemmat vaiheet ovat usein yksilötyötä, mutta toimenpiteistä päättää usein jonkinlainen raati. Pyrkimyksenä on tehdä realistisia päätöksiä ja pitää erilaiset vaihtoehdot avoinna alkaen virheen nopeasta korjauksesta korjauksen siirtoon seuraavaan isompaan päivitykseen. Viallinen komponentti ja sitä vastaava toiminto voidaan usein pudottaa pois tuotteesta. Joskus ongelman ratkaisu on sen vaihtaminen pienempään...

IEEE:n luokitusjärjestelmässä on mukana myös valtionhallintoon ja koulutusjärjestelmään kohdistuvia toimenpiteitä! Tämä on luontevaa kansallisille missioille ja suuryrityksille.

Järkevyys-pyrkimys onnistuu, kun toiminnan kypsyy on riittävällä tasolla ja tuotteenhallinta toimii siten, että kyetään käsittelemään toimintojen ja vaatimusten toteutusjärjestystä ja ajankohtaa rationaalisesti.

Tässä jäsenyksessä toiminta-vaiheeseen kuuluu myös korjauksen validointi ja verifiointi, jossa testaajat palaavat taas asiaan:

- Virheen paljastaneiden testien suorittaminen uudelleen.
- Korjauksen validointi – onko uusi ratkaisu hyvä?
- Korjauksen verifiointi – onko uusi ratkaisu speksien mukainen?
- Regressiotestaus.

Regressiotestauksella on tärkeä ongelman ratkaisun validoinnin rooli, sillä hyvä ratkaisu ei tuota uusia ongelmia.

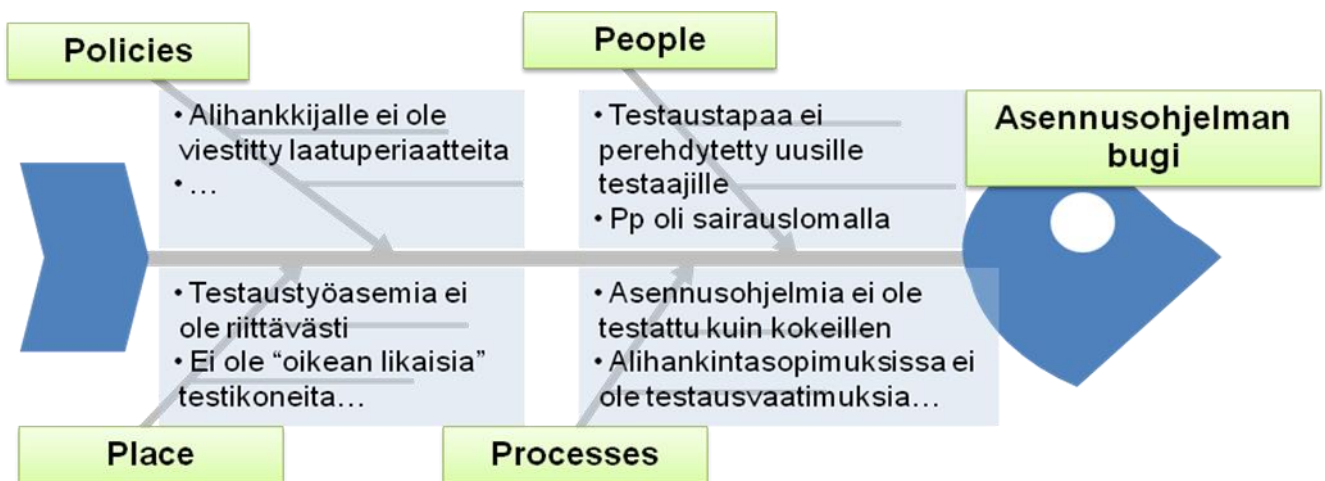
## Eroon virheistä yhden kerran jälkeen!

Monessa suomalaisessa yrityksessä tutkitaan tarkasti virheet, jotka havaitaan erilaisissa prosessin loppuvaiheen hyväksymistesteissä, koska silloin virheiden korjaus tuottaa paljon kustannuksia.

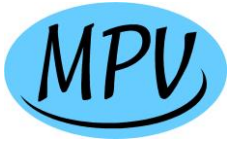
Leanissa pyritään samaan, mutta suhde tähän on kuitenkin hieman erilainen. Ennenkaikkea virheitä katsellaan ”sillä silmällä” jo hieman alemmilla testaustasoilla (esim. ”järjestelmätestaus”).

Leanissa ei vihata virheitä, sillä se tuottaa niiden peittelyä. Sensijaan **virheitä rakastetaan**, koska sellaisen löytyminen antaa uuden mahdollisuuden oppia ja ennenkaikkea on löydetty jotain sellaista, joka olisi muuten saattanut päätyä asiakkaalle asti.

Leanissa on pätevä, stabiili, sitoutunut ja oppiva organisaatio. Bugin taustalla olevat asiat **pohditaan yhteistyössä, sopivassa tiimissä**. Olennasta on a) tiimin luova synergia ja b) meneminen mekanistisen tarkastelun yli kulttuurisiin tekijöihin. Toyotan alkuperäisessä kulttuurissa työkaluna olisi kalanruotokaavio, eli vaikuttavien tekijöiden jäsenyys, jossa on neljä P:tä: People, Processes, Policies ja Place.



Kalanruotokaavio on yksi tärkeimmistä laatutyökaluista.



Tällaiset jäsenyydet pitää aina lokalisoida kulttuuriin ja aikakauteen, ottaen huomioon se, mitä kaikkea on vuosien myötä opittu organisaation toiminnasta. Käytännöllisiä asioita, joista etsittäisiin virheen syntymiseen vaikuttavia asioita olisivat esim.

- Tavat toimia tiimeissä
- Ohjeet
- Osaaminen
- Periaatteet ja politiikat (kaikilla organisaatiotasoilla)
- Yhteistyö
- Viestintä
- Välineet
- Tekniikka
- Laatuperiaatteet prosessin vaiheissa
- Jne...

**Tiimin luovaan ongelmanratkaisuun** kuuluu, että ratkaisuja katsellaan kalanruotokaavion avulla vapaasti ideoiden ja keskustellen ja tarkistuslistat ovat toisen tarkastelukierroksen väline, jolla varmistetaan, että relevantteja asioita ei ole jäänyt huomaamatta.

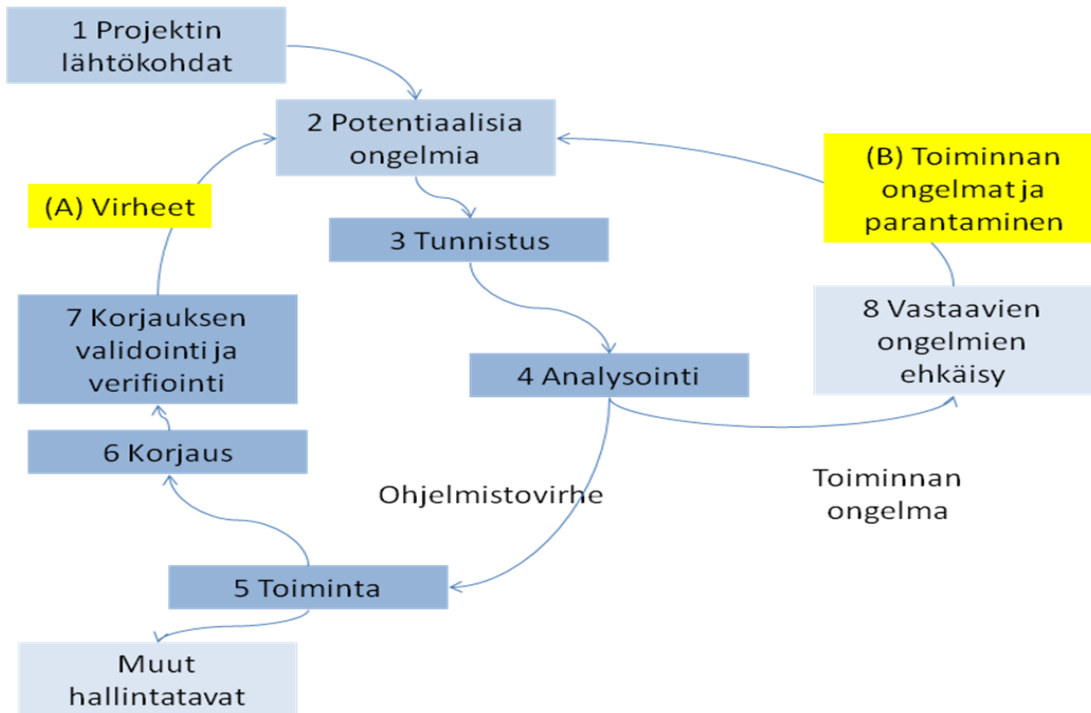
Näihin liittyviä parannuskeinoja konkretisoitaisiin tekemällä prosessimuutoksia, uudistamalla aiempien vaiheiden testausta, nostamalla katselmoiteihin uusia teemoja, parantamalla dokumentointia, preppaamalla ihmisten osaamista, jne...

Tätä toimintaa pitäisivät kasassa kirkaat periaatteet, "laadun talon pylvää":

- **Bugeja rakastetaan ja samalla vihataan. Mutta pää on aina kylmänä**, jotta osataan tehdä oikeita päätöksiä.
- **Bugin löytyminen on iloinen asia!** Jos bugien löytyminen on ikävää, niitä ei kohta enää löydetä...
- **Korkein teoreettinen ymmärtäminen** – testajaat ja kehittäjät tietävät, mitä ovat tekemässä. Heillä ei ole generisiä sertifikaatteja, vaan heille annetaan pätevää koulutusta..
- **Yhteinen oppiminen ja jatkuva parantaminen.** Bugit analysoidaan yhdessä, tiimissä
- **Vakioidut toimintatavat**, joita parannetaan heti, kun nähdään parantamisen mahdollisuus.
- **Luottamus** ihmisiin. Pitkät työsuhteet.
- Organisaation **sosiotekninen ymmärrys**: bugeissa ei ole kyse vain tekniikasta, vaan pääosin siitä, miten ihmiset toimivat yhdessä.
- Johto ja työntekijät jakavat saman **laatufilosofian**.
- **Realistinen täydellisyys**: asiakaslähtöinen, riskiperusteinen testaus.
- **Harkinnan ja nopeuden kulttuuri**: syvällinen bugien analyysi, harkinta prosessin kehittämisessä, mutta nopeus ongelmanratkaisussa ja bugin teknisessä korjauksessa.

## Väilyhteenvedo: laajennettu kaavio

Seuraavassa kaaviossa näkyvät nyt erityyppisten ongelmien käsittelyn haarat... ja siten se jatkuvan parantamisen "sarjatyön" prosessi, jolla virheistä päästään vähitellen kokonaan eroon (sis teoriassa...).



Oppivan organisaation jatkuvan parantamisen malli, jolla ongelmista pääsee vähitellen eroon.

Seuraavassa taulukossa on kaavion eri vaiheiden testaukseen liittyviä keskeisiä toimia.

| Vaihe                                  | Testauksen tehtävät   |
|--|---|
| (A) Virheet                            |   |
| 1 Projektin lähtökohdat                | Lähtökohdan validointi ja "kunnontarkastus" testaamalla<br>Testauksen suunnittelu<br>Testauksen ammattilaisten panos mm. luotettavuusteknisiin analyyseihin |
| 2 Potentiaalisia ongelmia              | Tilannetieto kokoamalla testauksen tuottamaa tietoa   |
| 3 Tunnistus                            | Ongelmien tunnistus testaamalla   |
| 4 Analysointi                          | Ongelman tutkiminen ja jäljittäminen diagnosoivilla testeillä<br>Testauksen ammattilaisten panos vian syntyyn vaikuttavien asioiden analyyseihin            |
| 5 Toiminta                             | Osallistuminen ongelman korjausstrategian valintaan   |
| 6 Korjaus                              | Korjaustyöhön integroitu ja matalan tason testaus   |
| 7 Korjauksen validointi ja verifiointi | Korjauksen onnistumisen ja vaikutusten testaus  |

| Vaihe                                  | Testauksen tehtävät   |
|--|---|
| (B) Toiminnan ongelmat ja parantaminen |   |
| 8 Vastaavien ongelmien ehkäisy         | Testauksen parantaminen<br>Niiden prosessien ja toimintamallien parantaminen, joihin testaus antaa panoksensa |

## Oppiva organisaatio ei intoile dokumentoinnissa

Esillä ollut standardi on suunniteltu järeän luokan ”missioihin” ja systemaattiseen tilastoivaan laatujohtamiseen. Mutta samoja periaatteita sovelletaan Leanissa siten, että tärkein dokumentaatio on se, mitä kertyy ihmisten mieliin bugeja analysoitaessa.

Siinä, missä ei-lean laatu toiminta kerää tietoja lomakkeille ja tietokantaan, Leanissa piirretään kalanruotokaavoita ja keskustellaan tiimissä toiminnan muuttamisesta.

## Ongelmanratkaisua parantavat toiminnan piirteet

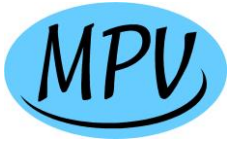
Millainen on hyvä ongelmanratkaisuprosessi?

- Soveltuu käsillä olevaan organisaatioon ja ihmisille
- Poistaa ongelman syyt, eikä vain oireita
- Tuottaa helposti hyödynnettäviä tuloksia
- Tuottaa siksi kestäviä parannuksia
- Ei aiheuta uusia ongelmia
- Osallistaa asianosaiset – hyödyntää kaikkien osaamista ja vähentää ratkaisun muutosvastarintaa
- Positiivinen ja energisoiva
- On piirteiltään tarkoituksenmukainen ja siksi riippuu toiminnan luonteesta ja organisaatiokäsityksestä
- Abstrahoi arkea sopivasti, jotta asioita voidaan käsitteellistää ja käsitellä
- On kustannustehokas
- Jne...

Bugien tunnistamisesta alkava prosessi on yleensä luonteeltaan aivan tervejärkinen ongelmanratkaisuprosessi. Eräät piirteet edistävät sitä:

- **Riskiperusteinen testaus** nostaa esille ohjelmiston piirteet, joihin liittyvät havainnot ovat ongelmia, ja siten nostaa orientaatiota.
- **Oppiva organisaatio tarvitsee tiimejä, joissa on erilaisia ihmisiä.** Ketterän kehityksen tiimit voivat olla tällaisia, mutta ihan perinteiset kehitystiimit tukevat samaa asiaa ja ovat sopivasti irti toiminnan mikrotasosta.
- **Ongelmanratkaisu tarvitsee kevyitä välineitä.** Perinteiset laatu työn välineet, kuten kalanruotokaavio, voivat olla arvokkaita. Mutta yhtä tärkeitä ovat järeämmät välineet, kuten vaikka Vika- ja vaikutusanalyysi (FMEA).
- **Luovat analyysimenetelmät estävät sokeiden pisteiden syntymisen.**
- Koska toiminnan muuttaminen edellyttää harkintaa, sen ei tarvitse olla sovitettu päivittäiseen toimintaan. Mutta bugien korjauksen pitää olla yksi keskeisistä työn piirteistä.
- Tarvitaan tervettä laatuajattelua, jossa **suhde virheisiin on kypsä** – niitä ei saa vihata, koska silloin niitä lakaistaan maton alle. Ja vihaisen ihmisen ajattelu on huonoa.





- **Organisaatiokäsityksen** pitää myös olla kypsä. Leanin oppiva organisaatio ja sosiotekninen käsitys toiminnasta ovat tässä suhteessa oivallisia. Jos organisaatio nähdään vain mekanistisena tuotantokoneena ja ihmiset resursseina, ongelmanratkaisu ei tuota hyviä tuloksia.
- Organisaation pitää olla joustava ja osata käyttää **erilaisia toimintatylejä**. Joskus pitää olla ripeä ja joskus harkitseva. Usein prosessit ovat yhden koon prosesseja eikä oteta huomioon, että erilaiset ongelmat vaatisivat erilaista prosessointia. Kykeneminen siihen toisi selvää etua.

## Kyse on yhteisestä asiasta

Kun bugin löytymisestä – tai oikeastaan löytämisen suunnittelusta – alkavaa prosessia tarkastelee, huomataan, että testaajilla on vetorooli sen kahdessa kriittisessä kohdassa: virheiden tunnistamisessa ja niiden poistumisen verifiointissa. Ja niiden välillä testaajien ja testauksen hyödyntäminen vaihtelee. Ongelman selvitysvaiheessa testaajilla on tärkeä rooli tiimissä ja erittäin potentiaalinen rooli laatuasiantuntijoina, vetämässä "uusia vanhoja" ongelmien selvittämisen prosesseja. Tämä on tosin tällä hetkellä olosuhdesyistä "imaginääritason potentiaali", johon ei kaikkialla ole mahdollisuuksia.

Ongelmanratkaisuprosessi on näin testauksen sulkema ja läpäisemä ja se voi toimia eheästi vain, jos testaus on kunnossa.

## Muistutuksia...

Ongelmanratkaisunäkökulman keskeinen etu on siinä, että testauksella on abstraktimpia ja strategisempia merkityksiä kuin aina muistetaan. Koska kypsään toimintaan on aina kuulunut myös sopiva abstraktimpien ja strategisempien käsitteiden ja ajattelumallien käyttö, miksipä emme soveltaisi sellaista. Ja jokainen syy, jolla saadaan nostettua esille oppivan organisaation ideaaleja, on sen arvoinen!