

Boom

AAN DE SLAG MET SOFTWARE TESTEN

PRINCIPES, PROCESSEN
EN TECHNIEKEN

HOSSEIN CHAMANI
GERARD KRUIJFF
GER OOSTING
JOS VAN ROOYEN

3^E DRUK



Aan de slag met software testen

Aan de slag met software testen

Principes, processen en technieken

**Hossein Chamani
Gerard Kruijff
Ger Oosting
Jos van Rooyen**

Derde druk

Boom

**inclusief
website!**

Met onderstaande unieke activeringscode krijg je via www.boomstudent.nl toegang tot de website met extra materiaal. Deze code is persoonsgebonden en gekoppeld aan de derde druk. Na activering van de code is de website twee jaar toegankelijk. De code kan tot zes maanden na het verschijnen van een volgende druk geactiveerd worden. De code is eenmalig te gebruiken.

Omslagontwerp: Coco Bookmedia, Amersfoort

Basisontwerp omslag: Dog & Pony, Amsterdam

Beeld omslag: Shulz, iStock

Opmaak binnenwerk: Nu-nique grafische vormgeving, Goor

© Chamani, Kruijff, Oosting, van Rooyen & Boom uitgevers Amsterdam, 2022

Behoudens de in of krachtens de Auteurswet gestelde uitzonderingen mag niets uit deze uitgave worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand, of openbaar gemaakt, in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnamen of enige andere manier, zonder voorafgaande schriftelijke toestemming van de uitgever.

Voor het overnemen van (een) gedeelte(n) uit deze uitgave in bijvoorbeeld een (digitale) leeromgeving of een reader in het onderwijs (op grond van artikel 16, Auteurswet 1912) kan men zich wenden tot Stichting Uitgeversorganisatie voor Onderwijslicenties, Postbus 3060, 2130 KB Hoofddorp, www.stichting-uvo.nl.

No part of this book may be reproduced in any form, by print, photoprint, microfilm or any other means without written permission from the publisher.

ISBN 9789024445592

ISBN e-book 9789024445608

NUR 173

www.boomstudent.nl

www.boomhogeronderwijs.nl

Inhoudsopgave

| | |
|--|-----------|
| Woord vooraf | 9 |
| Introductie | 11 |
| Deel 1 Principes en processen | 15 |
| 1 Het waarom, wat en hoe van kwaliteit | 17 |
| 1.1 Startoefening 1.1 | 17 |
| 1.2 Inleiding | 18 |
| 1.3 Wat is kwaliteit? | 19 |
| 1.4 Startoefening 1.2 | 23 |
| 1.5 Wat is het belang van kwaliteit? | 24 |
| 1.6 Startoefening 1.3 | 25 |
| 1.7 Hoe bepaal je de kwaliteit? | 27 |
| 1.8 Leeropdrachten | 31 |
| 1.9 Literatuur | 35 |
| 2 Testen tijdens het ontwikkelproces | 39 |
| 2.1 Startoefening 2.1 | 39 |
| 2.2 Methoden | 41 |
| 2.3 De watervalmethode | 44 |
| 2.4 Leeropdrachten watervalmethode | 48 |
| 2.5 Startoefening 2.2 | 50 |
| 2.6 Agile | 52 |
| 2.7 Leeropdrachten agile | 56 |
| 2.8 Additionele opdrachten | 57 |
| 2.9 Literatuur | 59 |
| 3 Risicoanalyse en de impact op de test | 61 |
| 3.1 Startoefening 3.1 | 61 |
| 3.2 Risico's en risico'soorten | 64 |
| 3.3 Productrisicoanalyse | 64 |
| 3.4 Risicoanalyse op het project | 68 |
| 3.5 Bedrijfsprocesrisico | 70 |
| 3.6 Leeropdrachten | 70 |
| 3.7 Literatuur | 74 |

| | | |
|----------|---|------------|
| 4 | Toetstechnieken in de praktijk | 77 |
| 4.1 | Startoefening 4.1 | 77 |
| 4.2 | Wat is toetsen? | 78 |
| 4.3 | Leeropdracht 1 | 80 |
| 4.4 | Toetstechnieken | 81 |
| 4.5 | Informele review | 81 |
| 4.6 | Structured walkthrough (gestructureerd doorlopen) | 82 |
| 4.7 | Inhoudelijke review | 83 |
| 4.8 | Inspectie | 83 |
| 4.9 | Leeropdracht 2 | 84 |
| 4.10 | Literatuur | 85 |
| 5 | Usability als kwaliteitskenmerk van software | 87 |
| 5.1 | Startoefening 5.1 | 87 |
| 5.2 | Wat is usability? | 91 |
| 5.3 | Het belang van usability | 96 |
| 5.4 | Leeropdrachten | 99 |
| 5.5 | Literatuur | 103 |
| | Deel 2 Testontwerptechnieken | 105 |
| 6 | Nut en noodzaak van testontwerptechnieken | 107 |
| 6.1 | Startoefening 6.1 | 107 |
| 6.2 | Waarom is het gebruik van testontwerptechnieken belangrijk? | 111 |
| 6.3 | Diverse soorten testontwerptechnieken | 113 |
| 6.4 | Toepassing van testontwerptechnieken en het type bevinding | 115 |
| 6.5 | Testontwerptechnieken in combinatie met dekkingsgraad | 115 |
| 6.6 | De keuze van een testontwerptechniek | 121 |
| 6.7 | Leeropdrachten | 122 |
| 6.8 | Literatuur | 124 |
| 7 | Equivalentieklassen en grenswaardenanalyse | 127 |
| 7.1 | Startoefening 7.1 | 127 |
| 7.2 | Equivalentieklassen | 128 |
| 7.3 | Startoefening 7.2 | 131 |
| 7.4 | Grenswaardenanalyse | 132 |
| 7.5 | Belangrijkste overeenkomsten en verschillen op een rij | 135 |
| 7.6 | Leeropdrachten | 135 |
| 7.7 | Literatuur | 137 |

| | | |
|-----------|--|------------|
| 8 | Use case test | 139 |
| | 8.1 Startoefening 8.1 | 139 |
| | 8.2 Use case test | 141 |
| | 8.3 Meer over use cases en testgevallen | 143 |
| | 8.4 Leeropdrachten | 145 |
| | 8.5 Literatuur | 147 |
| 9 | Procescyclustest | 149 |
| | 9.1 Startoefening 9.1 | 149 |
| | 9.2 Procescyclustest | 149 |
| | 9.3 Leeropdrachten | 158 |
| | 9.4 Literatuur | 160 |
| 10 | Decision table testing | 163 |
| | 10.1 Startoefening 10.1 | 163 |
| | 10.2 Het wat, waarom en hoe van decision table testing | 165 |
| | 10.3 Werkwijze | 166 |
| | 10.4 Meer over het vereenvoudigen van decision tables | 169 |
| | 10.5 Leeropdrachten | 171 |
| | 10.6 Literatuur | 172 |
| 11 | State transition testing | 175 |
| | 11.1 Startoefening 11.1 | 175 |
| | 11.2 State transtion testing | 177 |
| | 11.3 Leeropdrachten | 184 |
| | 11.4 Literatuur | 185 |
| 12 | Exploratory testing | 187 |
| | 12.1 Startoefening 12.1 | 187 |
| | 12.2 Het wat, waarom en hoe van exploratory testing | 188 |
| | 12.3 Leeropdrachten | 191 |
| | 12.4 Literatuur | 193 |
| 13 | Testautomatisering | 195 |
| | 13.1 Startoefening 13.1 | 195 |
| | 13.2 Wat is testautomatisering? | 196 |
| | 13.3 Redenen voor het toepassen van testautomatisering | 199 |
| | 13.4 Toepassingsgebieden van testautomatisering | 201 |
| | 13.5 De aanpak van testautomatisering | 205 |
| | 13.6 Het toepassen van testtools | 207 |
| | 13.7 Het uitvoeren van een unittest | 207 |
| | 13.8 Literatuur | 208 |

| | | |
|-----------|---|------------|
| 14 | Uitwerkingen van startoefeningen | 211 |
| 14.1 | Hoofdstuk 1 | 211 |
| 14.2 | Hoofdstuk 2 | 211 |
| 14.3 | Hoofdstuk 3 | 212 |
| 14.4 | Hoofdstuk 4 | 213 |
| 14.5 | Hoofdstuk 5 | 214 |
| 14.6 | Hoofdstuk 6 | 216 |
| 14.7 | Hoofdstuk 7 | 217 |
| 14.8 | Hoofdstuk 8 | 218 |
| 14.9 | Hoofdstuk 9 | 219 |
| 14.10 | Hoofdstuk 10 | 219 |
| 14.11 | Hoofdstuk 11 | 220 |
| 14.12 | Hoofdstuk 12 | 221 |
| 14.13 | Hoofdstuk 13 | 221 |
| | De auteurs | 223 |
| | Bijlage A: Checklists voor softwarekwaliteit | 225 |
| | Index | 237 |
| | Illustratieverantwoording | 240 |

Woord vooraf

De rol die software speelt in het leven van miljoenen mensen, is nauwelijks te overschatten. En sinds de covidpandemie is de snelheid waarmee de samenleving gedigitaliseerd wordt alleen maar toegenomen. In alle sectoren, van de zorg tot veiligheid en van onderwijs tot logistiek is software onmisbaar geworden voor het dagelijks functioneren. Als gebruikers vertrouwen we er op dat de software gewoon zijn werk doet, robuust en veilig is. Om aan deze eisen te voldoen, worden hoge eisen gesteld aan professionele software developers. Zij zullen kwaliteitsbewustzijn moeten ontwikkelen in hun werk en hebben zowel theoretische als praktische kaders nodig om hun vaardigheden te ontwikkelen.

De nieuwe editie van *Aan de slag met software testen* biedt software developers een heldere introductie in softwarekwaliteit en testen. Van processen en principes tot testontwerptechnieken om direct praktisch mee aan de slag te gaan. Er wordt veel aandacht besteed aan agile development en automated testen en sluit zo nauw aan op de praktijk in steeds meer bedrijven. Het collectief van auteurs met achtergronden in software educatie en bedrijfsleven zorgt voor een goede aansluiting tussen beide werelden.

In de HBO-i domeinbeschrijving IT-bacheloronderwijs dat alle hogescholen gebruiken om hun eindtermen te definiëren, is kwaliteit en testen als integraal onderdeel opgenomen in de levenscyclus van softwaresystemen. *Aan de slag met software testen* biedt IT-studenten een handzame en actuele introductie om aankomende software developers hun kwaliteitsbewustzijn te laten ontwikkelen en in de praktijk mee te oefenen, of dit nu in integraal projectonderwijs is of in de beroepspraktijk tijdens stages.

Voorjaar, 2022

Ing. Sandra van Steenvelt MA EMITA

Voorzitter Stichting HBO-i

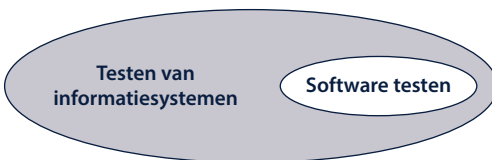
Introductie

Aan de slag met software testen is geschreven voor drie doelgroepen die zich willen verdiepen in het vak testen:

- Hbo-studenten die een ICT-opleiding volgen zijn de primaire doelgroep. Zij kunnen de verworven kennis direct toepassen in het projectonderwijs of in projecten bij stage- en afstudeerbedrijven. Op deze manier worden ze voorbereid op hun eerste baan.
- ICT'ers zoals requirements engineers, architecten, ontwikkelaars en applicatiebeheerders zijn de secundaire doelgroepen. Zij kunnen met dit boek de kwaliteit van het eigen werk verhogen of de eigen inzetbaarheid vergroten.
- Acceptanten van softwareproducten of vertegenwoordigers van gebruikers behoren ook tot secundaire doelgroepen. Zij kunnen samen met ontwikkelteams de kwaliteit van software vroegtijdig testen.

Reikwijdte van het boek

Software testen is een van de focusgebieden van het vakgebied testen van informatiesystemen. Zie figuur 1. Een informatiesysteem in brede zin is het geheel van mensen, methoden en middelen waarmee systematisch gegevens verzameld, vastgelegd en verwerkt worden. Het informatiesysteem in enge zin is een combinatie van hardware, software en infrastructurele componenten voor een breed spectrum aan toepassingen of organisatie specifieke toepassingen.



Figuur 1 Software testen als een van de focusgebieden binnen testen van informatiesystemen

Testen van informatiesystemen is omvangrijker dan software testen. Denk bijvoorbeeld aan het controleren van:

- De juistheid van de informatie en gegevens op een webwinkel. Zijn de productcodes en de prijzen juist?
- De beveiliging van de informatie en gegevens. Hebben slechts de bevoegde personen toegang tot de informatie en gegevens van de klanten? Gebruiken deze personen de informatie en gegevens waarvoor deze daadwerkelijk bedoeld zijn?

- De integratie van een webwinkel binnen de bedrijfsprocessen. Zijn de handmatige en geautomatiseerde processen van de webwinkel op elkaar afgestemd?
- De koppelbaarheid van de webwinkel met de systemen van derden zoals banken, transportbedrijven en leveranciers. Ontvangt de webwinkel actuele informatie van de transportbedrijven over de status van de levering?

Tabel 1 toont een overzicht van de genummerde hoofdstukken.

Tabel 1 Genummerde hoofdstukken

| Deel 1 Principes en processen | Deel 2 Testontwerpstechnieken |
|---|---|
| 1. Het waarom, wat en hoe van kwaliteit | 7. Equivalentieklassen en grenswaardenanalyse |
| 2. Testen tijdens het ontwikkelproces | 8. Use case test |
| 3. Risicoanalyse en de impact op de test | 9. Procescyclustest |
| 4. Toetstechnieken in de praktijk | 10. Decision table testing |
| 5. Usability als kwaliteitskenmerk van software | 11. State transition testing |
| 6. Nut en noodzaak van testontwerpstechnieken | 12. Exploratory testing |
| | 13. Testautomatisering |
| | 14. Uitwerkingen van startoefeningen |

Het boek beperkt zich tot onderwerpen die de studenten direct kunnen oefenen of toepassen. Op deze manier kunnen ze ervaring opdoen met het vak software testen. Hierdoor ontwikkelen de studenten een solide kennisbasis om zich verder te verdiepen in het vak.

Studiewijzer

Figuur 2 laat de structuur zien van de hoofdstukken. Elk hoofdstuk bestaat uit vier vaste onderdelen:

1. leerdoelen
2. startoefening
3. theorie
4. leeropdrachten

| | | | |
|-----------------|--------------------|--------------|---------------------|
| 1 Leerdoelen | 2 Startoefening | 3 Theorie | 4 Leeropdrachten |
|-----------------|--------------------|--------------|---------------------|

Figuur 2 De structuur van de genummerde hoofdstukken

Wanneer je een nieuw hoofdstuk wilt bestuderen, bekijk dan eerst de leerdoelen. De leerdoelen vormen het uitgangspunt voor alles wat je in het hoofdstuk gaat leren. Maak daarna altijd de startoefening. Deze oefening geeft je inzicht om de theorie sneller en beter te kunnen begrijpen. Bestudeer vervolgens de theorie. Voer daarna alle leeropdrachten uit. Deze leeropdrachten zijn gekoppeld aan de leerdoelen. De startoefening en de leeropdrachten zijn uitermate belangrijk voor het verhogen van het leerrendement en voor het behalen van ‘een tien voor testen’. Via www.boomstudent.nl vind je extra trainingsmateriaal, zoals meerkeuzevragen, testopdrachten en uitwerkingen van de leeropdrachten. Verder zijn er diagnostische toetsen en hun uitwerkingen opgenomen.



Een investering voor minimaal vijf jaar

Aan de slag met software testen kan worden gebruikt vanaf het eerste jaar van een vierjarige ICT-opleiding in het hbo. Daarom hebben de auteurs ervoor gekozen om onderwerpen te behandelen die langer dan vijf jaar relevant zijn voor de opleidingen en voor de potentiële werkgevers van de studenten. Op deze manier is het boek voor zowel de studenten als voor de opleidingen een investering voor de lange termijn.

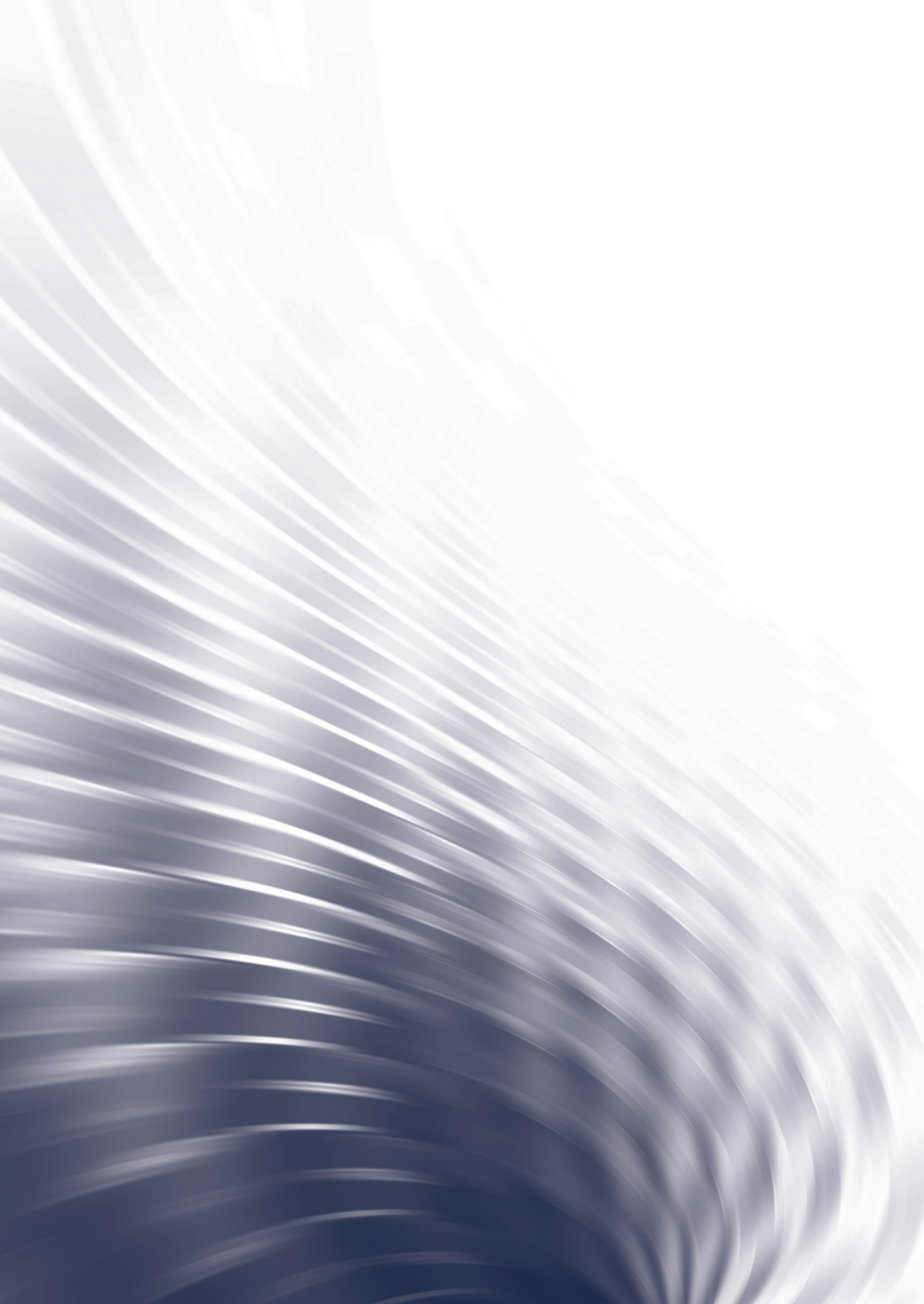
Dankwoord

Wij zijn als schrijvers collectief dank verschuldigd aan allen die op enigerlei wijze hun bijdrage hebben geleverd aan de totstandkoming van dit boek. Van meet af aan werden we ondersteund en gefaciliteerd door onze werkgevers. Hun bijdrage was onmisbaar.

Deel 1

Principes en processen





Het waarom, wat en hoe van kwaliteit

1

In dit hoofdstuk verwerf je kennis en vaardigheden over het begrip kwaliteit. De onderstaande tabel toont de leerdoelen.

| Leerdoelen |
|--|
| 1. Je kunt binnen een bepaalde context beschrijven wat kwaliteit is. |
| 2. Je kunt binnen een bepaalde context het belang van kwaliteit beschrijven. |
| 3. Je kunt binnen een bepaalde context de kwaliteit vaststellen. |

1.1 Startoefening 1.1



Volgens doctor Joseph Juran betekent kwaliteit: fitness for use (geschiktheid voor gebruik).

Er zijn diverse formele definities van kwaliteit. Fitness for use is waarschijnlijk de eenvoudigste.

Je kunt de kwaliteit van een softwareproduct testen, mits:

- je weet welke behoeften het product moet vervullen;
- je weet aan welke eisen het product moet voldoen.

Deze behoeften en eisen worden in de ICT requirements genoemd.

Opdracht

Lees de onderstaande negen requirements van het cijfersysteem van de hogeschool en geef aan:

- welke requirements de behoeften van de gebruikers beschrijven (user requirements);
- welke requirements de eisen aan het systeem beschrijven (system requirements).

Requirements cijfersysteem hogeschool

1. De docent wil de cijfers van de studenten invoeren, lezen, wijzigen en verwijderen.
2. Het cijfersysteem moet de docent de mogelijkheid bieden om de cijfers van de studenten te kunnen invoeren, lezen, wijzigen en verwijderen.
3. De student wil de eigen cijfers weten.

4. Het cijfersysteem moet de cijfers van de geautoriseerde student aan de student weergeven.
5. Het cijfersysteem moet voorkomen dat de student de eigen cijfers invoert of wijzigt.
6. Het cijfersysteem moet een rood uitroepteken achter een cijfer plaatsen als het cijfer gewijzigd is.
7. De leden van de examencommissie willen weten wie een cijfer heeft ingevoerd, gewijzigd of verwijderd.
8. Het cijfersysteem moet aan alle geautoriseerde gebruikers tonen wie een cijfer heeft ingevoerd, gewijzigd of verwijderd.
9. Het cijfersysteem moet de cijfers binnen maximaal drie seconden weergeven nadat de geautoriseerde gebruiker deze heeft opgevraagd. (*De responstijd moet kleiner zijn dan drie seconden.*)

1.2 Inleiding

Kwaliteit is een breed begrip en kan op alles van toepassing zijn. De kwaliteit van onderwijs, de kwaliteit van de maaltijd enzovoort. Zo breed gaan we niet in dit hoofdstuk. De focus van dit hoofdstuk ligt op software testen. Tegenwoordig zit bijna in ieder systeem of product software waarmee het systeem of product wordt aangestuurd. Denk daarbij aan bijvoorbeeld parkeerautomaten, espressomachines, tv's, navigatiesystemen, dijkbewaking en apparatuur op de intensive care van een ziekenhuis. Vaak blijkt dat de software de zwakke schakel is als het gaat om de correcte werking van de systemen. Dat wordt mede veroorzaakt doordat in de software en in de gekoppelde informatiesystemen continu wijzigingen worden doorgevoerd, in tegenstelling tot bijvoorbeeld hardwarecomponenten.

De impact van software die kwalitatief onder de maat is, kan groot zijn. Dit is afhankelijk van het type systeem. Bij medische systemen kan het leven van een patiënt in gevaar komen. Bedrijven kunnen failliet gaan als de website verkeerde prijzen berekent. Echter, alle systemen moeten aan bepaalde kwaliteitsnormen voldoen. Kwaliteitsnormen worden vaak uitgedrukt in termen als kwaliteitseigenschappen, kwaliteitskenmerken of kwaliteitsattributen. Met deze termen wordt hetzelfde bedoeld. Een bepaald aspect van een informatiesysteem wordt hiermee uitgewerkt. In het boek kunnen beide termen gebruikt worden. In dit hoofdstuk gaan we verder in op het begrip kwaliteit en de diverse methodieken die gebruikt kunnen worden voor het vaststellen van de kwaliteit van een informatiesysteem.

1.3 Wat is kwaliteit?

Wat onder kwaliteit wordt verstaan, is erg subjectief. Iedere gebruiker van een product of dienst kan er een eigen invulling aan geven en doet dat dan ook. Die subjectiviteit blijkt ook uit de definitie van het begrip 'kwaliteit' die rekening houdt met de vanzelfsprekende behoeften.

ISO definieert kwaliteit als volgt:

Kwaliteit is het geheel van eigenschappen en kenmerken van een product of dienst dat van belang is voor het voldoen aan vastgestelde of vanzelfsprekende behoeften.



Vanzelfsprekende behoeften zijn niet expliciet. Je gaat ervan uit dat iedereen er mee rekening houdt. Soms zijn deze behoeften zo vanzelfsprekend dat je er niet eens van bewust bent. Je komt er pas achter als deze behoeften niet worden vervuld.

Voorbeeld

Nederlandstaligen gebruiken standaard QWERTY-toetsenborden. Wanneer je in Nederland een toetsenbord koopt, kun je ervan uitgaan dat het toetsenbord een QWERTY-indeling heeft. Je hoeft over het algemeen niet expliciet te vermelden dat je een QWERTY-toetsenbord nodig hebt. Dit is een vanzelfsprekende behoefte van Nederlandstaligen. Franstaligen gebruiken AZERTY-toetsenborden. Als je in Frankrijk een QWERTY-toetsenbord wilt kopen, moet je deze vanzelfsprekende behoefte expliciet maken. Anders krijg je een AZERTY-toetsenbord. Het gevolg is dat je minder tevreden bent over je aankoop.



De kwaliteit van een informatiesysteem wordt dus bepaald door de mate waarin het voldoet aan vastgelegde of vanzelfsprekende behoeften. De belanghebbenden zoals de opdrachtgever en de eindgebruiker zijn degenen die deze behoeften vaststellen en uiteindelijk bepalen of hieraan is voldaan. Dat noemen we productkwaliteit. Daar tegenover staat proceskwaliteit. Hiermee wordt bedoeld of het ontwikkelproces is ingericht op een zodanige wijze dat deze bijdraagt aan een gewenste productkwaliteit. Naast product- en proceskwaliteit komt een aantal begrippen veelvuldig voor, zoals verificatie, kwalificatie en validatie. Deze begrippen worden veelvuldig gebruikt bij het bepalen van de kwaliteit van een informatiesysteem en worden later uitgelegd.

Proceskwaliteit en productkwaliteit

Bij de beoordeling van kwaliteit kan een onderscheid gemaakt worden tussen proceskwaliteit en productkwaliteit.

Proceskwaliteit

Proceskwaliteit heeft betrekking op de vraag hoe goed het productieproces is ingericht. Denk bijvoorbeeld aan een controlevraag over de manier waarop in een fabriek auto's worden geproduceerd.

- Zijn de juiste onderdelen op het juiste moment op de juiste plaats voor de montage zodat de productie zonder vertraging verloopt?

Productkwaliteit

Productkwaliteit heeft betrekking op de vraag hoe goed het product is dat door het productieproces wordt voortgebracht. Denk bijvoorbeeld aan een controlevraag over de kwaliteit van de auto zelf.

Geeft het waarschuwingssysteem van de auto een continu geluidssignaal af zodra de bestuurder het eigen portier openmaakt terwijl het licht aanstaat?

Kort gezegd:

- Proceskwaliteit heeft betrekking op de kwaliteit van de voortbrenging.
- Productkwaliteit heeft betrekking op de kwaliteit van het eindproduct.

Proceskwaliteit wordt beschouwd als een hulpmiddel bij het realiseren van productkwaliteit. Echter, een kwalitatief slecht ingericht proces zal de noodzakelijke productkwaliteit negatief beïnvloeden.

Daarnaast wordt de productkwaliteit beïnvloed door het beschikbare budget en tijd. Hoe meer budget beschikbaar wordt gesteld, hoe meer aandacht aan de kwaliteit van het product kan worden besteed door bijvoorbeeld meer te testen.

Voorbeeld



Voorbeeld van proceskwaliteit in de ICT

- De programmeurs werken samen als duo achter één computer (pair-programming). De eerste programmeur schrijft daadwerkelijk de code. De tweede programmeur controleert en verbetert voortdurend de code van de eerste programmeur.
- De vertegenwoordiger van de gebruikers voert de gebruikersacceptatietest uit zodra de programmeurs een nieuwe functionaliteit hebben opgeleverd.

Deze voorbeelden geven weer dat een bepaalde werkwijze binnen een bedrijf wordt gevolgd om goede kwaliteit op te leveren. De werkwijze is dan bedrijfs-specifiek.

Voorbeeld van productkwaliteit in de ICT

- Het systeem moet minimaal tweehonderd transacties per seconde kunnen verwerken.
- Het systeem moet de klant de mogelijkheid bieden om een bestelling te kunnen annuleren zolang de leverancier het product niet heeft verzonden.

Validatie, verificatie en kwalificatie

Binnen systeemontwikkeling zijn de volgende vormen van productbeoordeling van belang: validatie, verificatie en kwalificatie.

De definities van validatie, verificatie en kwalificatie zijn over het algemeen vrij abstract. Daarom beginnen we met voorbeelden voordat we de definities beschrijven.

Validatie

Een ICT-bedrijf ontwikkelt voor een hogeschool een elektronische leeromgeving. De hogeschool voert de acceptatietest uit. Tijdens deze test moeten de acceptanten antwoord geven op de vraag: vervult de elektronische leeromgeving hun behoeften en voldoet deze aan de overige kwaliteitseisen? Hieronder zie je vier acties als voorbeeld om de elektronische leeromgeving te kunnen valideren:

1. Het ICT-bedrijf demonstreert bij elke levering de werking van de elektronische leeromgeving aan de acceptanten.
2. De vertegenwoordigers van de studenten en van de docenten controleren de elektronische omgeving op geschiktheid voor hun gebruiksdoeleinden.
3. De opleidingsmanagers controleren of het systeem de vereiste administratieve processen kan ondersteunen.
4. De security expert van de hogeschool controleert de veiligheidsaspecten van het systeem.

Validatie is zeker stellen dat het product de behoeften van de belanghebbenden vervult en aan de overige kwaliteitseisen voldoet. Met andere woorden: bouwt de leverancier het goede systeem?



Validatie:

Bevestiging door onderzoek en door aanleveren van objectief bewijsmateriaal dat aan de eisen ten aanzien van een specifieke wijze van gebruik of toepassing is voldaan.

Verificatie

De hogeschool stelt de requirements voor de elektronische leeromgeving op in een zogenaamd SRS, Software Requirements Specification document. Deze requirements zijn de basis voor het ontwikkelen van het systeem. Het ICT-bedrijf controleert de kwaliteit van de (tussen)producten zoals het ontwerp, de code en de documentatie op basis van de gespecificeerde requirements. Op deze manier wil het ICT-bedrijf antwoord hebben op de volgende vraag: bouwen we de elektronische leeromgeving op de juiste manier? Hieronder zie je vier acties als voorbeeld voor de verificatie van de tussenproducten of van het eindproduct:

1. De senior ontwikkelaar controleert de doeltreffendheid van het ontwerp op basis van de requirements.
2. De senior ontwikkelaar controleert bij elke (deel)oplevering de code op logische fouten.
3. De testers voeren de systeemtest uit om de fouten in de deelproducten en in het eindproduct op te sporen voordat de klant met de acceptatietest start.
4. De beheerder controleert de installatieprocedures op volledigheid, correctheid, duidelijkheid en bondigheid.

Bij verificatie gaat het om de vraag of het informatiesysteem voldoet aan de specificaties die zijn opgesteld op basis van de wensen van de klant. Met andere woorden: bouwt de leverancier het systeem op de juiste manier?



Verificatie:

Bevestiging door onderzoek en door het aanleveren van objectief bewijsmateriaal dat aan de gespecificeerde eisen is voldaan.

Kwalificatie

Zowel de hogeschool als het ICT-bedrijf houdt zich aan bepaalde interne of externe normen of standaarden. Vier acties als voorbeeld om te kunnen voldoen aan de vereiste normen en standaarden:

1. De kwaliteitsmanager van de ICT-afdeling van de hogeschool controleert of de requirements volgens de ISO-norm 25010 zijn opgesteld.

ICT'ers leveren betere softwareproducten wanneer ze kwaliteitsgericht denken en werken. Kennis van software testen is hiervoor cruciaal.

Aan de slag met software testen behandelt op een gestructureerde manier fundamentele aspecten zoals principes, processen en technieken. Het geeft studenten handvatten om zich zelfstandig te verdiepen in het vak. De verworven kennis is direct toepasbaar in projecten. Elk hoofdstuk begint met een praktische oefening, gevolgd door theorie en verhelderende voorbeelden, en wordt afgesloten met leeropdrachten. Hierdoor wordt de student simultaan voorbereid op de toetsen van de opleiding en op testactiviteiten in het werkveld.

De derde druk is geactualiseerd en uitgebreid. Daarnaast is er een hoofdstuk toegevoegd over testautomatisering.

Via www.boomstudent.nl heb je toegang tot de website bij **Aan de slag met software testen**. Hier vind je onder andere meerkeuzevragen, het online boek, diagnostische toetsen met uitwerkingen, de uitwerkingen van de leeropdrachten uit het boek en verschillende workshops.

Aan de slag met software testen is geschikt voor het informaticaonderwijs en voor professionals die testexpertise willen ontwikkelen.

Hossein Chamani is als vakdocent verbonden aan de opleiding Informatica van Hogeschool Rotterdam. Gerard Kruijff werkt als trainer, docent en coach bij J & Y opleidingen. Ger Oosting is als vakdocent software engineering verbonden aan Avans Hogeschool. Jos van Rooyen is principal consultant en werkt als kwaliteitsregisseur en publicist.



9 789024 445592 >

www.boomhogeronderwijs.nl

www.boomstudent.nl