

Testauksen onni ja haasteet osana tuotekehitystä



Maaret Pyhäjärvi, Senior Quality Engineer, F-Secure Oy

Maaret Pyhäjärvi toimii testauksellisissa tehtävissä F-Securella ja osallistuu tuotekehitys- ja toimitusprojekteihin, joissa toimitetaan tietoturvatuotteita Windows työasema- ja palvelinympäristöihin. Hänen työhönsä kuuluu testausta ja testauksen opastamista projekteissa, sekä yleistä testaustoiminnan ja -käytäntöjen kehittämistä F-Securella.

Testaus on haastavaa ja hauskaa - ainakin kun tuotekehitysporukassa on yhteistyössä tekemisen meininki. Tämä artikkeli on tarina siitä miltä testaus meillä F-Securen Windows työasema- ja palvelintuotteiden tekemisessä näyttää, kun suuntana kehitys on vienyt ketterästä testauksesta ketterään ohjelmistokehitykseen.

Ketterästä testauksesta ketterään kehitykseen

Testausporukka - tai laatuinsinöörit, niinkuin meitä F-Securella kutsutaan - on jo jonkin aikaa omannut varsin kohtuullisen ja keskeisen roolin osana tuotekehitystä. Testaukseen keskittyviä asiantuntijoita on yksi kahta ohjelmistokehittäjää kohden, ja olen kahden F-Securella viettämani vuoden aikana oppinut tuntemaan itseni etuoikeutetuksi saadessani tehdä töitä F-Securen Windows työasema- ja palvelintietoturvatuotteiden parissa osana osaavaa ja oppivaa porukkaa. Osana tuotekehitystä testausta on kehitetty eteenpäin riskeihin ja tiedon arvoon perustuen. Resurssit ovat aina rajalliset, aikataulut haastavat ja valitut käytännöt haetaan toiminnasta ja sen tavoitteista, ei testauksen teoriakirjoista. Testauksessa on mietitty vuosien varrella tarkkaan testausdokumentaation arvoa ja roolia, testaustyön jakamista ohjelmistokehittäjien kanssa sekä läheistä yhteistyötä liiketoiminnan kanssa päätöksenteon osalta. Jo reilu vuosi sitten olin valmis luonnehtimaan F-Securen testaustoimintaa ketteräksi. Toimimme mukavasti yhdessä eri alueiden testaajien kesken, jaoimme töitä ja tavoitteita suhteessa siihen mikä oli kokonaisuudessamme tärkeintä ja neuvottelimme asioista erilaisten sidosryhmien kanssa. Haimme aktiivisesti soveltuvia käytäntöjä ja kehitimme toimintaamme eteenpäin.

Viime loppusyksy toi toimintaamme uutta lisäväriä ja haastetta kahden suuren ohjelmistokehitykseen kohdistuneen muutoksen myötä: vaihdoimme koko ohjelmistokehitysrakenteemme perustumaan ketteriin menetelmiin sekä toimme samanaikaisesti käytäntöön tuotelinjan korvaamaan aiemmat erilliset tuoteprojektit. Ketterään

kehitykseen siirtyminen on pakottanut meidät oppimaan paljon ja muuttamaan toimintatapojamme. Tuotelinja-ajattelu on kuitenkin monella tapaa ollut muutoksena testauksen näkökulmasta vielä haastavampi. Tuotelinja vaihtoi tavoitteen yhden tuotteen tekemisestä kaikkien tuotteiden tekemiseen, ja siinä missä aiemmin haasteenamme oli käydä läpi yksittäinen tuote ja järjestelmä, nyt lisäksi haemme testaustulosten uudelleenkäyttöä ja sopivaa täydentämistä samoihin teknologioihin perustuvien varsin erilaisten tuotteiden toimittamiseen. Kun tämän vielä yhdistää 30-päivän inkrementaalisiin toimituskierroksiin, opittavaa on ollut paljon varsin vauhdikkaan vuoden aikana.

Testauksen organisointi suhteessa ohjelmistokehitykseen

Testaus ei ole itsenäinen toiminne, vaan palveluntarjoaja varsin moneen suuntaan ohjelmistokehityksessä. Testausta tekevät meillä niin erikoistuneet testausrooleissa toimivat laatuinsinöörit kuin ohjelmistokehittäjämmekin, ja vastaavasti testaustyötä ja tavoitteita jaetaan yhtä lailla liiketoiminnan kuin tuen edustajien kanssa. Testaus on investointia laatutietoon, ja organisoinnin keskeisenä tavoitteena on saada oikeat asiat tehtyä oikeaan aikaan oikeiden ihmisten toimesta kokonaisuuden optimoimiseksi. Perusjakona testaajien ja kehittäjien roolituksessa on että siinä missä kehittäjät toimittavat mahdollisimman hyviä osia järjestelmiin, testaajat lähtevät pilkkomaan riskejä järjestelmän ja käyttäjän näkökulmasta.

Ketterän ohjelmistokehityksen myötä olemme siirtyneet pienien tiimien toimintamalliin, jossa testaajat ovat osana ohjelmistokehitystiimiä, joista kukin toimittaa omia muutoksiaan päivittäisen integroinnin kautta osaksi kokonaisjärjestelmää. Viimeisen vuoden aikana olemme kokeilleet niin erillistä järjestelmätestausryhmää osana ketterää ohjelmistokehitystä kuin nyt valittua pienten tiimien polkua. Käytännön kokemukset viittaavat että nyt valitsemamme polku vie oikeaan suuntaan: tiimien kyky nähdä muutokset osana kokonaisuutta ja toimittaa asioita jotka ovat lähempänä asiakkaan kokemaa valmista järjestelmää on

parantunut. Työn jakamista on helpottanut että ei ole ryhmää jolle asioita voisi siirtää, vaan asiat ovat oman porukan listoilla - porukan kokoonpanoa säädellään sopimaan tavoitteisiin siirtämällä osaa testaajista tiimeistä toisiin eri kuukausina.

Testaajien yhteinen suunta ja tavoite

Testaajien liittyminen ja jakaantuminen ohjelmistokehitystiimeihin on otettu varsin hyvin vastaan. Jo ennen ketterää ohjelmistokehitystä toimimme varsin tiiviisti yhdessä ja yhteistyö on entisestään tiivistynyt. Ehdotus erillisen testausporukan perustamisesta saa nykyään jo varsin yleisen vastarinnan aikaan, menettäisimme sisäpiirin tiedon siitä millaisia muutoksia ohjelmistoon on tullut ja tulossa. Testausnäkökulman kärjistyminen liian kehittäjäsuuntautuneeksi on ollut huolenaiheena, joka tosin F-Secure kokee testausporukan kanssa ei ole muodostunut ongelmaksi. Olemme tiimeissä mukana olemisen lisäksi ylläpitäneet käytäntöä testausnäkökulman viikkokohtaamisista, jossa vaihdamme kunkin edustajan keräämää sisäpiiritietoa muutoksista ja riskeistä, ja etsimme mahdollisuutta optimoida toimintaamme yli tiimirajojen.

Yhteinen suunta ja tavoite on parantunut myös testauksen ja hallinnollisen puolen osalta. Testauksen haasteita osana toimitusta ratkotaan yhteistyössä resurssiomistajien kanssa, etsin luovia ratkaisuja niin oman porukan tekemisten ja tavoitteiden kuin ulkoisten toimittajien käyttämisen osalta. Testaus tuottaa hyödyllistä ja käytökelpoista tietoa, ja saa vastineeksi yhä paremmat mahdollisuudet toimia tämän tiedon toimittajana.

Vaatimukset, toimitukset ja suunnittelun tärkeys

Ketterä ohjelmistokehitys on muuttanut suunnittelun käytäntöjä ja roolia. Etenkin kokonaisuuden suunnittelu ja yhtä tiimiä suuremman porukan työn synkronointi on opettanut vuoden aikana paljon ketterän ohjelmistokehityksen luonteesta. Toimintaa ohjaa varsin konkreettisesti kuukausittaiset toimitukset ja sen miettiminen mitä seuraavassa sprintissä toimituksen tavoitteeksi otetaan. Pidemmän aikavälin suunnittelun mekanismit perustuvat tuotebacklogiin ja kunkin kuukauden karkean toimitussuunnitelman tarkasteluun. Tuotebacklog korvaa projektisuunnitelman, ei niinkään vaatimusmäärittelyä. Vaatimusten ymmärtämiseksi perinteiset menetelmät käyttöpausten läpikäymiseksi ovat osoittautuneet edelleen toimiviksi.

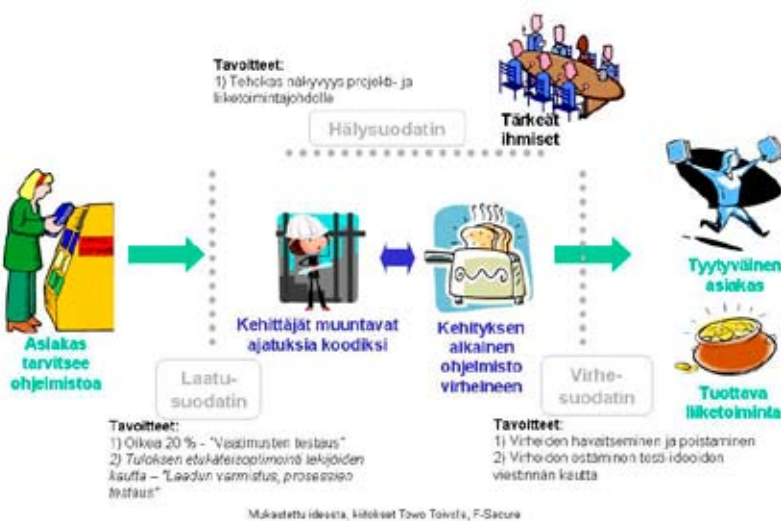
Iso muutos virheraporttien käsittelyssä on ollut muutoshallinnan mekanismin jakaminen tiimeissä toimiviin muutosraateihin (change control board) ja yli tiimirajojen toimivaan muutosneuvostoon (change advisory board). Tiimeissä tiedetään muutoksen yksityiskohdat, kustannukset ja riskit, kun taas ulkopuolelta neuvotaan lähinnä riittävän laadun konkretisoinnissa. Muutosneuvosto auttaa oppimaan konkreettisemmin millaista on hyvä laatu meidän toimitustemme osalta, jotta tiimien muutosraati osaa itseohjautuvasti tehdä oikeansuuntaiset kompromissit. Pienentynyt lisäkustannus on auttanut meitä parantamaan laatua merkittävästi, kun usein parannus tehdään vastaavassa ajassa kuin perinteisin prosessein on saatu neuvoteltua josko muutosta yleensäkin voi tehdä.

Testausdokumentaatio

Testauksen tarkoituksena on tuottaa tarpeellista tietoa, ja dokumentaatiolla on rooli siltä osin kuin se on hyödyllistä. Käytännön kautta on opittu tekemään testitapauksia varsin toisella tapaa kuin testauskirjallisuus opastavaa, käyttäen testitapausmäärittelyjä testaustoiminnan runkona ja muistilistana, mutta ei yksityiskohtien dokumentoijana ja puolustusmekanismina. Ei ole itsestään selvää että saman testin suorittaminen uudelleen olisi parasta ajankäyttöä, joskin tärkeistä näkökulmista pidetään toistuvasti huolta.

Nykyinen sovittu toimintatapamme hyödyntää käytötapauksia testauksen lähtökohtana. Testitapausmäärittelyihin kirjataan käytötapauksia kahdella tapaa: hyötynäkökulmasta (mitä käyttäjä haluaa) ja toteutusnäkökulmasta (mitä toteutamme käyttäjälle). Näiden käytötapauksen ympärillä teemme tutkivaa testausta, rajaten kunkin testin laajuutta budjetoidulla ajalla. Sama tavoite kahden tunnin aikarajalla ja kahden viikon aikarajalla löytää varsin erilaisia asioita, ja sopii erilaisten muutosten riskien analysointiin.

Mitä "testaus" tekee?



Testauksen tilanteen raportointi perustuu yhä suuremmassa määrin porukan näkemysten yhteen vetämiseen ja perinteiset testauksen mittarit toimivat lähinnä muistuttajina, eivät raportoinnin lähtökohtana.

Ketterän kehityksen myötä testaussuunnitelmien asemesta muokataan tuotebacklogia sisältämään toimituskohteita jotka sisältävät kohtuullisen testauksellisen tavoitteen, sekä luodaan suoraan tuotedokumentaatiota loppukäyttäjälle testausraporttien asemesta.

Opittua: mitä jäi käteen?

Toimiminen varsin haastavassa ympäristössä pakottaa oppimaan joka päivä. Yhteistyössä tekeminen haastaa itse kunkin auttamaan kaveria ratkomaan esteitä. Oppiminen vaatii melkoista luovuutta ja jatkuvaa halua etsiä parasta tapaa kulkea kohti tavoitetta - kunhan ensin huolehditaan että tavoite on ymmärretty puolin ja toisin.

Opit yhteistyöstä pienissä tiimeissä

Testaajan ammattitaito ja eheys pannaan koetukselle kehittäjien lähellä

Etenkin uudet testaajat saattavat suostua väärin kompromisseihin tärkeiden näkökulmien osalta kehittäjien lähellä toimiessaan, ja unohtaa muita palveltavia sidosryhmiä. Testaajaverkoston ylläpitäminen tuntuu auttavan säilyttämään oikeita painotuksia ja kehittymään eteenpäin, niin että läheisyyden hyödyt ylittävät mahdolliset haitat.

Kehittäjät osaavat testata

Ohjelmistokehittäjät osaavat testata ja analysoida omien osiensa toimintaa osana järjestelmää, kunhan vain aikaa analysointiin on. Tästä huolimatta työnjakomielessä toimii varsin mainiosti nopeamman aikataulun saavuttamiseksi se että kehittäjät testaavat osan ja erilliset testaajat osan. Ketterien tiimien resursoinnissa olemme joutuneet joinain kuukausina luomaan myös porukoita joissa on vain kehittäjiä, jotka osaavat varsin mainiosti kantaa vastuun niin tekemisen kuin toimituksen laadun osalta. Läheinen yhteistyö on opettanut arvostamaan osaamista puolin ja toisin, ja etenkin automatisoinnin osalta yhteistyö tuottaa mainiota tuloksia tavoitteen ja toteutuksen kohtaamisen kautta.

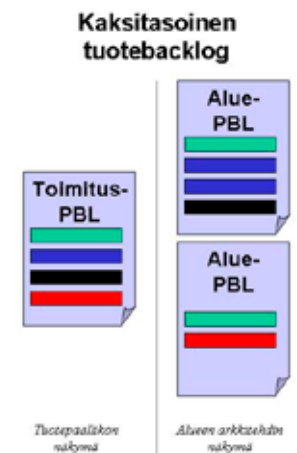
Testaus tarvitsee joustoa myös kuukausitoimitusten sisällä

Kuukausitoimituksen suunnittelu perustuu kiinteisiin työmääriin, joiden käyttö suunnitellaan kuukausi kerrallaan tarkemmin. Tämä voi väärin sovellettuna johtaa tarpeettomaan jäykkyyteen testauksen työmäärien suhteen, kun viikon testauksen jälkeen opitaan että työmäärät toisella tapaa kohdistettuna toisivat paremman lopputuloksen. Testaus on jatkuvaa oppimista riskeistä, ja kukin päivä testausta oleellisesti muuttaa suuntaa

ja tarvetta loppuinvestoinnille. Tämä oppi on ollut erityisen haastava kun testaustyö jakaantuu useille tiimeille. Uusi suunnittelukäytäntö alkuun vähensi testauksemme ketteryyttä, kunnes opimme suunnittelemaan järkevällä yksityiskohtien tasolla.

Testauspäällikön rooli murroksessa

Testauspäällikön toimenkuva ja tarpeellisuus on etsinyt käytännön toteutustaan uudessa toimintamallissa, ja on muuttunut varsin paljon. Aiemman kokonaisuuden testauksellisen koordinoinnin sijasta testaustentantaa osana tuotebacklogien tekemistä, pitäen huolta siitä että valmiuden käsite konkretisoituu yhä paremmin. Suoraa käskeyttä itseohjautuviin tiimeihin ei voi olla, tiimit suunnittelevat testaustyönsä ja testauspäällikön rooliksi jää osoittaa mahdollisia huolenaiheita vastuiden päällekkäisyyden tai puutteen osalta useiden tiimien kuviossa.



Testausyhteistyö useiden tiimien kesken

Päällekkäisyyksien ja aukkojen välttäminen samaa kokonaisuutta muuttavien pienten tiimien kesken vaatii yhteistyötä ja koordinoitua useiden tiimien kesken. Testausnäkökulman synkronointi viikottain ja aktiivinen yritys optimoida kokonaisuutta on tuntunut toimivan, etenkin kun sitä täydentää toisinaan testaussuunnittelun työpajoilla, joissa jaetaan kokonaisuuden vastuita yhteistyössä. Tiedon siirto tiimien välillä oikealla yksityiskohtien tasolla on edelleen suurimpia haasteita, vaikka olemme sen osalta työtä paljon tehneet.

Opit järjestelmätestauksen järjestämisestä

Stabilointisprintit ovat tarpeellisia

Vaikka yritämme huolehtia että työtä ei vuoda ulos kuukausitoimituksista, toimituskokonaisuudet laajuudessaan vaativat enemmän aikaa. Toisinaan on tarpeen suunnitella kokonaisia kuukausia selvitys- ja stabilointitavoitteen ympärille, parantaen laatua myös aiemmista tuotejulkaisuista, sallien pidemmän aikavälin katsoa järjestelmää kokonaisuutena yli tiimirajojen.

Oppimisen aikaväli on suurempi kuin sen työmäärä

Usein asiaan omaksuminen vaatii aikaa kypsytykseen. Ihmiset tarvitsevat usein aikaa hauduteluun ja uusien ideoihin syntymiseen. Oppimisen mahdollistava aikamäärä tuntuu usein olevan suurempi kuin oppimisen työmäärä. Yksilökohtaiset erot ovat merkittäviä, toisilla oppiminen oivallukseen saakka tuntuu olevan nopeampaa. Ketterän kehityksen lyhyet toimitusajat vaativat nopeaksi oppijaksi oppimista.

Valmiuden määrittely tuotebacklogin kautta

Testaussuunnitelmien sijasta huomio on siirtynyt tuotebacklogin toimitussisältöjen testausteliseen näkökulmaan. Isoimpia oivalluksia meille oli tajuta että tuotebacklog ei korvaakaan vaatimusmäärittelyä, vaan projekti- ja testaussuunnitelmat, ja sitä kautta auttaa meitä keskittymään jatkuviin toimituksiin.

Sitoutumisesta ja laadusta raportointi

Eri osapuolten yhteisen ymmärryksen työstäminen siitä mitä tullaan saamaan valmiiksi on edelleen päivittäisen oppimisen kohde. Eri osapuolet ajattelevat eri abstraktiotasoilla, ja kompromissina olemme hakeneet kaksitasoisen tuotebacklogin keskeisimpien näkökulmien mukaan tuomiseksi. Laadun kontrolli on siirretty tiimeihin, muutoksista ja korjauksista päättää tiimien sisäinen muutostyö ulkoisen muutosohjauksen perusteella - ulkopuolelta kerrotaan sääntöjä tärkeydelle ja tuetaan jatkuvaa oppimista siitä mikä laadulle on tärkeää.

Jatkuvan beta-ohjelman organisointi

Ulkoisiin toimituksiin liittyy lisäkustannus, ja meidän oli tarpeen minimoida ulkoisten toimitusten määrää kuukausitoimitussyklin aikaan saamiseksi. Teimme tämän yhdistämällä eri tuotteiden

betaohjelmia yhteen yhdeksi jatkuvaksi betaksi, johon toimitetaan kuukausittain. Isoja oppeja on saatu beta-ohjelmien tavoitteista ja hyödyistä, ja niiden luovasta yhdistelemisestä oikean asiakaspalautteen saavuttamiseksi oikeaan aikaan.

Opit suunnittelun tärkeydestä

Tuotebacklogit ovat yhteistyön tulos

Olemme huomanneet parhaiden tuotebacklogien syntyvän yhteistyössä niiden osapuolien kesken jotka sitä yhteisenä mekanismina käyttävät: luonnollinen yhdistelmä on projektipäällikön, testauspäällikön ja tuotepäällikön yhteistyö, niin että tuoteomistaja (tuotepäällikkö) omistaa priorisointinäkökulman.

Etukäteissuunnittelu sisältyy tuoteomistajan rooliin

Alkuun keskityimme oppimaan kuinka toimitetaan kuukauden osissa, kiinnittäen vähemmän huomiota kokonaisuuden suunnitteluun pidemmällä aikavälillä. Sitten olemme huomanneet roolien muutoksen siltä osin että pidemmän aikavälin suunnittelu tehdään tuotebacklogin kautta osana tuoteomistaja-näkökulmaa, joka meidän rakenteissamme jakaantuu kolmelle roolille: tuotepäällikölle (prioriteetit), program managerille (resurssit) ja projektipäällikölle (viestintä). Etukäteissuunnittelu on keskeistä, jopa suuremmassa määrin ketterässä kehityksessä kuin perinteisessä, joskin varsin toisenlaisin mekanismein.

Laaturaportti

	Julkaisuhaaran koonnit	Kehityshaaran koonnit	Muutos-tahti
KULLEKIN ALUEELLE			
Toiminnallisuusalue 1	1	1	↑
Toiminnallisuusalue 2	3	1	↑
Toiminnallisuusalue 3	3	2	→
Toiminnallisuusalue 4	2	2	↑
Toiminnallisuusalue 5	2	2	↓
Toiminnallisuusalue 6	1	2	↓
JÄRJESTELMÄLLE			
F - toiminnallisuus	2	2	↑
U - käytettävyys	1	1	↓
R - luotettavuus	1	1	↑
P - suorituskyky	0	0	↑
S - tietoturva	1	1	↓
S - tuettavuus	0	0	↓

Tulkinta:

Numerot - testauksen kattavuus; testaajien subjektiivinen arvio ryhmänä

- 0 - emme ole testanneet
- 1 - hiukan katsottu, mutta ammattimainen testaja ei voi väittää testanneensa
- 2 - peruskattavuus, perustilanteet
- 3 - aidosti testattu

Värit - ohjelmiston laatu; testaajien subjektiivinen arvio ryhmänä

- vihreä - toimii, ei isoja ongelmia
- keltainen - jotain ongelmia, mutta toimii
- punainen - isoja ongelmia, ei toimi

Nuolet - muutospäättäjien ohjelmistossa; yhteinen arvio kehittäjien kanssa

- nuoli ylös - paljon muutoksia, mitätöi tietotason, testattava uudelleen
- nuoli sivulle - joitain muutoksia, hallittava testaus tuottaa merkittävää tietoa
- nuoli alas - vakiintuva alue, vain pieniä muutoksia, testaus etenee eikä toista

Järjestelmäajattelu ja arkkitehtuurit ovat tärkeitä

Järjestelmän ja sen arkkitehtuurin ymmärtäminen on perusta jolla tiimimme toimittavat. Sen merkitys ei ole lainkaan vähentynyt ketteryyden myötä, päin vastoin arkkitehtuurien merkitys on kasvanut tuotelinjan myötä. Arkkitehtuurit ovat kriittinen onnistumistekijä kehityksen menetykselle.

Haastetta tulevaan

Jokainen päivä opettaa meille edelleen uutta. Jatkamme samojen pohdintojen kanssa ja haemme päivittäin ratkaisuja, joilla työn sisältöä ja kykyämme toimittaa voidaan parantaa. Tekemistä riittää edelleen: haluaisimme toki yhä enemmän vastaavalla panostuksella, aina hommista ja haasteista nauttien.