

---

# CASE Jydacom: Tuotekehitys testaajaystävälliseksi

*Tositarina siitä, miten testaajien elämää  
helpotettiin ja kaikki voittivat!*

**Veikko Lohi**

Test Manager

TietoEVRY / Jydacom

**JD** JYDACOM



# **Veikko Lohi**

---

## **Test Manager Jydacom**

### **Polkuni tähän hetkeen**

- ❑ Oulu → Helsinki/Espoo → Jyväskylä → Laukaa
- ❑ Carnegie → Digia → Qvantel → Jydacom
- ❑ Prosessien kehitystä, teknistä konsultointia, testausta ja testauksen hallintaa





# Jydacom



## Rakennusalan toiminnanohjausjärjestelmä

- ❑ Vuodesta 1981
- ❑ Asiakkaina noin 400 rakennusalan yritystä ympäri Suomen
- ❑ Tällä hetkellä TietoEVERYn omistama → siirtymässä EG:lle



# Lähtötilanne (1)

---

- ❑ Vuodessa 4 projektia, joissa tuotantoon vietiin isompi satsi uusia toiminnallisuuksia
- ❑ Jokaisen projektin jälkeen lukuisa määrä bugikorjauspäivityksiä tuotantoon
- ❑ 3 testaajaa + n. 15 betatestaajaa liiketoiminta-asiantuntijoista (5-10 kerrallaan)
- ❑ Monta eri työkalua, joissa määrittäviä, kehitystaskeja, testitapauksia, bugeja jne.



# Lähtötilanne (2)

Jokainen uusi versio käsiteltiin projektina, johon tehtiin projektikohtaisesti aina mm. seuraavaa testaajien toimesta (testauksen rinnalla)

- Yleinen testaussuunnitelma
- Etsittiin 5-10 hlöä betatestaajiksi
- Betatestaajille kirjalliset testausohjeet kyseistä testauskierrosta varten
- Pidettiin betatestauksen aloituspalaveri
- Autettiin päivittäin betatestaajia työssään jne.
- + yleiset projektipalaverit

# Lähtötilanne (3)

- Testiympäristöjä päivitettiin käsin ja niiden päivittämiseen meni iso osa työajasta
- Uusien ominaisuuksien tekeminen lopetettiin (teoriassa) 1 kk ennen tuotantoonvientiä  
-> sen jälkeen vain bugikorjauksia (teoriassa)



# Lähtötilanne (4)

Jokaisen tuotantopäivityksen lähestyessä stressitaso nousi ja nousi...

- ❑ Jatkuvasti löytyi ”nurkan takaa” asioita, jotka olivat vielä testaamatta
- ❑ Kehittäjät tekivät jotain ja testaajat testasivat jotain
- ❑ Kokoonnuttiin isolla porukalla aprikoimaan, uskalletaanko viedä tuotantoon
- ❑ Päivityspäivämääriä siirreltiin

# Lähtötilanne (5)

—  
Kun se versio sitten lopulta laitettiin tuotantoon...

- ❑ Hirveä haloo päällä seuraavat pari-kolme viikkoa
- ❑ Asiakaspalvelun puhelin soi kuumana
- ❑ Kiireisimpiä korjauksia vietiin tuotantoon hyvin ohuella testauksella
  - > Ja sitten niitä taas korjailtiin





# Ongelmia

- ❑ Laatuongelmat
- ❑ Uusien ominaisuuksien hidas matka ideasta tuotantoon
- ❑ Muuttuvat tuotantopäivitysaikataulut
- ❑ Jatkuvasti kasvava stressi tuotantoonvientipäivän lähestyessä
- ❑ Ei mahdollisuutta kunnon regressiotestaukselle pikakorjausten yhteydessä (ei automaatiota)



# Toimenpiteitä (1)

## Projektit → Prosessi

- ✓ Ohjataan kehitystiketeillä
- ✓ Aikaa ei tuhlaannu projektien suunnitteluun ja projektipalaveriisiin
- ✓ Koko ajan kaikilla selvänä, mitä tehdä seuraavaksi
- ✓ Toiminnan kehitys jää suoraan pysyvään käyttöön
- ✓ Aikataulut pitää – sisältö joustaa



## Toimenpiteitä (2)

### Siirryttiin yhteen työkaluun (Azure DevOps), jossa

- ✓ Kehitystiketit ja määrittelyt
- ✓ Testauskirjanpito bugeineen
- ✓ Kommunikointi (kysymykset ja vastaukset)
- ✓ Testiautomaation ajaminen ja tulosten seuranta
- ✓ Versionhallinta, buildien kääntäminen
- ✓ Testiympäristöjen päivittäminen
- ✓ Dashboard, josta kokonaistilannetta seurataan



## Toimenpiteitä (3)

### Automatisoitiin regressiotestit sekä testiympäristöjen päivitys

- ✓ Robot Framework + Selenium
- ✓ Ajot käynnistetään ADO:sta
- ✓ Buildien teko ADO:sta
- ✓ Testiympäristöjen päivitys ADO:sta
- ✓ Nämä kaikki ajastettuna päivittäin tai manuaalisesti milloin vain
- ✓ Myös tikettien tilat päivittyvät automaattisesti, kun esim. korjaus on päivittynyt testiympäristöön





# Toimenpiteitä (4)



## Yksikkötestit ja koodikatselmoinnit käyttöön

- ✓ Kehittäjien tekemänä
- ✓ Koodin laatu parani jo ennen kuin se tuli testaajalle
- ✓ Toimivampaa sovellusta on mukavampi testata



## Toimenpiteitä (5)

### Luovuttiin testitapausten kirjoittamisesta

- ✓ Prosessi ohjaa kehitystarpeet pieniksi tiketeiksi → hahmottaminen helpompaa → testaaminenkin helpompaa suoraan määrittelyn pohjalta
- ✓ Tutkivalle testaukselle enemmän aikaa
- ✓ Jokainen muutos testataan tiketti tiketiltä
- ✓ Testitapauksia tehdään edelleen tarvittaessa



## Toimenpiteitä (6)

### — Luovuttiin suurelta osin betatestauksesta

- ✓ Edeltävän prosessin kehityksen myötä betatestaajien löytämät bugit jäivät vähäisiksi
- ✓ Betatestauksen tukemisesta säästyneellä ajalla varsinaiset testaajat ehtivät testata enemmän ja betatestaajat ehtivät keskittyä varsinaisiin asiantuntijahommiinsa
- ✓ Edelleen kohdennettuja asiantuntijatestauksia tarpeen mukaan

# Alkutilanne vs Lopputilanne

## Mihin testaajan aika meni ennen

- Projektipalaverit (ja –dokumentit)
- Määrittysten etsiminen (niitä oli eri paikoissa eri työkaluissa)
- Määrittäisiin perehtyminen
- Testitapausten kirjoittaminen
- Testiympäristöjen päivittäminen
- **Testaaminen** (testitapausten suorittaminen manuaalisesti)
- Testauskirjanpito (projektista tai ominaisuudesta riippuen eri työkaluihin)
- Betatestaajien tukeminen

## Mihin nyt

- Määrittäisiin perehtyminen
- **Testaaminen** (automaatio + tutkiva testaus)
- Testauskirjanpito (samaa yhteen työkaluun, jossa on määrittäyksetkin)





# Vaikutuksia

- Reaaliaikainen kokonaistilanne kaikilla selkeänä
- Uusien ominaisuuksien ja tuotantokorjausten läpimenoaika pieneni huomattavasti
- Samalla porukalla ehditään tehdä enemmän
- Aikataulut alkoivat pitää
- Laatu parani selkeästi
- Tuotantoonvienteihin liittyvät jännitys, stressi ja ylityöt vähenivät
- Fiilismittari (1-5) nousi 3,7:stä 4,3:een



# Mitä tästä opimme?

## Testaajien stressiä voi vähentää!

- ❖ Iso harppaus vanhasta vesiputousmallista nykyaikaisempaan ja ketterämpään ohjelmistokehitykseen
- ❖ Työsarkaa riittää yhä, mutta suunta on selkeästi oikea
- ❖ **Mites teidän organisaatiossa?**
  - Kaikilla tuskin tällainen lähtötilanne, mutta voisiko joku näistä asioista sopia teidänkin ympäristöönne?



# Harkitse voisiko teilläkin... (1)

---

- ❖ Projekteistanne muodostua jatkuva prosessi?
- ❖ Eri toimintojanne siirtää samaan työkaluun?
- ❖ Regressiotestaustanne automatisoida?
- ❖ Testiympäristöjen ylläpitoa automatisoida?
- ❖ Kehittäjät testata&katselmoida enemmän?
  - Yksikkötestit, käyttöliittymätestit, pakotetut koodikatselmoinnit



## Harkitse voisiko teilläkin... (2)

---

- ❖ Tuotantoonvientien aikataulujen siirtelystä luopua?
- ❖ Laittaa Feature freeze ja Code freeze -käytännöt kuntoon?
- ❖ Turhaa dokumentointia vähentää?



---

# Jäikö kysyttävää tai kommentoitavaa?

**Veikko Lohi**

[veikko.lohi@tietoevry.com](mailto:veikko.lohi@tietoevry.com)

 veikkolohi

[www.jydacom.fi](http://www.jydacom.fi)

 JYDACOM

