

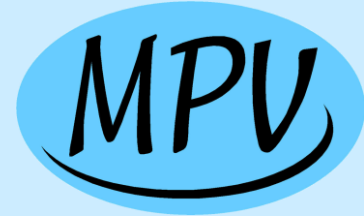
# Menestyksenkäs hyväksymistestaus



Näkökulmia järjestelmän hyväksymistestauksen monimuotoisiin menestystekijöihin järjestelmän tilaajan näkökulmasta.

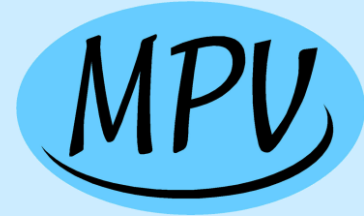


Matti Vuori, [www.mattivuori.net](http://www.mattivuori.net)



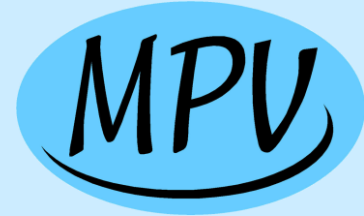
# Sisällysluettelo 1/3

|  |           |
|--|-----------|
| <u>Hyväksymisen tavoitteet</u>   | <u>5</u>  |
| <u>Kiire bugien löytämiselle</u>   | <u>6</u>  |
| <u>Suhde muihin testeihin</u>  | <u>7</u>  |
| <u>Tekijät</u>   | <u>8</u>  |
| <u>Ympäristö</u>   | <u>9</u>  |
| <u>Perusmenettelyt</u>   | <u>10</u> |
| <u>Joskus asiat sujuvat</u>  | <u>12</u> |
| <u>Hyväksymistestauksen ideaalit</u>                                       | <u>13</u> |
| <u>Hyväksymistestauksen karu todellisuus</u>                               | <u>14</u> |
| <u>Hyväksymistestauksessa testattavat asiat</u>                            | <u>16</u> |
| <u>Erityyppisten ja eritasoisten hyväksymistestien erottaminen</u>         | <u>17</u> |
| <u>Osapuolet kokevat hyväksymistestauksen joskus eri tavoilla</u>          | <u>19</u> |
| <u>Ohjelmiston hankkijan riippumattomuus, riskienhallinta ja oppiminen</u> | <u>22</u> |
| <u>”Ohjelmistohankintojen lukutaito”</u>                                   | <u>23</u> |



# Sisällysluettelo 2/3

|   |    |
|---|----|
| <a href="#">Lyhyt lähtökohtainen sanasto</a>                        | 24 |
| <a href="#">Toimittajan eettiset periaatteet</a>                    | 27 |
| <a href="#">Kokonaislaadun varmistaminen</a>                        | 28 |
| <a href="#">Mitä vasten testataan?</a>                              | 29 |
| <a href="#">Hyväksymistestaus ja riskienhallinta</a>                | 30 |
| <a href="#">Tietojärjestelmähankinnan riskikartta</a>               | 31 |
| <a href="#">Hyväksymistestauksen projektointi</a>                   | 32 |
| <a href="#">Projektissa tarvitaan</a>                               | 33 |
| <a href="#">Jos alihankintana</a>                                   | 35 |
| <a href="#">Valmiuksien ja suunnitelman katselmointi</a>            | 36 |
| <a href="#">Järjestelmätoimittajan antama apu</a>                   | 37 |
| <a href="#">Projektivalmennus tarpeen</a>                           | 38 |
| <a href="#">Ns. liiketoimintatestaajien käyttäminen testaukseen</a> | 39 |
| <a href="#">Testattavat asiat ja järjestys</a>                      | 42 |



# Sisällysluettelo 3/3

|   |    |
|---|----|
| <a href="#">Testidata</a>   | 45 |
| <a href="#">Testiautomaatio?</a>  | 46 |
| <a href="#">Inkrementaalinen hyväksyntä</a>   | 47 |
| <a href="#">Vikatiedon keruu ja käsittely</a>   | 49 |
| <a href="#">Yleinen toimintamalli tarpeen</a>   | 50 |
| <a href="#">Ketterän kehittämisen mahdollisuudet ja potentiaaliset ongelmat hyväksynnälle</a> | 51 |
| <a href="#">Scrumin ”hyväksymistestauksen” kritiikki</a>                                      | 52 |
| <a href="#">10 keskeistä menestystekijää tilaajalle</a>                                       | 56 |
| <a href="#">10 keskeistä menestystekijää järjestelmätoimittajalle</a>                         | 57 |
| <a href="#">Yhteenveto</a>  | 59 |

# Hyväksymisen tavoitteet

- Ohjelmiston hankkija todistaa itselleen, että
  - Ohjelmisto on käyttöön kelpaava ja täyttää tarpeet ja vaatimukset
  - Ohjelmistotoimittajan työ voidaan hyväksyä ja laskut maksaa
- Ohjelmiston hankkija antaa toimittajalle hyväksymisen heidän työlleen

# Kiire bugien löytämiselle

- Ideana on takuuaikana löytää kaikki ne bugit, jotka muuten kenties löytyisivät hiljalleen vuosien varrella
  - Näin päästään tuottavaan työhön heti
  - Kun bugit löydetään takuuaikana, toimittaja maksaa niiden korjaukseen!
- Näkökulman pitää siis olla tulevaisuudessa: ei ajatella pärjäämistä järjestelmän kanssa vain ensi viikolla

# Suhde muihin testeihin

- Hyväksymistestaus on selvää jatkoa järjestelmätestaukselle.
- Se on esiaskel pilottitestaukselle asiakasorganisaatiossa.
- Jossain ympäristöissä hyväksymistestauksen kevyempi muoto on "vastaanottotestaus" tai tarkastus.
- Käytettävyydestestaus on myös asiakasnäkökulmaltaan lähellä hyväksymistestausta.
- Asiakkaan käyttöönottoprosessissa seuraava askel on usein pilottikäyttö pienessä mittakaavassa.

# Tekijät

- Asiakas tekee testauksen tai teettää sen kolmannella osapuolella.
- Kolmas osapuoli voi olla testaustalo, joka ei osallistu prosessiin millään muulla tavalla.
- Tietojärjestelmähankkeissa testaukseen osallistuu asiakkaan IT-osasto sekä loppukäyttäjät.
- Asiakaslähtöisessä kehittämisessä ohjelmistotoimittaja ei yleensä voi itse tehdä hyväksymistestausta.



# Ympäristö

- Tilaajan tuotantoympäristöä mahdollisimman hyvin vastaava ympäristö
  - Kopio
  - Viimeiset testit todellisessa tuotantoympäristössä
  - Työasemat loppukäyttäjien työasemia
- Mielellään muutama eri taso, joissa haetaan luottamusta ja lisätään yhteyksiä muihin järjestelmiin

# Perusmenettelyt 1/2

- Menettelyihin ei ole vakiintunutta tapaa, vaan ne sovitaan aina asiakkaan kanssa tapauskohtaisesti.
- Testaus perustuu kaikkiin ohjelmiston vaatimuksiin.
- Perusmenettelyt samat kuin järjestelmätestauksessa, mutta
  - yleensä keskittyen tärkeimpiin toiminnallisuuksiin,
  - yhteensopivuuteen ja
  - järjestelmän käytettävyyteen – eli niihin asioihin, jotka tyypillisessä tietojärjestelmähankkeessa jäävät ohjelmistotoimittajan testauksessa vähemmälle huomiolle.

## Perusmenettelyt 2/2

- Testausta edeltää ohjelmiston katselmointi ja analyysi, jossa sitä verrataan toiminnallisiin ja teknisiin määrittelyihin ja sopimukseen.
- Testaus voi olla kertaluonteinen hyväksymismenettely tai se voi sisältää puutteiden korjauksen (ohjelmistotoimittaja tekee) ja uudelleentestauksen.

# Joskus asiat sujuvat

- Yrityksillä, joilla on vahvat tuotantoon siirron testausrutiinit, on tekninen laatu helpompi varmistaa
  - Monitasoiset testausjärjestelmät.
  - Mm. pankki- ja vakuutusala.
- Pitkä historia lisää hyväksymistestauksen ymmärrystä ja osaamista
- Mitä isompi yritys, sitä todennäköisemmin siihen on päätynyt testaus-, laatu- ja riskienhallintaosaamista

# Hyväksymistestauksen ideaalit

- Järjestelmän vaatimukset selvitetään huolellisesti tilaajan ja toimittajan yhteistyönä.
- Vaatimukseen liitetään tieto niiden hyväksymiskriteereistä.
- Myöhemmin todetaan objektiivisella tavalla, että kriteerit täyttyvät.
- Tämä voidaan tehdä yksinkertaisella tavalla, koska sitä edeltää hyvin huolellinen ohjelmistotoimittajan tekemä testaus kaikilla testausasteilla.
- Kaikki ovat tyytyväisiä.
- Tämä ideaalimalli on vesiputous-mindsetin mukainen.

## Hyväksymistestauksen karu todellisuus 1/2

- Ohjelmistoja ei koskaan ymmärretä täysin niiden määrittelyvaiheessa.
- Hyväksymiskriteerejä ei vaatimuksia tarkemmin voida määrittää, koska ei ole riittävästi tietoa ja kokonaisuus on liian vaikea jäsentää.
- Vaatimukset muuttuvat matkan varrella.
  - Kukaan ei oikein tiedä, mitä on sovittu ja mitä siis pitäisi hyväksyä
  - Ja muuttuuko joko toiminto vielä
- Ohjelmistotoimittajaan ei voi luottaa, eikä pidä luottaa.
  - Worst case on, että ohjelmisto tai sen räätälöinti on hyvin heikosti testattu.

## Hyväksymistestauksen karu todellisuus 2/2

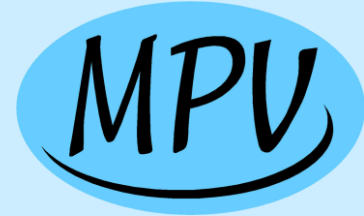
- Tilaajan väellä on ylikire.
- Osallistumisresurssit on aliarvioitu.
- Hyväksymistestaus on osa käyttöönottoprosessia, johon ei saada toimittajan tukea, koska se ei ole osa sopimusta.
- Hyväksymistestausta ei ole valmisteltu ja se tehdään suunnittelemattomasti vasta, kun törmätään ongelmiin tuotannossa.
- **Sitä ei edes yritetä, koska kenelläkään ei ole ymmärrystä sellaisen tarpeesta tai osaamista sellaisen tekemiseen.**



## Hyväksymistestauksessa testattavat asiat

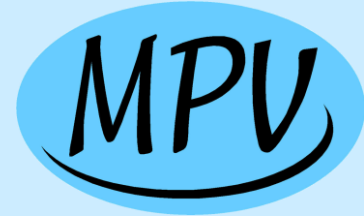
- Hyväksymistestaus perustuu mieluiten ohjelmistoprojektin suunnitteludokumentteihin tai niiden puuttuessa olemassa olevan ohjelmiston toteutukseen.
- Tärkeimmät dokumentit hyväksymisvaatimusten selvittämisessä ovat:
  - Tilaus tai sopimus – mitä on tilattu ja luvattu?
  - Ohjelmiston vaatimusmäärittely, toiminnallinen ja tekninen määrittely.
  - Järjestelmän toteutuksen / räätälöinnin / toimituksen projektisuunnitelma.



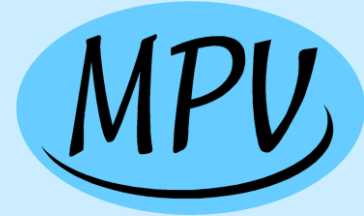


- Prosesseissa on useita ”hyväksymistestejä”, joiden on suuri vaara mennä sekaisin:
  - Ohjelmistokehityksen vaiheiden hyväksymistestit. Esimerkiksi testit, joilla todetaan, että versio on riittävän toimiva, jotta se voidaan toimittaa järjestelmätestaukseen.
  - Ohjelmistotoimittajan julkaisulle tekemä viimeisen vaiheen testi, jolla toimittaja todistaa itselleen, että julkaisu on toimituskelpoinen.
  - Ja keskeisimpänä tilaajan tekemä hyväksymistestaus ja sen yhteydessä liiketoiminnan ja käyttäjien hyväksyntä.

## Erityyppisten ja eritasoisten hyväksymistestien erottaminen 2/2

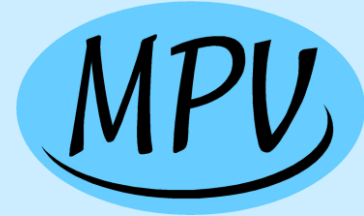


- Hyvissä kokonaisprosesseissa kaikilla erilaisilla testeillä on oma terminsä.
- (Oleellisempaa kuin testit on kuitenkin tilaajan hyväksyntä ja lausunto hyväksymisestä.)



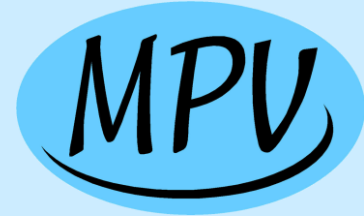
- **Kypsä asiakaslähtöinen ohjelmistotoimittaja** kokee, että tilaajan tekemä hyväksymistestaus on aivan kriittinen testausvaihe. Laatu paljastuu vasta asiakkaan käsissä.
- **Epäkypsä ohjelmistotoimittaja** kokee sen turhana ja haittana. Softahan on tehty juuri niin kuin pitääkin!
- **Epäkypsä tilaaja** luottaa liikaa toimittajaan eikä ymmärrä, että pitää testata itse kunnolla. Hyväksymistestaus, jos sitä tehdään, on kevyt rutiini.
- **Kypsä tilaaja** lähtee siitä, että hyväksymistestaukseen pitää suhtautua niin kuin järjestelmää ei olisi lainkaan testattu. Siksi pitää testata niin toimintoja kuin laatuominaisuuksia niin paljon kuin on rahaa. Muuten seuraa kallis korjauskierre.
- **Tekniset ihmiset** kokevat hyväksymistestauksen vain toiminnallisuuden testauksena.

## Osapuolet kokevat hyväksymistestauksen joskus eri tavoilla 2/3



- **Naivi tilaaja** kokee kiistanalaisena olevan vain ”tilatut toiminnot”, mutta valistunut tilaaja ymmärtää varmistettavana olevan koko toimituksen odotetun laadun, olipa asioita määritetty tai ei.
- **Laatuihmiset** kokevat sen koko uuden toimintajärjestelmän toimivuuden todentamisena. Hyvä johtaja arvelee, että käytettävyys ja henkilöstön hyväksyminen on joskus kaikkein tärkeintä.
- **Muutosjohtaja** näkee testauksessa organisaation muutospisteen, jossa käytännön testauksella on monia psykologisia merkityksiä
- **Riskienhallintaihminen** näkee siinä tulevien vuosikymmenten kannalta aivan kriittisen pisteen, jossa ratkaistaan kenties organisaation tulevaisuus.
- **Lakimies ja talousjohtaja** näkevät sen rutiininomaisena osana sopimusprosessia.

## Osapuolet kokevat hyväksymistestauksen joskus eri tavoilla 3/3



- **Toimittajan myyntipäällikkö** näkee sen tuoteviestinnän mahdollisuutena, markkinointina ja koulutuksena.
- **Epäkypsät johtajat** kokevat sen turhana IT-rutiinina.
- **Kypsät johtajat** tietävät, että siinä testataan tekniikkaa, toimintaa, prosesseja ja myös luottamusta kaikkien tahojen välillä.
- **Testauspäällikkö ja kypsä tilaajan projektipäällikkö** haluavat nähdä sen riittävän pätevänä testausprojektina.
- **Epäkypsä tilaajan projektipäällikkö** arvelee, että ”kokeilu” riittää.
- **Taloustietoinen projektipäällikkö** muistaa, että hyväksymistestauksella varmistetaan, että viat korjataan toimittajan laskuun, eikä niitä löydetä vasta takuuajan jälkeen.
- Jne...

## Ohjelmiston hankkijan riippumattomuus, riskienhallinta ja oppiminen

- Ohjelmistotoimittajalla ja tilaajalla on aina väistämättä oma näkökulmasta ohjelmistoon – ja siksi myös sen testaukseen.
  - Tilaajan on ”ajateltava itse” myös testaus.
- Riskienhallinnan näkökulmasta toimittajan tekemään testaukseen ei voida luottaa.
  - Molemmilla on aivan eri tavoitteet ja eri riskitaso
  - Riippuu sovelluksen liiketoimintakriittisyydestä, miten paljon tehdään sellaista testausta, joka olisi oikeastaan kuulunut toimittajan rooliin.
- Erilaiset testit täydentävät toisiaan.
- Jokainen ohjelmistoprojekti on oppimiskokemus. Asiakkaan oppimisen on näyttävä myös testauksessa.
  - Lopullinen hyväksymistestauksen sisältö paljastuu, kun siihen ryhdytään.

## ”Ohjelmistohankintojen lukutaito”

- Nykyään puhutaan mm. medialukutaidosta, mutta haastavampaa on ohjelmistojen tilaajien lukutaito toimittajan viestintään.
- Lukeminen rivien välistä, mikä oikeasti on asioiden tila, mitä pitää kyseenalaistaa ja mitä voi uskoa.
- Tilaajat ovat yleensä positivistisia, halutaan luottaa tekniikkaan ja toimittajaan. Silloin kielipeli peittää todellisuuden.
- Oikea viestinnän lukeminen ja oikeat kysymykset mahdollistavat oikean orientaation omatoimiseen toimitusten laadun varmistamiseen.
- **Nämä ovat asioita, jotka vaikuttavat koko ohjelmistokriisiin, eivätkä pelkästään toimitusten hyväksymiseen.**

## Lyhyt lähtökohtainen sanasto 1/3

- Meillä on kattava testausprosessi  
= Joissakin projekteissa saatetaan testatakin jotain. Onko näyttää muutakin kuin prosessikaavioita? Testiraportteja, vikatilastoja?
- Toimintamme vastaa ISO 9001:n vaatimuksia  
= Se ei kerro paljonkaan ohjelmistoprosessista; kysykääpä onko 90003 tuttu?
- Olemme erikoistuneet tällaisiin järjestelmiin  
= Niin oli erikoistunut myös se putkimies, jonka tekemä remontti oli täysin susi. Erikoistuminen ei takaa osaamisesta mitään.
- Projektin riskit ovat seuraavat  
= Ette ole koskaan miettineet kanssamme järjestelmäprojektin riskejä meidän kannaltamme
- Perusohjelmiston toimivuus on todistettu useilla asiakkailla  
= Tarjouksia on tehty kymmenen, jokainen kahdesta toteutuneesta toimituksesta on ollut yhtä vaikea... Saako asiakkailta kysyä?



## Lyhyt lähtökohtainen sanasto 2/3

- Toimintojen määrä on kilpailijatuotteita paljon laajempi  
= Ok, mutta ovatko ne oikeat toiminnallisuudet?
- Ohjelmisto on helposti räätälöitävissä  
= Pienikin muutos vaatii vuoden projektin ja tuottaa ison määrän muita ongelmia.
- Meillä on ketterä projektitoimitusmalli, jossa asiakas saa määritellä toiminnot joustavasti  
= Emme oikein osaa tehdä vaatimusmäärittelyä ja laitamme asiakkaan suunnittelijaksi
- Ohjelmisto on tehty pk-yritysten tarpeisiin  
= Sillä voi tehdä vain perusasiat, jotka eivät riitä meille
- Ohjelmiston viimeisessä versiossa on sen luotettavuuteen panostettu  
= Ohjelmistossa on ollut kauheasti vikoja ja niitä riittää edelleenkin

## Lyhyt lähtökohtainen sanasto 3/3

- Ratkaisu perustuu valmishjelmaan  
= Räättälöinnit ja integroinnit ovat poikkeuksellisen vaikeita; käyttöliittymää ei voida muuttaa, vaikka se on hyvin epästandardi ja huono
- Sovellamme uusimpia Web 2.0 –tekniikoita  
= Emme osaa tehdä tehokkaita operatiivisia sovelluksia
- Käytettävyyteen panostetaan tuotekehityksessä erityisesti  
= Graafikko laati softalle uuden väripaletin ja ikonin. Osoittakaa toimia käytettävyyden varmistamiseksi.
- Ohjelmisto perustuu XXX, YYY ja ZZZ –teknologioihin  
= Kaikki nämä ovat uusia teknologioita. Osaatteko ne ihan oikeasti?
- Olemme asiakaslähtöinen toimittaja  
= Kehittäjäme eivät ole koskaan tavanneet yhtään loppukäyttäjää
- Jne...

# Toimittajan eettiset periaatteet

- Auta tilaajaa ymmärtämään omat tarpeensa ja vaatimuksensa.
- Auta tilaajaan ymmärtämään omatoimisen hyväksymistestauksen tarve ja sen edellyttämät asiat.
- Viesti tilaajalle tehdystä ohjelmiston testauksesta ja sen aikana löydetyistä virheistä.
- Älä manipuloi tilaajaa tekemään testausta samoin kuin itse teitte, vaan auta häntä löytämään oikeat näkökulmat omaan testaamiseensa.
- Toimita asiakkaalle sellainen järjestelmä, jonka testattavuus on kunnossa – ottaen huomioon asiakkaan ympäristöt ja osaaminen.
- Anna tilaajalle täysi tekninen tuki ja tietotuki testaamiseen.
- Paljasta kaikki rajapinnat, jos tietoja tarvitaan testaamiseen. Tilaaja tekee päätöksen tästä tarpeesta.
- Hyväksy asiakkaan löytämät virheet ja opi niistä.

# Kokonaislaadun varmistaminen

- Hyväksymistestaus on valitettavan usein positiivisilla testeillä tehtyä toiminnallisuustestausta – jos sitäkään.
- Luotetaan liikaa toimittajan perusohjelmistoon.
- On varmistettava:
  - Kaikkien osapuolten vaatimusten täyttyminen (liiketoiminta, IT-hallinto / ylläpito, hallinto, erilaiset käyttäjäryhmät...).
  - Kaikkien laatuominaisuuksien riittävyys (virheetön toiminta, käytettävyys, suorituskyky, luotettavuus, tietoturvallisuus). Näitä ei käsitellä yleensä riittävän systemaattisesti.
  - Jne...
- Robustius käytännön arjessa.
  - Hyväksymistestauksessa tiivistetään aikaa ja kohdataan tarkoituksellisesti kaikki käytännölliset ongelmat, jotka muuten ilmenisivät harvakseltaan viiden vuoden aikana.
  - Hyväksymistestaukseen liittyvät myös operatiiviset tuotantoonsiirtokriteerit, joiden on hyvä olla osin geneerisiä kaikille järjestelmille

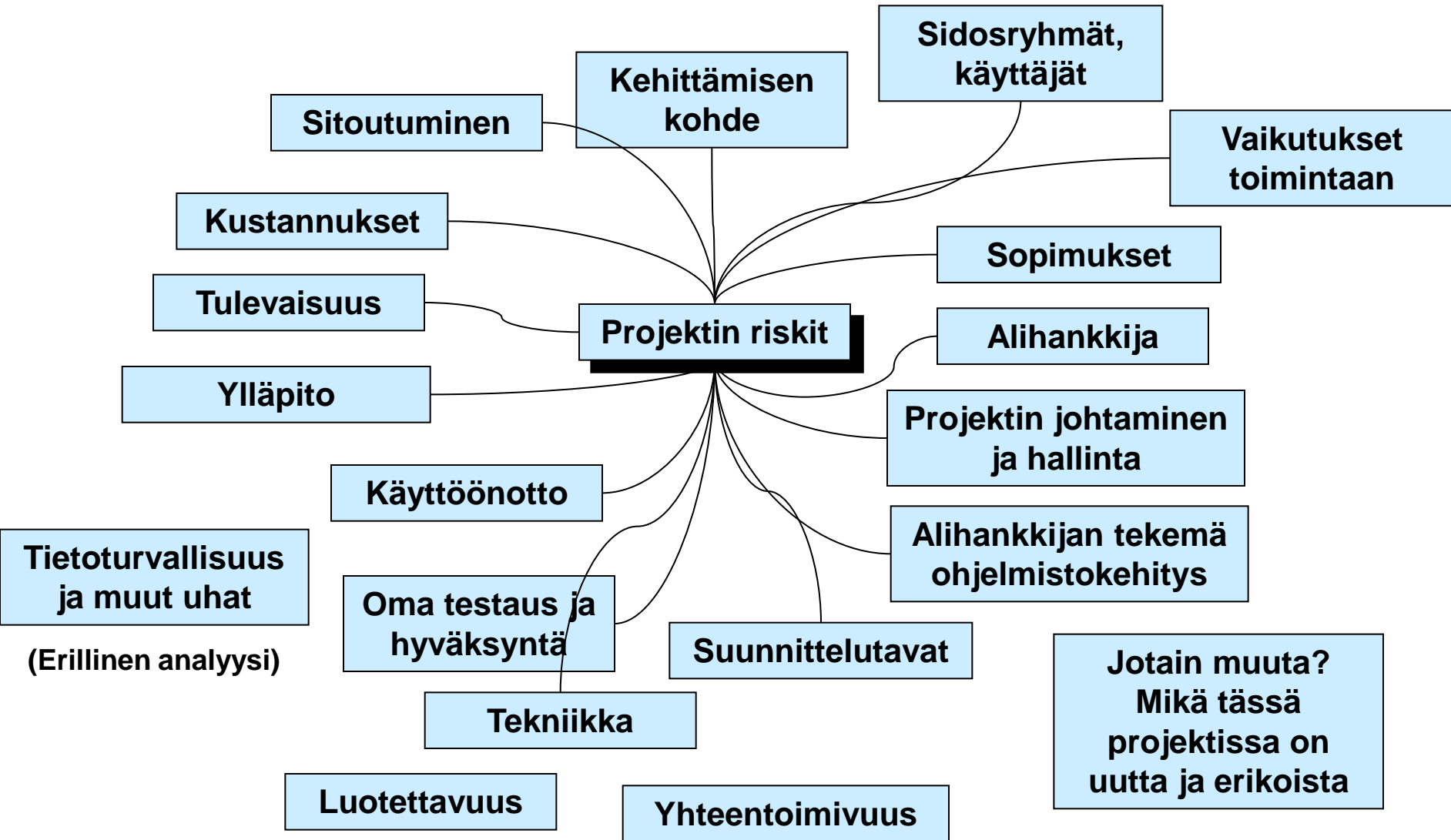
# Mitä vasten testataan?

- Formaalit vaatimukset
  - On vaikea tehdä hyvää testausta, jos vaatimukset on kuvattu yleisellä tasolla
  - Panostettava sen kuvaukseen, mitä vaaditaan
  - Ml. suorituskyky- ja käytettävyyksivaatimukset
- Odotukset – ja ajattelumalli
  - Mitä laadukkaalta ohjelmistolta voidaan edellyttää? Se on sitova vaatimus, ellei ole sovittu muuta
  - Sisäisen ymmärtämisen ja preppauksen ongelma
  - Perehdyttävä testaukseen osallistujat siihen, millaista mm. virheiden sietoa pitää voida edellyttää – ja mikä on siis testattava
  - Asenne testaukseen! Ei kohdella uutta järjestelmää silkkihansikkain, vaan vasaralla!

# Hyväksymistestaus ja riskienhallinta

- Järjestelmäprojekteihin liittyy aina suuri joukko erilaisia riskejä.
- Jokaisessa projektissa olisi tunnistettava riskit ja otettava ne huomioon
  - Järjestelmävaatimuksissa
  - Projektin koko elinkaareissa
  - Järjestelmän hyväksymisessä
- Hyväksymistestaus on yksi vaihe, jonka avulla tarkistetaan, että riskit ovat hallinnassa.
  - Esim. robustius järjestelmien yhteentoimivuudessa
  - Käyttäjien hyväksyntä – toiminnallisuus, käytettävyys...
- Laajamittaiselle projektoidulle hyväksymistestaukselle on hyvä tehdä oma riskianalyysi – testauksella on omanlaisensa projektiriskit.

# Tietojärjestelmähankinnan riskikartta



# Hyväksymistestauksen projektointi

- Järjestelmien omistajuus on useimmiten hajautettu.
- Ellei organisaatiossa ole perinteitä tuotantoon siirron rutiineissa ja tarvittavissa testeissä, hyväksymistestaus voi jäädä roolien ja vastuiden välimaastoon.
- Projektointi mahdollistaa hallitun professionaalisen testauksen ja eri organisaatioiden yhteispelin.
- Projekti on osa järjestelmän hankintaprosessia / käyttöönoton projektia



# Projektissa tarvitaan 1/2

- Projektipäällikkö
- Testausta suunnittelemaan ja siitä vastaamaan testauspäällikkö
- Testaajia
  - Pitääkö löytää jostain muutama hyvä ammattilainen?
- Testiympäristö tai useita
- Testaussuunnitelma
  - Miten testataan? Koska? Miten edistystä seurataan?
  - Testattavan version jäädyttäminen (toimittaja ei saa käydä sotkemassa sitä – pitää tietää, mitä testataan)
  - Suunnitelman katselmointi

## Projektissa tarvitaan 2/2

- Tapa kerätä vikatietoja
- Tapa viestiä vioista ohjelmistotoimittajalle
- Aikaa testaamiseen
- Ihmisten perehdyttämistä
- Järjestelmätoimittajan tukea

# Jos alihankintana

- Hyvän alihankkijan löytäminen
- Sopimus
- Alihankkijan tekemä suunnitelma
  - Aikataulu
  - Testaustavat
  - Raportointi
  - Tilaajan resurssit, dokumentit jne...
- Kuitenkin itsellä edelleen projektointi
  - Joku hoitaa asian, järjestee järjestelyt, seuraa tilannetta...



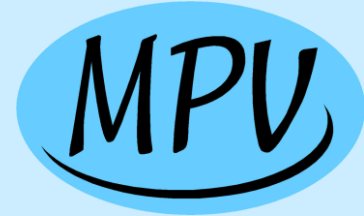
# Järjestelmätoimittajan antama apu

- Tieto järjestelmän rakenteesta ”konepellin alla”
  - Vaikka hyväksymistestaus on pääosin mustalaatikkotestausta, joskus on tarve mennä syvemmällekin
- Tieto siitä, miten järjestelmää on jo testattu
  - Testataan muutakin...
- Apu asennuksessa ja konfiguroinnissa testiympäristöön
  - Toimittaja on voinut olla siellä ”töissä” jo pitkään
- Automatisoitujen testausohjelmien skriptit, joilla testejä voi toistaa omassa ympäristössä
  - Esim. kuormitustestauksen skriptit
- Apua ongelmissa

# Projektivalmennus tarpeen

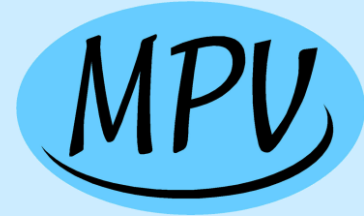
- Organisaatioissa ei kaikilla ole projektikokemusta eikä tietoa, miten projekteissa pitää toimia
- On muistettava kouluttaa liiketoiminnan ihmisiä paitsi testaamiseen myös projektitoiminnan logiikkaan:
  - Tavoitteet.
  - Kenelle raportoidaan.
  - Aikakysymykset.
  - Uudet välineet.
  - Jne...

# Ns. liiketoimintatestaajien käyttäminen testaukseen 1/3



- Testaajia järjestelmän käyttäjistä, järjestelmätuesta jne...
  - Ihmisiä, jotka oikeasti tietävät, miten systeemin pitää toimia ollakseen heille kelpaava!
- Eivät osaa testata
  - Osaavat toki käyttää ja kokeilla
- Suhtautumistapa sopii tehokkaaseen ja mukavaan käyttöön, mutta ei testaukseen
  - Ks. tarkemmin seuraavalla kalvolla
- Oma rooli organisaatiossa on välineitä käyttävä, mutta ei niitä kritisoiva
- Mutta: mitä kypsempi ja aikuisempi organisaatio, sitä paremmin testauskin onnistuu

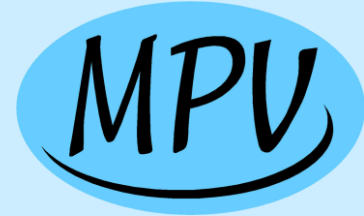
# Ns. liiketoimintatestaajien käyttäminen testaukseen 2/3



- ”Asenneongelmia” testauksessa:
  - Käytetään ”oikein” (kun osataan!) <> mutta väärinkäyttö paljastaa bugit
  - Syytetään itseä ongelmista <> pitää syyttää systeemiä, jossa on vielä bugeja
  - Koko ammatti-ikä on opeteltu pärjäämään huonojen järjestelmien kanssa ja kiertämään niiden vikoja (se on ammattitaitoa!) ja nyt pitäisi opetella avuttomaksi...
  - Ei tykätä valittaa, koska valittajat ovat hankalia ihmisiä <> testaajan pitää kertoa pienistäkin asioista
  - Ei olla opittu kyseenalaistamaan järjestelmiä, kun niitä suunniteltaessakaan ei saa vaikuttaa riittävästi <> miten osaisi – *ja miksi viitsisi* – olla kriittinen testatessaan?



# Ns. liiketoimintatestaajien käyttäminen testaukseen 3/3



- Pitää prepata!
  - Päivän perehdyttäminen testaukseen ei ole liikainvestointi
- Mutta täytyy myös pitää testaustavat helppoina ja antaa apua
  - Tarvitaan luontevia ohjeita, joihin ihmiset ovat tottuneet
  - Testaus voi olla etukäteen suunniteltua tai ketterää tilanteesta riippuen
- Ryhmätyössä ihmiset oppivat toisiltaan ja saavat tukea testaukseen
- Asioita pitää käsitellä ihmisten kanssa erikseen ja yhdessä

# Testattavat asiat ja järjestys 1/3

- Systemaattista hyväksymistestausta on edeltänyt jo jonkin aikaa kestänyt yhteistyö määrittelyn, suunnittelun ja toteutustenkin merkeissä, jossa on paljon sovittu ja paljon sovittu uudestaan...
- Ensimmäinen tehtävä on tarkistaa, että kaikki sovittu on tehty ja kaikki, mistä on valitettu, on korjattu – muuten ei ole mitään toivoa...
- Testauksessa on tärkeää riskiperusteisuus. Eli aloitetaan tärkeimmistä asioista, joiden pitää onnistua
  - Tärkeimmät käyttötapaukset,
  - Kokonaiset bisnesprosessit, joissa dataa kulkee koko systeemin läpi
  - Jne...

## Testattavat asiat ja järjestys 2/3

- On tärkeää paitsi kokeilla, myös tarkistaa asioita käsin ja silmin – vaikkapa tietokannan rakenteen oikeellisuus yms.
- Käytettävyytestaus ja käytettävyyden arviointi on tärkeää. Käyttöliittymiä ei saa koskaan hyväksyä pelkkien kuvien perusteella.
- Suorituskykytestaus pitää tehdä.
- Tietoturvatestauksesta puhumattakaan.
- Aineistojen siirto ja tietokantojen konvertointikin pitää testata.
- Laatuominaisuuksien taustalla on kuitenkin myös ajatus siitä, millaisesta laadusta on maksettu. Täydellisyys maksaa äärettömästi, ja tilaaja on maksanut vain kohtalaisesti...

# Testattavat asiat ja järjestys 3/3

- Ylipäätään siis on testattava:
  - Kaikki todelliset tavat käyttää järjestelmää
  - Kaikki *relevantit* asiat, mitkä on mainittu määrittelyissä, sopimuksissa ja suunnitelmissa.
  - Siksi niitä täytyy pitää yllä, ettei sovittuja asioita tarvitse kaivella sähköposteista
  - Ja myös ”standardivaatimukset”, joita ei tarvitse edes mainita! Esimerkiksi käyttäjien hallinnan pitää olla ok, vaikka sitä ei mainitsisi mikään dokumentti. Samoin tietoturvallisuuden
  - ...mutta joskus määrittelyt ovat vain valittamisen perusta – käyttöön sopivuus on tärkeämpää kuin paperit
- Jossain vaiheessa todetaan, että testattava versio ei kelpaa. Silloin ei kannata jatkaa testausta, vaan vaatia uusi versio ja aloittaa alusta.

# Testidata

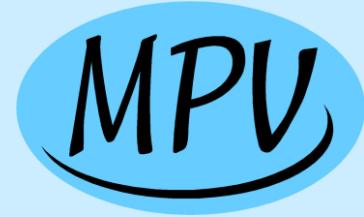
- Todellisen kaltaista... mutta suojeltava oikeaa dataa
- Testitietokannan luomismahdollisuus tärkeä
  - Mutta mites kaikki muut järjestelmät, joiden kanssa järjestelmä on yhteyksissä?...
  - Dataan kaikki mahdolliset virheet
- Äärimmäinen varovaisuus
- Aineistojen bulkkisiirrot tärkeä testata – ja niiden nopeus – jolle pitää asettaa vaatimuksia (ettei vie montaa päivää)

# Testiautomaatio?

- Testiautomaatiolla on vähäinen rooli hyväksymistestauksessa
- Tärkeintä on testata oikeaa käyttöä vastaavilla tavoilla
  - Siis manuaalisesti
- Poikkeuksia
  - Suorituskykytestaus
  - Tietoturvatestauksen eräät muodot (erit. murtautumistestaus)
  - Niissä tarvitaan automaation apua ja luodut testit ovat käyttökelpoisia jatkossakin systeemin toimintaa tarkistettaessa

# Inkrementaalinen hyväksyntä 1/2

- Monia asioita voidaan hyväksyä myös inkrementaalisesti, palanen kerrallaan.
  - Määrittelyt ovat tällaisia, jos ne eivät muutu....
  - Toteutuksia taas koskee regressioriski, eli niiden hyväksyntä ”peruuntuu” jossain määrin uusien versioiden myötä.
  - Koska useimpia toimintoja koskee myös ketterässä toiminnassa ketju määrittelystä toteutukseen, on oltava tarkkana, mitä hyväksyy!
- Seuraavalla sivulla on *havainnollistus* siitä, miten erilaisia asioita voidaan hyväksyä vaiheittain



# Inkrementaalinen hyväksyntä 2/2

| Aika ->             | Määrittely / prototyyppi   | Julkistukset   | Julkistukset   | Hyväksyntä   |
|---------------------|--|--|--|--|
| Vaatimukset yleensä | Katselmointi ja hyväksyntä   |  | Vertailu toimitukseen  | Hyväksyntä<br>Vertailu sopimukseen ja määrittelyihin ja odotuksiin |
| Toiminnallisuus     | Katselmointi ja hyväksyntä   | Toiminnallisuuden hyväksyntä yksittäisillä testeillä (positiiviset testit) | Toiminnallisuudet käyttäjänäkökulmasta Robustius (negatiiviset testit). Tietovar. tarkistus. | Regressiotestit  |
| Integraatio         |  | Päästä päähän testit   | Kokonaisjärjestelmän robustiuden testaus   | Regressiotestit  |
| Käyttöliittymä      | Look and feel:n hyväksyntä<br>• Perusratkaisujen käytettävyyden analyysi | Käytettävyyden arviointi, testit   | Räätälöintien ja muutosten käytettävyyden varmistaminen                                      |  |
| Tietoturva          | Tietoriskianalyysi, perusratkaisujen tarkastaminen ja analyysit          | Tarkemmat analyysit ja testit  |  |  |
| Suorituskyky        | Vaatimusten sopiminen  |  | Suorituskyky- ja kuormitustesti  |  |
| Luotettavuus        | Vaatimusten sopiminen  | Kirjanpito tuotantovirheistä   | Kirjanpito tuotantovirheistä   | Vertailu vaatimuksiin  |

**VISUALISOINTI  
EI EKSAKTI**



# Vikatiedon keruu ja käsittely

- Kun bugeja löytyy, ne pitää kerätä jonnekin
- Tarvitaan tulkintaa ja muotoilua kuvaaviksi
  - Testauspäällikkö tai ”error manager”
- Excel riittää yleensä ihan hyvin
- Oikea vikakanta voi kannattaa isoissa hankkeissa
  - Bugzilla, Mantis tms. avoimen lähdekoodin systeemi
  - Voidaan jakaa kaikkien osapuolten kesken – bugit saadaan heti korjaukseen
- Kuitenkin tietojärjestelmien pitää olla kevyitä – pääpaino on ongelmien etsimisessä!
- Käsittelyyn usein ”raati” ohjelmistotoimittajan kanssa, jossa käydään listaa läpi ja pohditaan, mitä bugeille tehdään – ja kenen laskuun!

# Yleinen toimintamalli tarpeen

- Vähänkin isommassa organisaatiossa otetaan vuosittain käyttöön uusia järjestelmiä
- Silloin pitää olla sovittu toimintamalli, minkä mukaan hankinnoissa toimitaan
  - Osa laatu järjestelmää
  - Lähtökohtana vaikkapa ohjelmistohankintojen CMMI – CMMI for Acquisition
  - Prosessi, vastuut, roolit, yhteistyö, laadunvarmistus

## Ketterän kehittämisen mahdollisuudet ja potentiaaliset ongelmat hyväksynnälle

- Ketterä kehittäminen tuottaa inkrementaalisuuden ansiosta etuja:
  - Systeemin toiminnan oikeellisuuden hyväksyminen voidaan tehdä osissa
  - Pienissä palasissa kaikki osapuolet ymmärtävät hyväksyttävät asiat paremmin ja ristiriitoja on vähemmän
  - Hyväksyminen on lähempänä määrittelyä, jolloin prosessi on eheämpi
- Hyväksymisen asteessa on oltava tarkka:
  - Esimerkiksi käyttöliittymän look and feel voidaan hyväksyä suunnitelmien perusteella, mutta sen oikea käytettävyys arjessa ja toteutuksen detaljit vaativat käyttökokeita

## Scrumin ”hyväksymistestauksen” kritiikki 1/4

- Jokaiseen sprinttiin liittyy jonkinasteinen asiakkaan hyväksyntä vähintään katselmoinnilla.
- Asiakkaalle toimitettuihin julkistuksiin liittyy hyväksymistestaus.
- Perinteinen ketterien menetelmien tapa on se, että asiakkaan edustaja määrittää hyväksymistestit ja ne tekee asiakas, testaajat tai kehittäjät. Niitä pyritään joskus myös automatisoimaan.
- **Tämä ajattelu on täysin riittämätön.**

## Scrumin ”hyväksymistestauksen” kritiikki 2/4

- Nämä testit lähinnä validoivat vaatimuksen algoritmisen toteutuksen bisnesprosessien näkökulmasta
  - Jokin raportti syntyy korrektisti
  - Lomakkeella saadaan tietoa tietokantaan
- Kyse on hyväksynnästä vain tässä kontekstissa
  - Ei asiakasorganisaation hyväksynnästä!
  - Vasta oikeastaan savutesti – sen jälkeen kunnollinen järjestelmätestaus

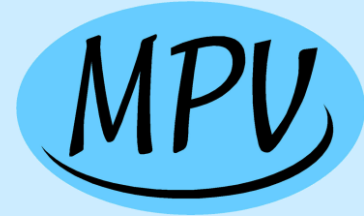
## Scrumin ”hyväksymistestauksen” kritiikki 3/4

- Aito hyväksymistestaus merkitsee asiakkaan ja käyttäjien hyväksyntää.
- Se on välttämätöntä tehdä avoimesta näkökulmasta, riittävän monipuolisesti ja ei-mekanistisesti.
- Se ei saa jäädä asiakkaan ”edustajan” tehtäväksi, vaan mukaan on saatava asiakkaan organisaatio tekemään:
  - Professionaalista testisuunnittelua.
  - Professionaaliset testit.
  - Pätevät päätökset hyväksynnästä.
- Tietojärjestelmäkehityksessä asiakkaalta tarvitaan usein myös – kenties monitasoinen – hyväksymistestausympäristö.

## Scrumin ”hyväksymistestauksen” kritiikki 4/4

- Hyväksymistestaus on suhteellisen jatkuva aktiviteetti.
- Koska sprintit tuottavat säännöllisesti uusia versioita asiakkaalle, niiden testaus tapahtuu monimuotoisesti:
  - Uusien ominaisuuksien hyväksymistestaus.
  - Uuden version testaus käytännössä, tuotantokäytössä.
- Esimerkiksi tietojärjestelmillä testaus voi olla monitasoista ja uusi järjestelmäversio läpäisee monet asiakkaan testitasot ja testijärjestelmät, ennen kuin se on todettu tuotantokelpoiseksi.

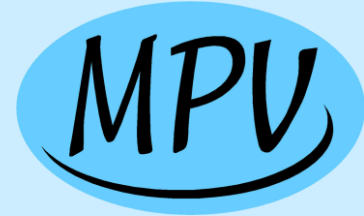
# 10 keskeistä menestystekijää tilaajalle



1. Vahva järjestelmähankinnan kokonaisprojekti
2. Oivallus hyväksymistestauksen tarpeellisuudesta ja vahva tunne sen välttämättömyydestä
3. Itsenäisyys – ei anneta toimittajan vedättää
4. Testauksen selkeä projektointi
5. Osallistuvien valmennus projektiin ja testaamiseen
6. Testaajien auttaminen aktiivisesti
7. Asiantuntija-avun hankkiminen
8. Riskianalyysi
9. Riskiperusteinen testaus
10. Eri osapuolten yhteistyö

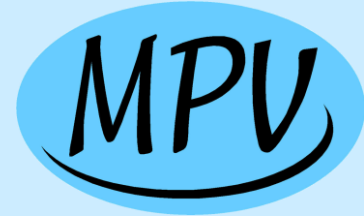


# 10 keskeistä menestystekijää järjestelmätoimittajalle 1/2



1. Oivallus, että tilaajan tekemä pätevä hyväksymistestaus on toimittajankin etu
2. Pätevä vaatimusmäärittely ja järjestelmän suunnittelu, jolla varaudutaan hyväksymistestaukseen
3. Selkeät sopimukset ja suunnitelmat, joiden avulla tiedetään toimituksen kokonaisuus ja laatutaso (paljonko saa olla pikkubugeja...)
4. Inkrementaalinen projekti, jossa asioita hyväksytään vähitellen – ettei tule yllätyksiä
5. Ohjelmistokehittäjille luotava ymmärrys järjestelmästä ja asiakkaan tarpeista

# 10 keskeistä menestystekijää järjestelmätoimittajalle 2/2



6. Omien testausvaiheiden toteutus asianmukaisesti
7. Vaatimus päästä tekemään omaa testausta hyväksymistestauksen ympäristössä
8. Keskustelu tilaajan kanssa tulevasta hyväksymistestauksesta
9. Hyvä yhteistyö tilaajan kanssa helpottaa yhteistyötä ja tilaajan tekemän testauksen onnistumista
10. Pätevä projektinhallinta

# Yhteenveto

- Hyväksymistestaus on tilaajille oikeasti haastavaa.
- Tilaajien on opittava itsenäisyyttä, laatuajattelua ja toimittajan kyseenalaistamista.
- Järjestelmähankintojen ja hyväksymistestauksen projektointi on syytä tehdä pätevästi.
- On luovuttava teknologiapositivismista ja muistettava, että useimmat ohjelmistot eivät täytä lupauksiaan.
- Testauksen luonne osaamisalueena on oivallettava.
- ...Nämä ovat kovia haasteita tietojärjestelmiä hyödyntäville tavallisille organisaatioille...