

THIADENS



De kleine komedie

Uw columnist is meer van het type organiseren voor, dan kijken naar. Uw columnist is een man van de grote lijnen. Hij maakt ex ante liever de randvoorwaarden zo, dat hij kwaliteit krijgt, dan dat hij achteraf moet controleren of er kwaliteit geleverd is. En dat betekent in ons vak het juiste vakmanschap bij elkaar zetten, hun voorzien van goed gereedschap en je organisatie opbouwen uit kleine teams. Zo waren bij mij immer kleine teams van medewerkers verantwoordelijk voor hun producten en diensten. Of dat nu ontwikkeling of beheer van programmatuur betrof. Deze teams deden hun werk van begin tot eind. Zelf definiëren, zelf ontwikkelen en zelf opleveren voor de gebruikerstest.

Een organisatie kan ook andere accenten leggen. Bijvoorbeeld door ex post de producten van de ontwikkeling en het beheer langs te lopen. Ik noemde dat altijd de kleine kwaliteit. De fouten zijn dan al gemaakt. Je spant het paard achter de wagen. En vaak heeft men enig theatertalent nodig om de maker van een product van zijn fouten te overtuigen. De reviewer moet over het talent van de kleine komedie beschikken, zal ik maar zeggen. Ook deze kleine kwaliteit heeft een kern. Maken van programmatuur kent een achttal principes. Op een congres kreeg ik ze onlangs even opgesomd. Het zijn de volgende acht:

1. *Klaar voor productie*: productieprogrammatuur heeft geen code die niet af is.
2. *Consistentie in de transactieverwerking*: gegevens moeten op een consistente wijze worden verwerkt en opgeslagen.
3. *Foutafhandeling*: als een programma onverwacht crasht, moet je op consistente wijze met de foutafhandeling omspringen.
4. *Defensief programmeren*: op een veilige manier omgaan met onverwachte gebeurtenissen.
5. *Gestructureerd programmeren*: programma's dienen een eenvoudige en heldere structuur te hebben.
6. *Consistent programmeren*: gelijke functies moeten er hetzelfde uitzien.
7. *Consistente lay-out*: de lay-out van een programma moet zijn structuur weergeven.
8. *Isoleren van parameters*: gegevens en instructies dient men gescheiden te houden.

En hoe passen we deze acht principes dan toe? Dat gebeurt in peer reviews en deze werken als volgt. Eerst

print de reviewer de code. Dan loopt hij deze code regel voor regel door. Als hierbij een regel in strijd is met één van de acht principes, omcirkelt hij hem. Elke pagina met één cirkel krijgt een gele sticker. Vervolgens loopt hij de code door op vergelijkbare functionaliteit, die hij als veilig beoordeelt. Ook deze omcirkelt hij. En hij voorziet deze pagina van een groene sticker. Zo loopt hij elke pagina door. Doorlopen van een pagina kost drie minuten. De resultaten worden doorgesproken met de man of vrouw, die het laatst het programma veranderde. En wat leveren deze reviews op? IBM geeft aan dat elk uur inspecteren van programmatuur twintig uur testen bespaart. Bij Imperial Chemical blijken de kosten van het onderhouden van programmatuur zonder zo'n inspectie, tienmaal hoger te liggen dan de kosten van programmatuur die wél een inspectie doorliep en daarna werd aangepast.

Ook in ons land blijken deze acht principes en het hanteren daarvan een probaat middel te zijn om de kwaliteit van programmatuur te onderkennen. Dit jaar is deze methode gebruikt in de jaarlijkse programmeerwedstrijd, waar teams om de eer strijden in zo kort mogelijke tijd een applicatie met een volwaardige functionaliteit te bouwen. Ook dit jaar won weer het team dat zijn applicatie baseert op een omgeving met frames, uitgaande van de Java-taal. Zou men de criteria om te winnen echter enigszins modificeren en kijken we naar de opgeleverde code volgens de genoemde acht eisen, dan was de uitslag misschien anders geweest. Wat is de les voor ons servicemanagers? Kwalitatief goede programmatuur ontstaat door de juiste condities te scheppen. Fine-tuning van kwalitatief goede programmatuur maakt deze programmatuur nóg beter te beheren. En dat laatste merken wij als servicemanagers in de doorlooptijden van wijzigingen. Zo'n kleine komedie is zo gek nog niet!

Bron

Graham Bolton, *the Peer Review experts*, Institute for Software Quality, www.ifsq.nl, Huis ter Heide, 2008.

DR. TH.J.G. THIADENS

is schrijver en docent.