

Legacy

maaretp

# Strategioista suunnitelmiin – selkeyttä käsitteiden sekamelskaan

Maaret Pyhäjärvi



by Maaret Pyhäjärvi is licensed under CC BY 4.0



<https://www.linkedin.com/in/maaret/>



[@maaretp@mas.to](https://twitter.com/maaretp)

# Sisältö

- Perinteinen ja ketterä ohjelmistokehitys – suunnittelun rooli
- Testauksen käsitteistön haaste testaussuunnittelulle
- Testausstrategian monet ulottuvuudet



# Suunnitelmaohjatut ja ketterät arvot

Lähde: Agile Alliance website <<http://www.agilealliance.com>>

## Suunnitelmaohjatut arvot

Prosessit ja työkalut

Perusteellinen dokumentaatio

Sopimusneuvottelu

Suunnitelman seuraaminen

## Ketterät arvot

Yksilöt ja vuorovaikutukset

Toimiva ohjelmisto

Asiakasyhteistyö

Muutokseen vastaaminen



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

# Suunnitelmaohjatut ja ketterät menetelmät

## Suunnitelmaohjattu

- Hyvin ymmärrettyihin, toistettaviin prosesseihin
- Määritelty, ennustettavissa oleva suunnitelma
- Suunnitelma projektin keskipisteenä
- Suunnitelma tehdään tavoitteiden saavuttamiseksi (vaatimukset)
- Seurataan:
  - etenemää
  - poikkeuksia suunnitelmista
- Korjaustoimenpiteillä päästään takaisin suunniteltuun tilaan

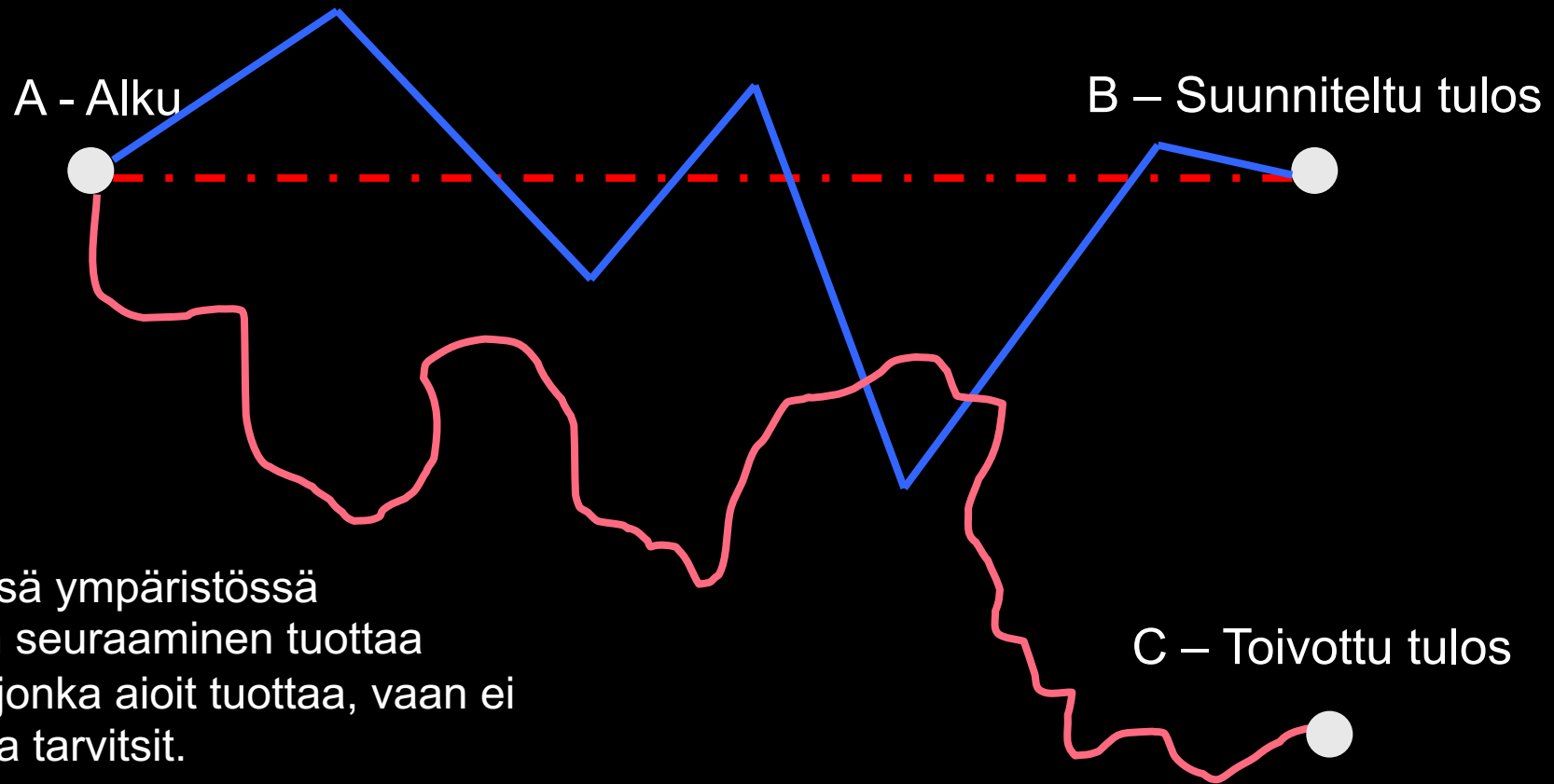
## Ketterä

- Monimutkasiin, odottamattomiin prosesseihin
- Tulokset projektin keskipisteenä
- Muutoksia ei voida eikä haluta välttää, ne kuuluvat ohjelmistokehityksen luonteeseen
- Ohjelmistojen rakentaminen on oppimiskokemus
- Oppiminen muuttaa suunnitelmaa

Usein verrataan raskas vs. kevyt ja formaali vs. vapaamuotoinen - pääeroavaisuus puuttuu!



# Suunnitelmaohjattu vs. ketterä – miksi?



Äärimmäisessä ympäristössä suunnitelman seuraaminen tuottaa sovelluksen, jonka aioit tuottaa, vaan ei sovellusta jota tarvitsit.



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

# Ketterä testaus

- Kaksi pääsuuntausta:
  - Tutkiva testaus (exploratory testing)
    - Lähtöisin testausyhteisöstä
    - Ottaa kantaa ”järjestelmätestaukseen”
  - Ääritestaus (extreme)
    - Lähtöisin ohjelmointiyhteisöstä (extreme programming)
    - Ottaa kantaa ”yksikkötestaukseen”
    - Vaatii korkean tason testit (”hyväksymistestit”) mutta ei määrittele yksityiskohtia kuinka nämä sisällytetään prosessiin



# Testauksen suunnittelu

- Keskeiset testauksen käsitteet
  - Testaustaso
  - Testausvaihe
  - Testaustyyppi
  - Testauskierros
  - Testausstrategia
- Kommunikointia
  - Lopputuloksena usein suunnitelma – kommunikaatioväline
    - Suunnitelman muoto vaihtelee tarpeen mukaan valtavasti



# Testaustaso

- Jatkuvaa testaustoimintaa tietyn tyyppisen testaustavoitteen ja testauskohteen ympärillä. Testaustoiminnasta tekee oleellisesti jatkuvaa vaatimus uusintatestauksesta muutoksen osalta.
- Tapa pilkkoa vaikeita kokonaisuuksia järkeviin näkökulmiin
- Englanniksi ladattu merkitys, vertaa level vs. stage





# Testaustyyppi

- Ryhmä testausaktiviteettejä, joilla on yhteisiä ominaisuuksia joiden perusteella ne voidaan tunnistaa omana luokkana, ja jotka on ryhmitelty arvioimaan yhtä tai useampaa toisiinsa liittyvää laatuominaisuutta.
  - Testaustyyppi voi sijoittua yhdelle tai useammalle testustasolle ja testausvaiheeseen.
  - Kaikki testaustyypit eivät ole oleellisia kaikissa asiayhteyksissä.
- Käytännössä testaustyypit usein muodostavat testausta suunnittelevan henkilön tarkastuslistan katettavista asioista
- **Toiminnallinen testaus** on ohjelmiston tarjoamien toimintojen testaamista yksittäisinä toimintoina, toimintoryhminä sekä yhdistettynä ohjelmiston aineiston kanssa.
- **Ei-toiminnallinen testaus** kohdistuu nk. ei-toiminnallisiin ominaisuuksiin, jotka vaikuttavat ohjelmistoon liittyvään kokonaislaatukokemukseen, mutta eivät ole liitettävissä suoraan toimintoon tai toimintoryhmään ohjelmistossa, kuten käytettävyys, suorituskkyky ja luotettavuus



# Toiminnallisen testauksen testaustyyppejä

- **Toiminnallisuustestaus** (functionality testing, feature testing)
- **Yhtäaikaistuustestaus** (concurrency testing)
- **Asennustestaus** (installation testing)
- **Alustatetaus** (platform testing)
- **Aloitustestaus** (build verification testing, smoke testing)
- **Konfiguraatiotestaus** (configuration testing)
- **Yhteensopivuustestaus** (compatibility testing)
- **Rinnakkaistestaus** (end-to-end testing)
- **Rajapintatetaus** (interface testing)
- **Poikkeustilannetetaus** (recovery testing)
- **Lokalisointitetaus** (localization testing)
- **Dokumentaation testaus** (documentation testing)
- **Aineiston laadun testaus** (data quality testing)
- **Alfatetaus** (alpha testing)
- **Betatetaus** (beta testing)
- **Muuntotetaus** (conversion testing)
- **Tuotantotetaus** (production testing, operational testing)
- **Standardien testaus** (standards testing)

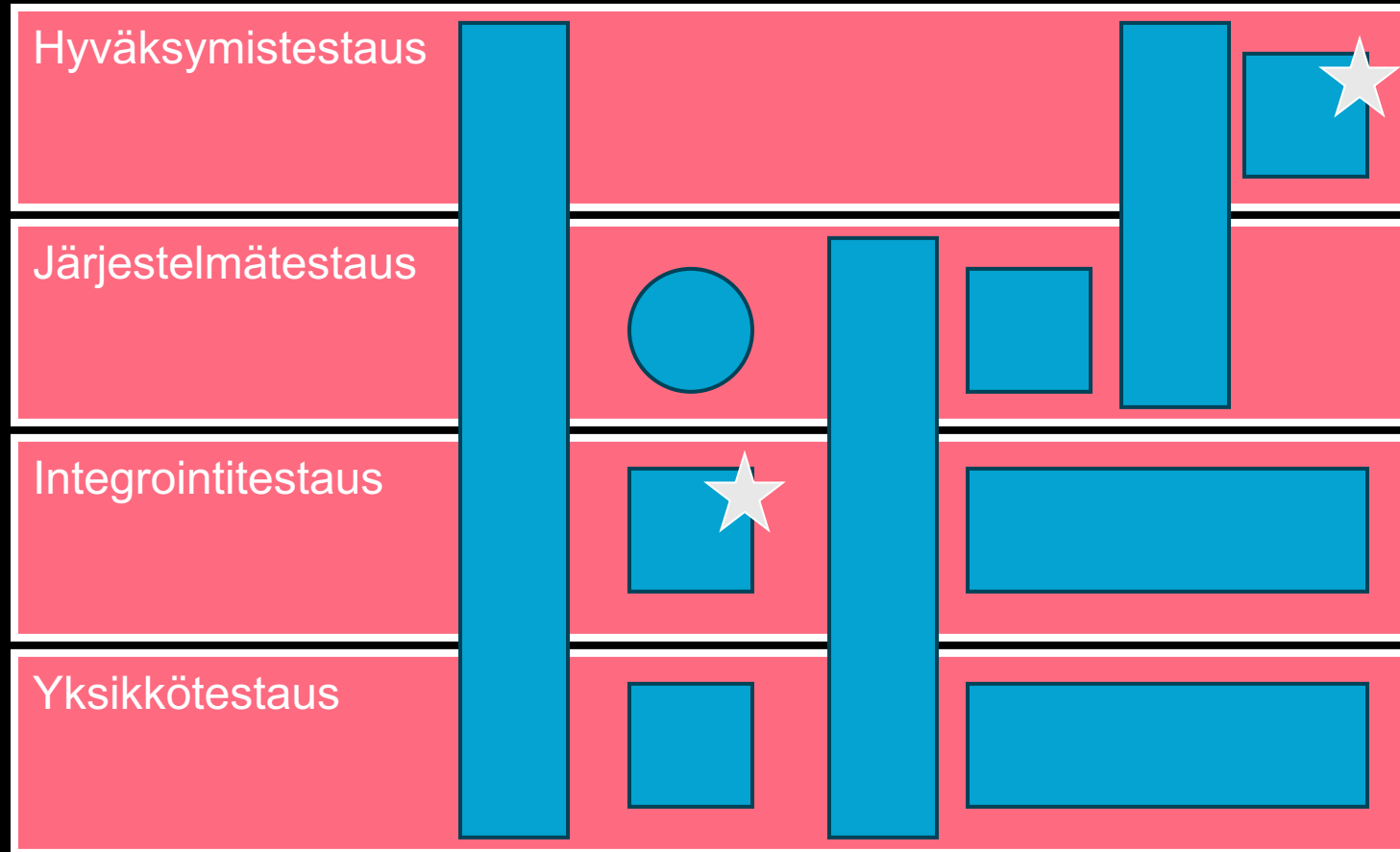


# Ei-toiminnallisen testauksen testaustyyppejä

- **Luotettavuustestaus** (reliability testing)
- **Suorituskykytestaus** (performance testing)
- **Kuormitustestaus** (load testing)
- **Rasitustestaus** (stress testing)
- **Paljoustestaus** (volume testing)
- **Kestävyystestaus** (endurance testing)
- **Tietoturvatestaus** (security testing)
- **Käyttöturvallisuuden testaus** (safety testing)
- **Käytettävyystestaus** (usability testing)
- **Esteettömyystestaus** (accessibility testing)
- **Palautettavuustestaus** (recoverability testing)
- **Tuettavuustestaus** (supportability testing)
- **Ylläpidettävyystestaus** (maintainability testing)
- **Siirrettävyystestaus** (portability testing)
- **Koodin laadun testaus** (code quality testing)



# Tasot, vaiheet ja tyypit

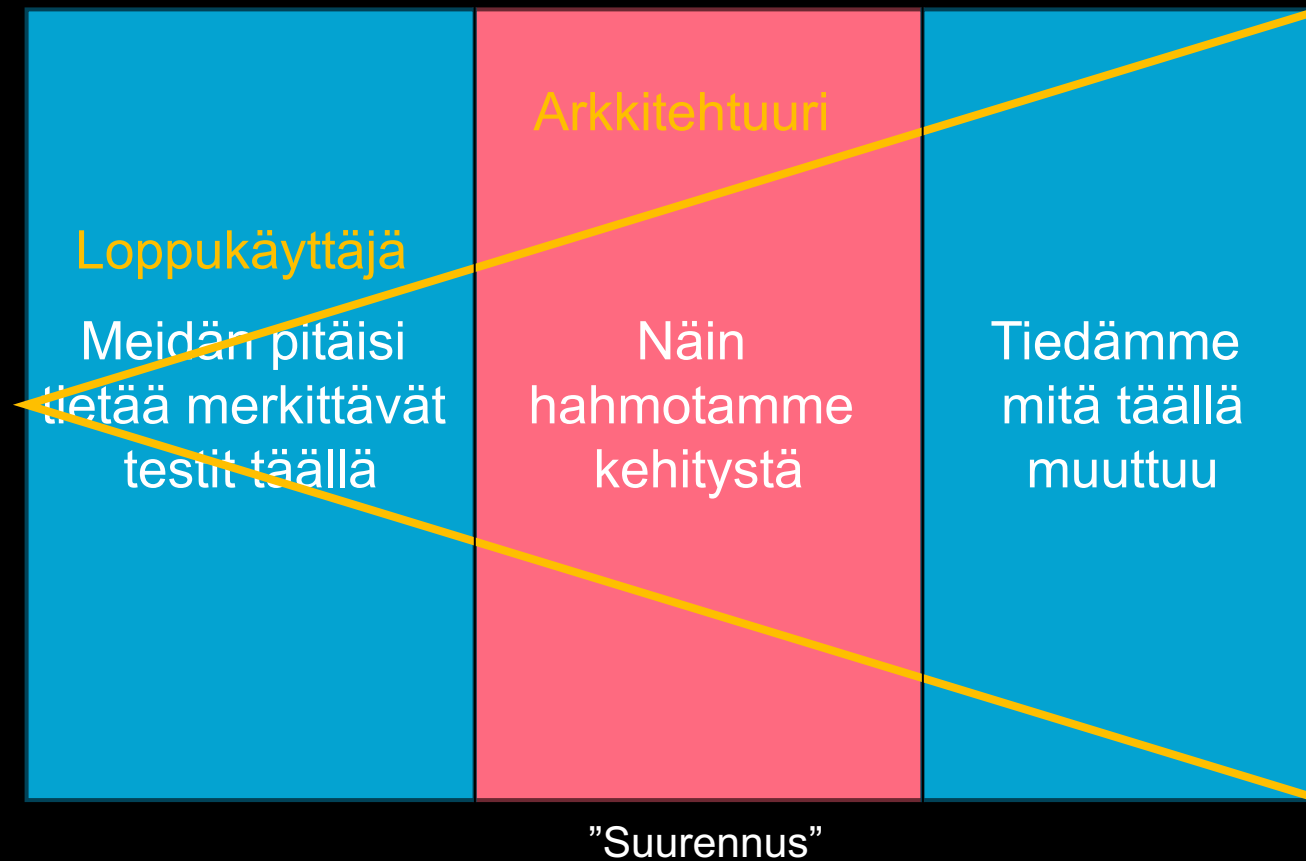


# Testikierros

- Valittujen, yhden tai useamman testijakson tai testijakson osan suorittaminen kertaalleen.
- Osa testeistä epäonnistuu, joten koko kierrosta ei usein saada ajettua
- Ohjelmisto muuttuu, joten joitain testejä pitää ajaa uudelleen
- Testikierrosten lukumäärä on keskeinen aikatauluun vaikuttava tekijä



# Käyttäjän tehtävistä sovelluksen komponentteihin



# Testitapausarkkitehtuuri

## Testien jäsentämisen testijaksot:

- Testattavat asiat
  - Aineistot
  - Ryhmittely
- Tärkeysjärjestykseen laittaminen
- Ajamisen valintakriteerien tukeminen

## Testauksen suorituksen testijaksot:

- Testattavat asiat vs. kokoonpanot
  - Testikierrokset
- Linkki raportointiin



# Testijakso

- Testitapausten suunnittelun ja suorittamisen kokonaisuus, johon testitapaukset ryhmitellään. Testijakso sisältää ne testitapaukset joilla testataan yksi järjestelmän / sovelluksen looginen kokonaisuus. Jakonäkökulma voi olla käyttö ja vaatimukset (järjestelmätestaustaso) tai ohjelmistorakenteet (ohjelma- ja integrointitestaustasot)





# Miksi strategia on vaikea käsite

- Testausstrategia == Testaustekniikka (Beizer)
- Testausstrategia on yksityiskohtaisempi kuin testaustekniikka (Bach)
- Testausstrategia = Testausprosessi (ISEB Practitioner)
- Strategia liiketoiminnassa – jotain laajempaa
  - Paras arvaus mahdollisuuksista ja uhkista suunnittelun näköpiirissä kaavailluille muutoksille ja toimenpiteille
  - Lopputulosten saavuttaminen, ei määrittäminen
  - Mitä organisaation tulisi tehdä?
  - Mitkä ovat päätökset, joita tavoittelemme ja kuinka voimme saavuttaa ne?



# Testausstrategia

- Testausstrategian tarkoituksena on mahdollistaa testausta saavuttamaan tavoitteensa
- Kuinka suunnittelemme kattavamme tuotteen siten että saadaan aikaan riittävä laadun arviointi
- Testausstrategian tarkoitus on selvittää testausprojektin suurimmat tehtävät ja haasteet



# Testauksen suunnittelu

Testauksen missio ja  
korkean tason tavoitteet

Testausstrategia

Yleistestaussuunnitelma

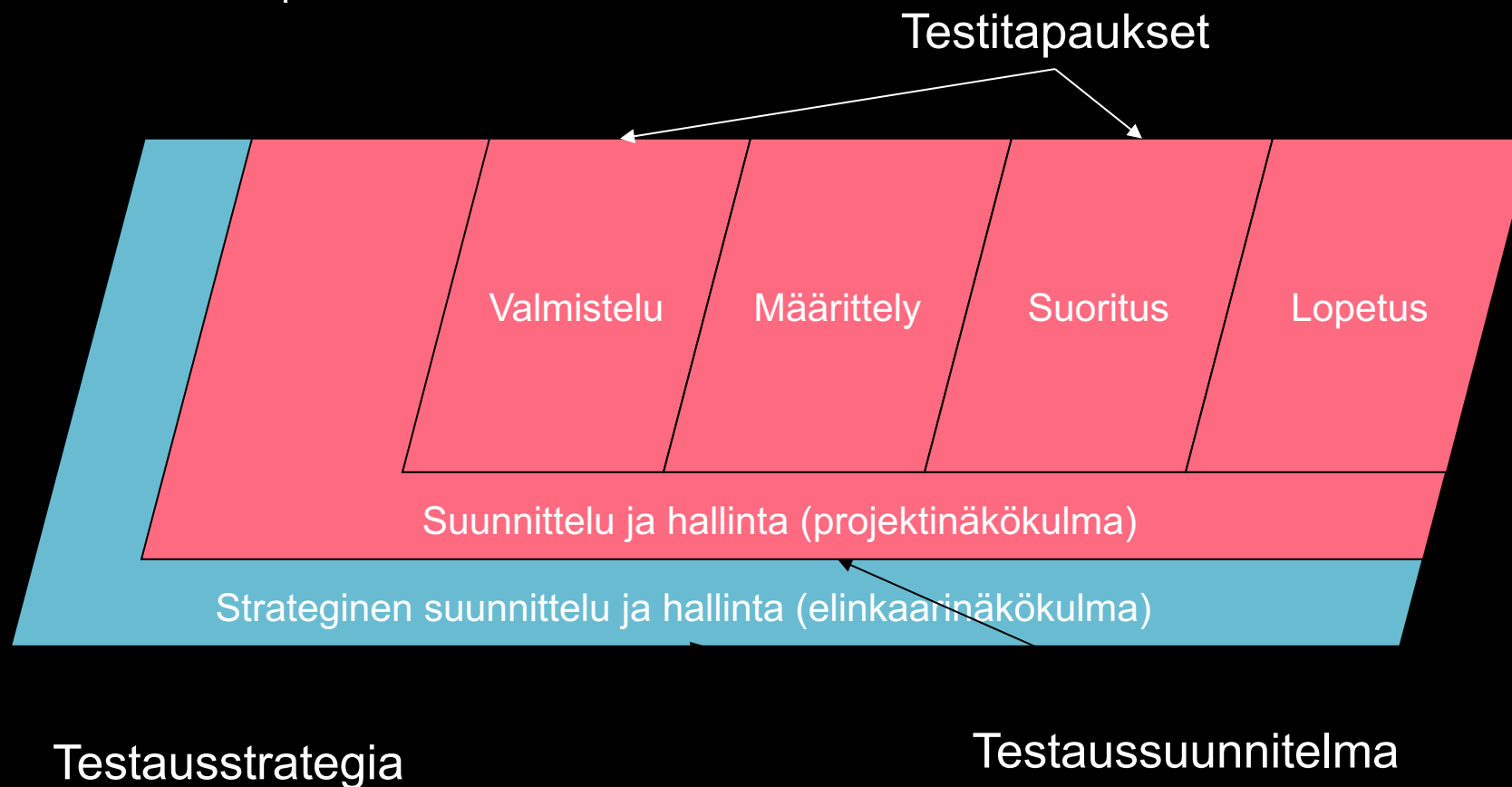
Testaussuunnitelma

- Organisaation odotukset testaustoiminnalle
- Laatupainotukset sovellusalueella
- Järjestelmä / tuote / palvelukohtainen
- Useita projekteja saman järjestelmän elinkaareissa
- Projektikohtainen
- Projektin lähestymistapa, strategian räätälöinti
- Tarkentavat suunnitelmat ryhmittäin



# Testausprosessi

Muokattu: Pol & Van Veenendaal.TMap.



# Projektin näkökulma testaussuunnitteluun

Lähde: Pol & Van Veenendaal.TMap.

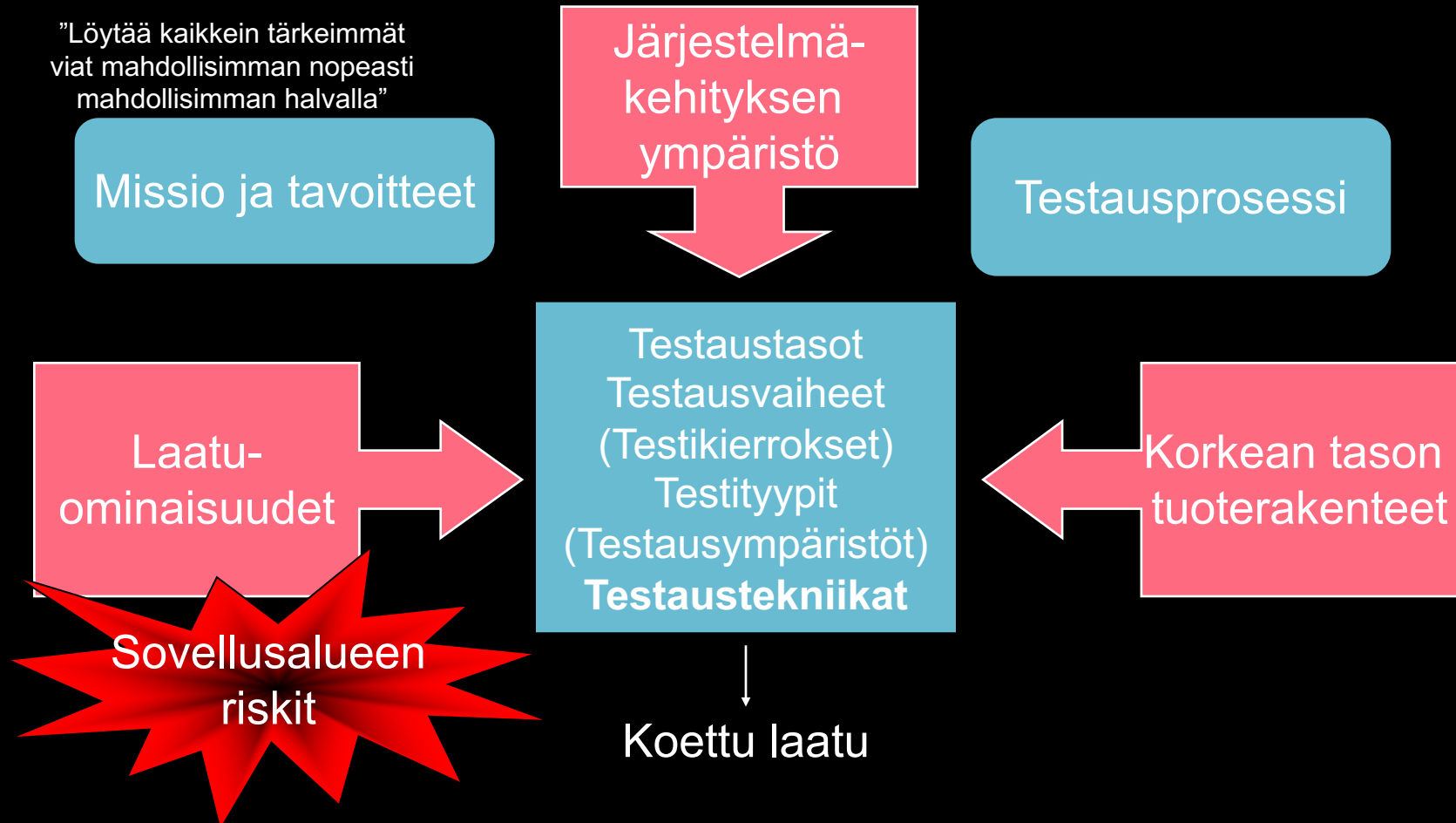


<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to

# Mitä sisällyttää testausstrategiaan

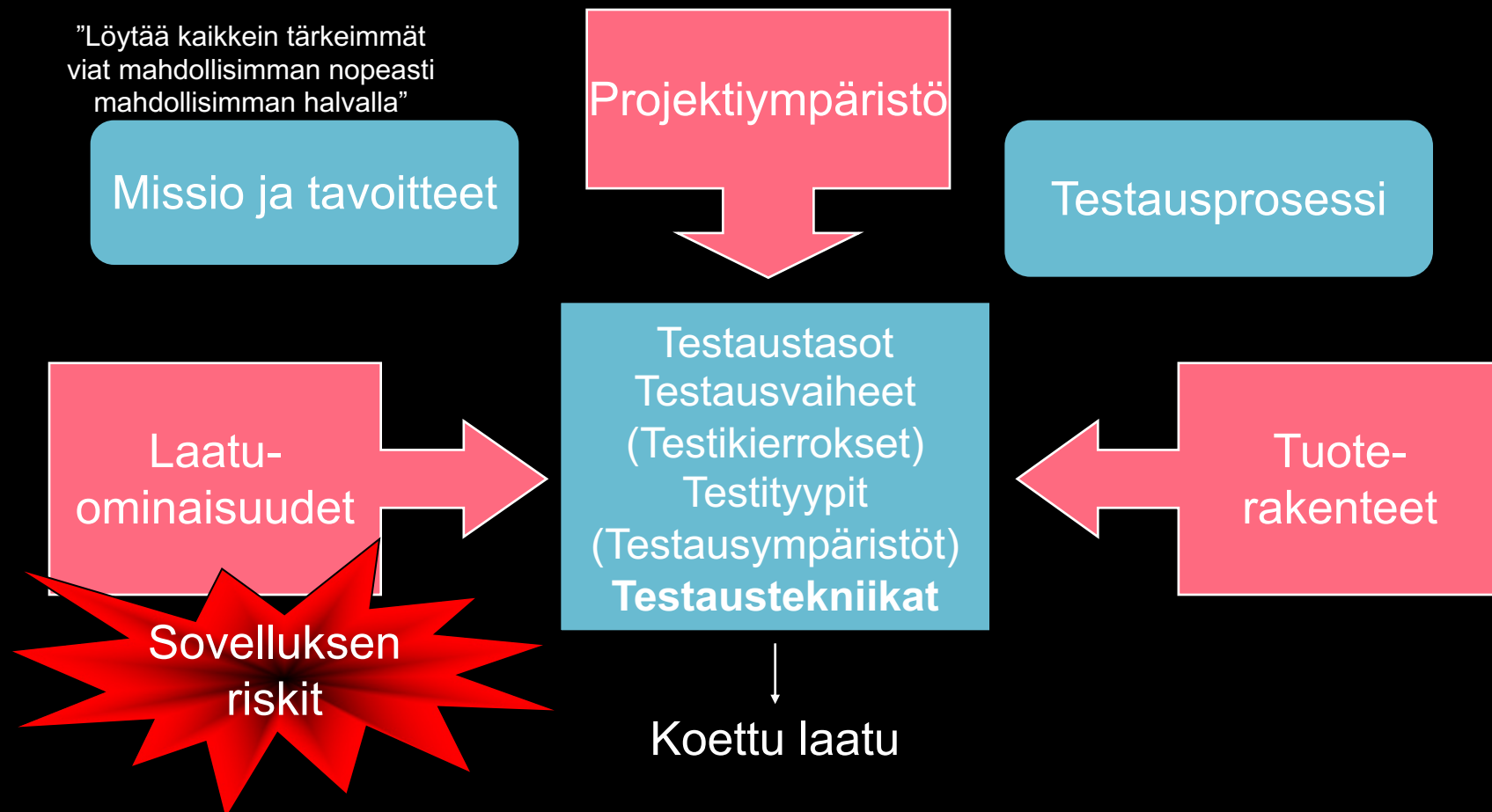


# Miksi - Entä jos?

- Testausstrategiat yleensä
  - Tehdään useille projekteille TAI
  - Tehdään yhdelle projektille ja käytetään uudelleen seuraaville
- Räätälöintikohdat pitää osoittaa
  - Tyypilliset omat tulkinnat rikkovat kokonaisuuden
  - Vaihtoehtojen tunnistaminen jo tehdessä järkevää



# Mitä sisällyttää projektin yleistestaussuunnitelmaan





# Testauksen suunnitelman sisältö

- Helppo vastaus: IEEE 829 sisältö, 16 otsikkoa
- Lähestymistapa
  - Laaja käsite
  - Strategia pääosin tässä
- Testitapausarkkitehtuuri
  - Testauksen kohde
  - Mitä testataan ja mitä ei
  - Ei pitäisi olla speksin kopio
  - Testaaja näkökulma (vs. suunnittelijan näkökulma)
- Tehtävät, Aikataulu & Roolit
- Resurssit, osaaminen, tarvittava koulutus



# Yleistestaussuunnitelman hahmottelu (1/5)

- Tilanteen ja asiayhteyden ymmärtäminen
  - Mistä tilanteesta testausta lähdetään suunnittelemaan?
  - Millainen on testauksen asiayhteys ja miten se vaikuttaa?
- Sidosryhmien tunnistaminen ja testauksen tavoitteet
  - Mitkä ovat testauksen sidosryhmät ja mikä on näiden sidosryhmien painoarvo?
  - Mitkä tekijät vaikuttavat testauksen onnistumiseen näiden sidosryhmien näkökulmasta?
  - Mitkä ovat testauksen tavoitteet niin että ne sisältävät onnistumisen elementit sidosryhmien näkökulmasta?



# Yleistestaussuunnitelman hahmottelu (2/5)

- Tuen ja liittoutumien muodostaminen
  - Mitkä ovat testauksen tekemisen onnistumisen edellytykset? Minkä käytäntöjen pitää olla selvinä?
  - Mitä johtoryhmän tuen osalta vaaditaan ja kuinka se saavutetaan?
  - Ketkä toimivat päätöksentekijöinä testaukseen liittyvissä päätöksissä? Miten osoitetaan että testaus on tehokasta näiden päätöksentekijöiden näkökulmasta?
  - Mitä liittoutumia ja yhteistyökuvioita pitää ylläpitää onnistuneen viestinnän aikaansaamiseksi?



# Yleistestaussuunnitelman hahmottelu (3/5)

- Testauksen rajauksen ja prioriteettien määrittely
  - Mitkä testaustyypit ovat mukana testauksessa?
  - Miten laatuominaisuudet ja laadun arvotekijät painottuvat tälle järjestelmälle?
  - Voidaanko laatuominaisuuksia ja prioriteetteja jakaa osajärjestelmille?
  - Liittyykö testaukseen aika- tai resurssirajauksia? Miten näiden rajoitteiden puitteissa voi toimia mahdollisimman hyvin?
  - Puuttuuko jotain päätöksiä testauksen rajaamiseksi?



# Yleistestaussuunnitelman hahmottelu (4/5)

- Koordinointi järjestelmäkehitysaktiviteettien kanssa
  - Miten koonnit julkaistaan testaukseen ja saadaan tieto koonnin sisällöstä?
  - Miten virheet raportoidaan ja käsitellään sekä miten toimintatapa muuttuu projektin edetessä?
- Testauksen painotusten määrittely riskipohjaisesti
  - Mitkä ovat merkittävimmät riskit?
  - Miten testaus pitäisi suunnata näihin riskeihin pohjautuen?
  - Mitä alueita ei tarvitse testata tai voidaan testata hyvin kevyesti?
  - Kuinka testauksen työmäärä jaetaan eri testaustyyppeihin?



# Yleistestaussuunnitelman hahmottelu (5/5)

- Kattavuustavoitteiden konkretisointi
  - Kuinka paljon testausta tarvitaan? Millaiset kattavuustavoitteet pitäisi olla? Kuinka nämä mitataan ja arvioidaan?
- Testien suunnittelutekniikoiden valinta
  - Mitä tekniikoita käytetään testivaatimusten tunnistamiseen, testitapausten luomiseen, testitapausten katselmointiin?
  - Mitä keinoja käytetään testien läpimenon arviointiin?
  - Mitä keinoja käytetään väärin hälytysten välttämiseen?
- Testiautomaatiosuunnitelman määrittäminen
  - Mitä osia testauksesta voidaan automatisoida, millä välineillä ja millainen työmäärä siihen liittyy?



# Testaussuunnitelman konkretisointi (1/3)

- Testauksen aloitus- ja lopetuskriteerien jäsentäminen
  - Minkä kriteerien pitää täytyä testauksen aloittamiseksi?
  - Minkä kriteerien pitää täytyä testauksen lopettamiseksi?
- Testausresurssien tunnistaminen ja organisointi
  - Mitä resursseja testaukseen tarvitaan? Ketkä suorittavat testauksen? Missä tiloissa ja millä välineillä? Millaisin toimintatavoin?
  - Millaisia rooleja ja vastuita testauksessa on?
- Oikeiden tehtävien valinta oikeille ihmisille
  - Miten käytössä olevien ihmisten vahvuudet saadaan hyödynnettyä ja heikkouksien vaikutukset minimoitua?



# Testaussuunnitelman konkretisointi (2/3)

- Testauksen työsuunnitelman laatiminen
  - Mitä tehtäviä pitää tehdä järjestelmän testaamiseksi?
  - Millaisia riippuvuuksia näiden tehtävien välillä on?
- Varasuunnitelmien laatiminen
  - Mitä varasuunnitelmia todennäköisimmin voidaan tarvita?
  - Voidaanko näiden riskien suhteen tehdä jotain jo etukäteen vaikutuksen vähentämiseksi?
- Ylläpitomuutosten ja uusintatestauksen suunnittelu
  - Kuinka suunnitelmaan saadaan mukaan joustoa suunnitelmien muuttumiseen?
  - Mikä on uusintatestauksen rooli ja kuinka se toteutetaan?





# Testaussuunnitelman konkretisointi (3/3)

- Testauksen työmäärän ja aikataulun arvioiminen
  - Mitä resursseja ja millä ajanjaksolla tarvitaan järjestelmän testaamiseksi?
- Testaussuunnitelman muodon valinta
  - Missä muodossa testaussuunnitelmasta on eniten hyötyä?
- Testaussuunnitelman kirjaaminen
  - Kuinka luodaan ensimmäinen versio testaussuunnitelmasta?
  - Kuinka suunnitelma esitellään johtoryhmälle?



# Suunnitelman päivittäminen

- Testaussuunnitelman katselmointi
  - Kuinka testaussuunnitelma katselmoidaan sisällön oikeellisuuden ja käyttökelpoisuuden näkökulmasta?
- Testaussuunnitelman käyttäminen
  - Kuinka testaussuunnitelmaa käytetään projektin aikana testauksen ohjauksessa ja tilannearvioinnissa?
- Testaussuunnitelman päivittäminen
  - Kuinka testaussuunnitelmaa mukautetaan tilanteiden muuttuessa siten että se on linjassa projektin kanssa?



# Testauksen suunnittelun keskeisimmät periaatteet

( Cem Kaner, A Course in Black Box Software Testing (Professional Version),  
Summer-2002, [www.testingeducation.org](http://www.testingeducation.org) )

- Kaikkea ei voi kiinnittää
- Vaikeita priorisointipäätöksiä pitää tehdä ja monet rajoitteet eivät ole hallinnassasi
- Voit vaikuttaa useisiin rajoitteisiin selittämällä asiaa muille projektin osapuolille – enemmän kuin nimen saaminen suunnitelman alle
- Todellisuus on tärkeämpää kuin kyky osoittaa joku muu syylliseksi
- Tehtävänäsi on hallita projektitason riskejä sisältäen sekä kustannukset, aikataulut, ihmisten välisen dynamiikan että projektin toteuttamat ominaisuudet ja luotettavuuden

# Yhteenveto

- Suunnittelua monilla tasoilla
  - Suunnittelu ei ole sama asia kuin suunnitelmadokumentti
  - Viestintä!
  - Suunnittelun langoista vetämällä koko testaustermistö tulee perässä, mutta vähemmälläkin selviää
- Strategia eri taustoihin peilaten eri asia
  - Johdolle kommunikoinnin haaste johtanut esitettyyn tulkintaan
- Eri ihmiset käyttävät testaustermistöä eri tavoilla
  - Opeteltava kysymään merkityksistä
  - Sanat aina latautuneita henkilökohtaisten kokemusten perusteella määrittelystä riippumatta

# Lähteet

- Agile Alliance website <<http://www.agilealliance.com>>
- Kaner, Bach, Pettichord. 2002. Lessons Learned in Software Testing
- Pol, Van Veenendaal. Software Testing. A Guide to the TMap Approach

# Hyviä testaustiedon lähteitä verkossa

- Stickyminds – Forum for testing
  - <http://www.stickyminds.com/>
- ”Better Software”-lehti, ilmestyi ennen nimellä ”Software Testing and Quality Engineering”
  - <http://www.bettersoftware.com/>
- Cem Kaner’s website
  - <http://www.kaner.com/>
- James Bach’s website
  - <http://www.satisfice.com/>
- Rex Black’s website
  - <http://www.rexblackconsulting.com/>
- Karl Wiegers website
  - <http://www.processimpact.com/>
- Tulevaisuudessa suomenkielistä materiaalia
  - <http://www.testauskirja.com>
- Testauksen sertifiointia
  - <http://www.bcs.org.uk/iseb/>
  - <http://www.istqb.org/>
- Software Testing Hotlist
  - [http://www.io.com/~wazmo/qa/#test\\_tools](http://www.io.com/~wazmo/qa/#test_tools)
- Brian Marick’s Website
  - <http://www.testing.com/>
- Bret Pettichord’s Website
  - <http://www.pettichord.com/>
- TestingEducation Promotion site
  - <http://www.testingeducation.org/>
- Suomalainen testauskerho
  - <http://www.pcuf.fi/sytyke/kerhot/testaus/>
- Suomalainen testaajien keskusteluryhmä
  - <http://groups.yahoo.com/groups/fi-testaus/>



<https://www.linkedin.com/in/maaret/>



@maaretp@mas.to