

Hoewel sinds de 80-er jaren van de vorige eeuw veel is veranderd in softwareontwikkeling, is de essentie van testrapportage gelijk gebleven: adviseren over kwaliteit en risico's. Dat betekent echter niet dat het opstellen van een goede testrapportage voor iedereen gesneden koek is. Het komt regelmatig voor dat collega's op zoek zijn naar een goed voorbeeld van of sjabloon voor een testrapport in agile, waterval, devops, outsourced, standaard-pakket of nog een andere context. Ervaring leert dat achter zo'n vraag vaak een andere vraag schuilt: wat en hoe moet ik in mijn situatie rapporteren over testen?

In een aantal blogs ga ik in op die vraag. De basis is gelegd in drie korte stukjes die eerder zijn gepubliceerd:

[De essenties van testrapportage deel 1](#)

[De essenties van testrapportage deel 2](#)

[De essenties van testrapportage deel 3](#)

## **Dit artikel gaat over de behoefte van de projectmanager aan kwantitatieve informatie**

*Hoeveel testgevallen zijn er al uitgevoerd?*

*Hoeveel testgevallen zijn er ok en nok?*

*Hoeveel testgevallen zijn er nodig?*

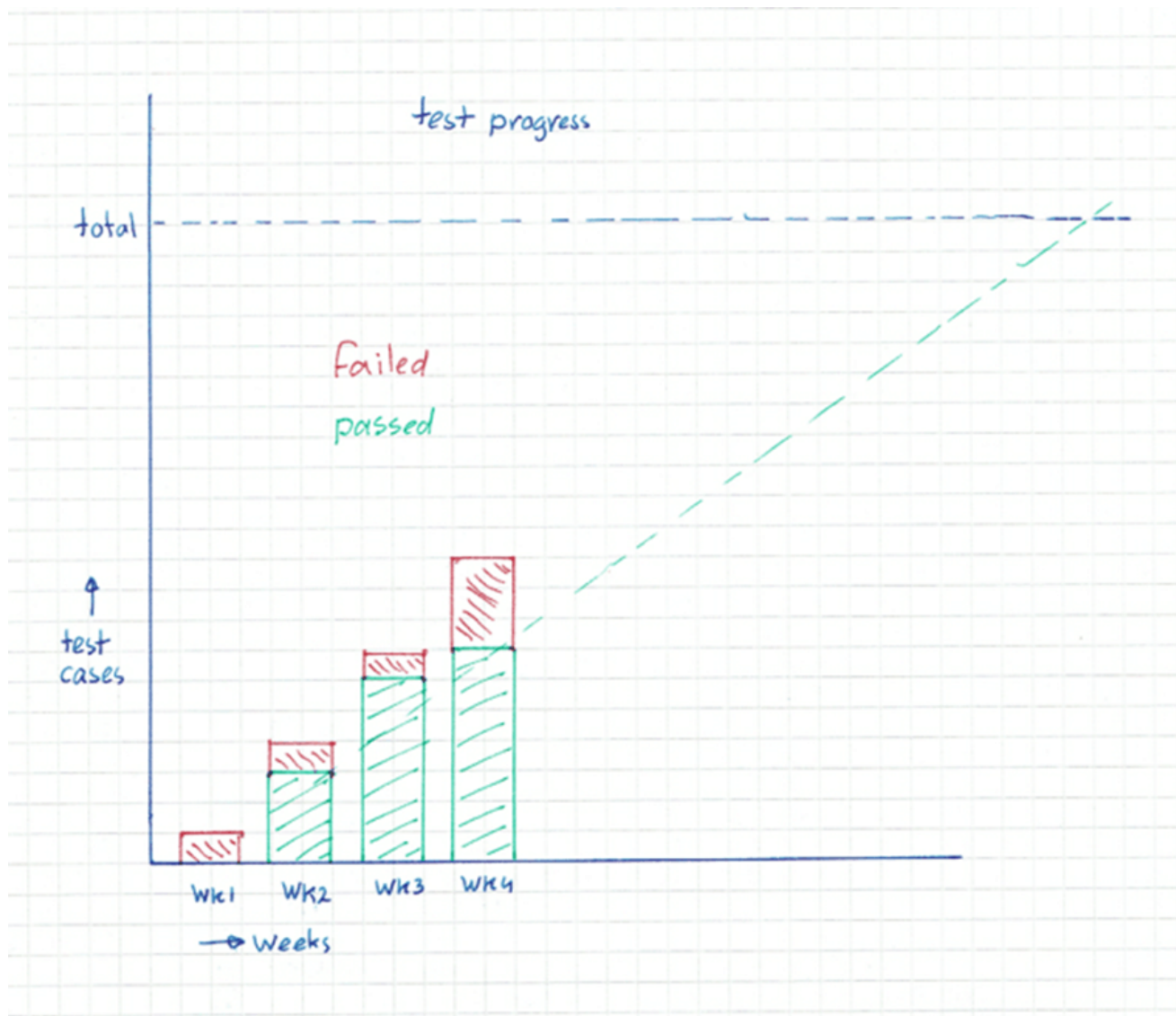
*Wanneer zijn we klaar?*

Zomaar wat vragen die een projectmanager op je af kan vuren. Ik begrijp de behoefte aan antwoorden op die vragen wel. De projectmanager (PM) wil graag een voorspelling van het projecteinde en wil zien hoe hard er aan bepaalde touwtjes getrokken moet worden om de gewenste opleverdatum te halen. Ook als er in sprints wordt gewerkt, heeft de klant vaak een wensdatum voor de oplevering van een *minimum viable product*. De eerste drie blogs over de essenties van testrapportage hebben een kwalitatieve focus. Hoe kunnen we van daaruit de brug slaan naar de traditionele behoefte van de PM aan meer kwantitatieve gegevens die meetbaar zijn en bijdragen aan een hogere voorspelbaarheid van *wanneer het klaar is?*

## **Welke kwantitatieve informatie hebben we voor de**

## PM?

Dat brengt me terug naar mijn allereerste testcoördinatieklus. Met drie collega's moest ik *performance monitoring* testen. We hadden goede specificaties (context: telecom, waarbij je aan gedetailleerd uitgewerkte standaarden moet voldoen) en we konden daar een representatieve set met testgevallen uit afleiden. Zodra we de beschikking hadden over de versie van het systeem die performance monitoring ondersteunde, begonnen we met testen. Ik hield per week bij wat de resultaten waren. Dat deed ik op papier, een ruitjesvel op een tafel in het testlab geplakt. Dat moet er ongeveer uitgezien hebben als hieronder. De grafiek geeft aan welke van de beschreven testgevallen inmiddels succesvol waren (groen) en met welke testgevallen we op een probleem stuiten (rood) waarvoor nog een oplossing moest komen.



De grafiek geeft een projectmanager bepaalde informatie:

- De gefaalde tests laten zien dat er nog problemen zijn: de PM kan meer prioriteit geven aan het oplossen van bevindingen om het aantal resterende problemen te verlagen.
- Als je een lijn trekt door de groene staafjes krijg je een eerste orde voorspelling van het moment dat alles oké is. Door dat punt week bij week in de gaten te houden kan de PM beoordelen of aanvullende maatregelen nodig zijn.

Een goede PM snapt uiteraard dat gegevens uit het verleden geen garanties bieden voor de toekomst. De kwaliteit van het product in gebieden die we nog niet hebben bekeken, heeft bijvoorbeeld een grote invloed op de toekomstige testvoortgang. Aan PM's die deze bijsluiter niet goed (willen) begrijpen, heb je als testcoördinator soms je handen vol... En er is meer slecht nieuws, want testgevallen zijn helaas vaak onbetrouwbare makers.

## Waarom tellen van testgevallen vaak geen houvast biedt

*De specificaties zijn onvoldoende representatief.* In mijn historische voorbeeld waren de specs en de afgeleide testgevallen representatief. Die luxe hebben we niet zo vaak in projecten. De PM kan er dus niet vanuit gaan dat het project klaar is als alle tests op basis van de beperkte specificaties *passed* zijn.

*Verschillen in de omvang van tests heeft invloed op het extrapoleren van de testvoortgang naar een projecteinde.* Wat is een test? Eén controle die tien minuten duurt of het doorlopen van een scenario met misschien wel tien checks waar je een halve dag mee zoekt bent? Als een set van tests grote verschillen vertoont tussen de tijd die het kost om tests uit te voeren, dan is de onzekerheidsmarge om de verspelde einddatum groot en biedt het de PM minder houvast.

*Het telbaar maken van tests is geen prioriteit voor Agile teams.* Een agile team richt zich op het opleveren van werkende software, waarbij wel veel wordt getest, maar niet per se met achterlating van gedocumenteerde tests. Testen laat zich niet als geïsoleerde activiteit monitoren. Daarnaast is het team niet bezig met het bepalen van concrete tests voor alle volgende sprints. Het voorspellen van het aantal tests dat nodig is tot het *minimum viable product* is opgeleverd, is daarmee een hachelijke zaak.

*Exploratory testen* - We laten ons bij het testen leiden door allerlei informatie, die deels pas beschikbaar komt tijdens het testen zelf. We laten ons dus niet alleen leiden of beperken door vooraf opgestelde testgevallen. Omdat deze manier van testen, gelukkig, in de meeste projecten plaatsvindt, beschikt de PM ook daarom niet over aantallen tests als basis voor het voorspellen van **het projecteinde**.

## Acceptatiecriteria bieden houvast

Projecten die geen acceptatiecriteria hebben, missen een belangrijk kijkvenster op het project. 'Ergens' zou men een lijst moeten hebben met criteria op basis waarvan de klant of businessvertegenwoordiger bepaalt of er een go gegeven kan worden. De scenario's die zijn opgesteld in een *business driven development* proces (BDD), bijvoorbeeld, zijn de acceptatiecriteria. User stories kunnen ook als kapstok dienen voor acceptatiecriteria, of de business processen die ondersteund moeten worden; liefst alle drie met elkaar gekoppeld. Vervulde acceptatiecriteria zijn *eenheden van vertrouwen in het product* en de voortgang daarin geeft een goed beeld van wat er af is en wat er nog niet af is.

Als we nu *test cases* in mijn voorbeeldplaatje vervangen door *BDD-scenario's* of *acceptatiecriteria* dan kunnen we voor de PM best een interessant grafiekje maken. Groen

staat voor acceptatiecriterium gehaald en bij rood is er nog een probleem op te lossen. We geven daarmee niet per se antwoorden op de vragen aan het begin van dit artikel, maar we dragen wel bij aan de beantwoording van de vraag achter de vragen van de PM: zijn aanvullende maatregelen nodig om project- en productrisico's te beheersen?

## Kwaliteit is de essentie

Dat de blogs over de *essenties van testrapportage* geen aandacht schenken aan kwantitatieve gegevens uit het testproces is geen manco. Projectmanagers begrijpen namelijk steeds beter dat de kwaliteit van een softwareproduct niet is uit te drukken in getallen. Niet voor niets gaat het in de *product story* helemaal niet over testen maar over kwaliteit. De *testing story* en de *story about the quality of the testing* onderbouwen de *product story*. Kwaliteit is de essentie!

**Kees Blokland**