

# Agile Testing

12/11/2009





iLean

Mijn vraag rond agile testing:

Kan agile testing zonder automated testing? of maw. is er een alternatief indien automated testing geen optie is?

Naam: Joke Gysels Email: joke.gysels@acerta.be

iLean

Mijn vraag rond agile testing:

Indien geen budget of steun voor automated testing -> is het dan haalbaar voor SCRUM a ha?

Naam: NANCY BEUTELS Email: nbe@quasus.com

Geen automated → reguleren testen moeilijk haalbaar

↳ voor functionele elementen die niet aan mekaar hangen.

Wie schrijft testen?

↳ developer?

↳ indien ja. eigen code met testen

↳ tester

↳ Fitness

↳ vorm is door developer gemaakt

↳ testen vult vorm in

↳ output is zichtbaar

↳ aanpakken automated: niet zo moeilijk  
waken je niet vanuit het scherm gaat

XP Days → eind november

↳ demo fitness

Testers geven input voor testen die developers moeten uitvoeren

Interfacie testen niet vergeten, als je <sup>eg</sup> browser gebonden werkt.

## Hoe unit testen integreren in legacy-code?

Applicatie opsplitsen in functionele gehelen.

→ business logica blokkeren

& hieruit parameters selecteren die een verwacht resultaat ~~blokkeren~~ opleveren (bv. a.u.v. eigen geschreven code / FITness)

Dus: Unit testing overlapt hier met functional testing / regressietesten

↳ Komen uit deze testen fouten → aanpakken van code & uitschrijven van een "echte" unit test

Kan je agile testing toepassen in een niet agile project?

→  NEE

↳ Agile zonder unit testing?

- Het kan als de technieken van onderuit mee helpen / ondersteunen
- Als je testable deliverables hebt.

Mijn vraag rond agile testing:

iLean

Hoe inschatten / planning opstellen  
qua testing in een agile context?

Naam: Joke Gysels Email: joke.gysels@acenta.be

Mijn vraag rond agile testing:

iLean

WANNEER KOMT HET OPMAKEN VAN  
TEST PLAN AAN BOED IN HET AGILE  
PROCES ?

Naam: DE BACKERE  
DANY Email: DANNY.DE BACKERE  
@ De Post. BE

M2Q nv  
Business Park King Square  
Veldkant 33A  
B-2550 Kontich

+32 (0)3 451 36 60

+32 (0)3 451 36 69

## Kommunicatie

Unit testing nageschat door ontwikkelaars.

+

Functionele testen voor begin ontwikkeling gespreid met ontwikkelaar  
Na unit testen door ontwikkelaars weer eerst een  
samenspraak tussen ontwikkelaar en tester.

Testteam doet schatting samen met ontwikkelaar

Antwoord op vraag over middelen / Planning gebieden

- Planning ontstaat door communicatie tussen ontwikkelaar en tester
- Schatting via punten geven aan de stories + overeenkomen  
ook als tester zegt of testwerk veel werk nodig is.

Vraag = schatting tester onafhankelijk van ontwikkelaar  
OF

schatting: combinatie schatting ontwikkelaar en tester



Jef Cumps <jef.cumps@gmail.com>

---

## Samenvatting gesprek gisteren

---

Weyts Koen <Koen.Weyts@qframe.be>

13 november 2009 11:47

Aan: Cumps Jef <jef.cumps@ilean.be>

Jef,

De samenvatting van het gesprek in ons groepje van gisteren:

De vragen

1. Hoe overtuig / motiveer je mensen om testen te schrijven voor de code (TDD dus)?
2. Hoe onderhoud ik acceptance testen efficient wanneer functionaliteit wijzigt? Hoe ga je om met functionele en technische wijzigingen zonder veel tijd te verliezen?

Onze antwoorden:

1.
  - a. Waarom TDD werken? Wel, om de kwaliteit van de code te doen stijgen (je denkt na over de code en het design voor je daadwerkelijk begint te coderen. Anderzijds ook om de kwaliteit van de testen op te trekken. Je gaat niet dezelfde foutieve aanpak testen achteraf wanneer de druk groot is om snel iets op te leveren.
  - b. Hoe motiveer je mensen? Moeilijk. Mensen moeten eerst overtuigd zijn dat er iets aan de traditionele aanpak moet veranderen, dat bepaalde zaken beter kunnen. Mensen laten ondervinden wat er gebeurt als vanalles mis gaat is daar een optie. Het is belangrijk dat je de mensen meehebt. Iets opleggen van bovenaf werkt niet. Je kan daarnaast ook een coach gebruiken om de mensen zaken te leren, zowel op technisch als op organisatorisch niveau.
2. Acceptatie en regressietesten hebben snel de neiging te groot te worden en te veel zaken tegelijk af te checken. Opsplitsen is daar een oplossing. Refactoren zodat bepaalde stukken hergebruikt worden is ook nuttig bij het schrijven van testen. Daarnaast is het belangrijk dat de verantwoordelijkheid mbt het onderhouden van de testen bij iedereen ligt. Wie het potje breekt, het potje betaalt.

Mvg,

# Meer over iLean...

[www.iLean.be](http://www.iLean.be)

[www.iLean.be/training](http://www.iLean.be/training)

Voor onze interessante, gratis avondsessies en onze scrum opleidingen.

The logo for iLean, featuring the word "iLean" in a dark blue, handwritten-style font. The letter "i" is red. Above the "n" is a stylized graphic of a person with arms raised, rendered in grey and black.