

Onze passie is **testen falen**

Een samenvatting van de presentatie die Wim ten Tusscher gaf op de Polteq Conferentie 2019.

Wat een rare titel! Klopt hij wel? Wat is falen eigenlijk?

Er zijn verschillende definities: een fout begaan, afgaan, mislukken, allemaal definities met een negatieve lading. Maar ook: het gewilde niet bereiken. Dat is al veel minder negatief en in de wetenschap volledig geaccepteerd; daar gebruikt men een gefaald experiment om het volgende experiment te verbeteren!

Ook in agile werken zit “mogen falen” en verbeteren ingebouwd. Helaas gebruiken maar weinig teams dit mechanisme. Vaak omdat niemand zijn/haar nek durft uit te steken (omdat verbeteren voelt als zeggen dat je het niet goed doet).

Binnen Polteq verbeteren we onze assessment dienst continu om hem aan te passen aan de marktbehoefte.

Voor 2015 waren we gewend aan grote, meerweekse assessments. De opkomst van agile werken daagde ons uit om een agile variant te ontwikkelen die voor één team in één week kan worden uitgevoerd. Ook dat bleek voor sommige klanten nog te lang en/of te duur. Inmiddels weten we, door het te proberen met durf om te falen, dat het ook in twee dagen of zelfs in twee uur kan.

Recent hebben we weer een succesvolle stap genomen. We hebben nu voor het eerst een tweedaags assessment gecombineerd met hulp bij implementatie van de verbeteringen door het geven van een maatwerk Scrum game training waarin de uitkomsten van het assessment als trainingsdoelen zaten verwerkt.

Het gebruiken van anti-patterns maakte de versnelling in het doen van assessments mogelijk. Anti-patterns zijn heuristische voor testverbetering. Een paar voorbeelden van anti-patterns zijn:

Anti-pattern 1: Ontbreken van een teststrategie.

Kenmerken van dit anti-pattern zijn:

- Het testen van de user story wordt niet gerefined
- Testdekking per testniveau (testpiramide) niet besproken
- Alleen functioneel testen, andere kwaliteitscriteria niet getest

Verbetersuggestie: Testen meenemen in de user story refinement. Impact/risico mapping per user story uitvoeren.

Anti-pattern 2: Veel handmatig testen en/of veel UI-testautomatisering (de IJSCO)

Verbetersuggesties: transitie naar de testpiramide door invoering van unit testen, API testen door de developers. Code refactoren voor testability.

Anti-pattern 3: Alleen de tester test, vaak nog handmatig en de piek van testen ligt aan het eind van de sprint

Verbetersuggesties: Bespreek testen tijdens de user story refinement, Benadruk dat binnen agile er een “whole team approach to testing” moet zijn, benadruk de noodzaak tot testautomatisering.

Als je de anti-patterns van agile testen zoekt in je team, ze prioriteert en er vervolgens van sprint tot sprint mee aan de slag gaat in het ritme Plan (product backlog prioritering + planning) – Do (sprint) – Check (retrospective) – Act (retrospective + refinement) kun je stapje voor stapje verbeteren.

Vergeet niet te werken aan de randvoorwaarden:

- Het team bepaalt de “Hoe vraag” dus PO, SM en management bewerken daar waar nodig is
- Stimuleer initiatief, enthousiasme en durf van de teamleden. Vaak zie je dat ze zich verschuilen achter tijdsdruk maar die laat je je als agile team zelf opleggen!
- Werk toe naar een omgeving waar durf en falen wordt geaccepteerd en gewaardeerd.

We moeten af van de angst voor falen! Bekijk falen niet zo negatief. Het is een leerervaring, je weet meer dan als je niets had gedaan, je weet dat deze oplossing niet werkt en daarmee vergroot je de kans dat je de juiste oplossing wel vindt!

Denk bij twijfel aan de wetenschappers die vanuit de intrinsieke drive om dingen te onderzoeken en te ontdekken geen angst hebben om te falen.

Durven falen is DE enabler om verbetering in gang te zetten, om uit de verlamming van de negatieve emoties die horen bij angst te komen en aan de slag te gaan om beter te worden.

Als afsluiter een ezelsbruggetje om de boodschap van dit verhaal te onthouden: wist je dat het Engelse woord FAIL eigenlijk een acroniem is voor First Attempt In Learning?



Wim ten Tusscher