

Basis standalone

Testen en verbeteren

hoofdstuk

1

Introductie





Algemene informatie

Onderwerp	Introductie professioneel testen en verbeteren
Leerdoel(en)	<ol style="list-style-type: none">1. De student maakt kennis met het brede scala aan software testen.2. De student reflecteert op het testen en verbeteren van software op de BPV-werkplek van Thema 5 en 6.3. De student vergelijkt zijn/haar eigen in de BPV opgedane kennis en ervaring met het kwalificatiedossier.
Vereiste voorkennis	<ol style="list-style-type: none">1. Alle kennis en vaardigheden van Testen en Verbeteren Thema 1-4.2. BPV-ervaring.
Kwalificatiedossier	<ul style="list-style-type: none"><input type="checkbox"/> B1-K1-W1: Plant werkzaamheden en bewaakt de voortgang<input type="checkbox"/> B1-K1-W2: Ontwerpt software<input type="checkbox"/> B1-K1-W3: Realiseert (onderdelen van) software<input checked="" type="checkbox"/> B1-K1-W4: Test software<input checked="" type="checkbox"/> B1-K1-W5: Doet verbetervoorstellen voor de software <input checked="" type="checkbox"/> B1-K2-W1: Voert overleg<input checked="" type="checkbox"/> B1-K2-W2: Presenteert het opgeleverde werk<input checked="" type="checkbox"/> B1-K2-W3: Reflecteert op het werk



Inhoudsopgave

Algemene informatie	2
Inhoudsopgave	3
Introductie	4
Inhoud	4
Hoe ging dat op je BPV-plek?	4
Vergelijk het met het KD	5
Een afkorting uitgelegd	5
Een omschrijving van een werkproces	5
Verschillende soorten testen	7
Quality Assurance	8
Twee benaderingen	8



Introductie

We zitten in Thema 7. De opleiding begint al vorm te krijgen en krijgt al trekjes van het examen. Dit examen, gebaseerd op het Kwalificatie Dossier zal in het laatste jaar van de opleiding worden afgenomen in de vorm van een Proeve van Bekwaamheid (PvB). Tijdens deze PvB wordt de student getoetst op 8 verschillende onderdelen: de Werkprocessen. Deze werkprocessen worden dan weer verdeeld in twee groepen: de Kerntaken.

De acht werkprocessen worden in dit document ook benoemd op pagina 2 bij Kwalificatiedossier. Je ziet ook aan de vinkjes welke werkprocessen er in dit vak worden behandeld.

Los van de theoretische kennis en praktische uitvoering ervan die wij jou proberen aan te leren heb je met je BPV natuurlijk ook al veel kennis en ervaring opgedaan, misschien zelfs wel specifiek op het gebied van Testen en Verbeteren (TV).

Dit hoofdstuk van TV richt zich vooral op die opgedane kennis, het terugkijken op TV-onderdelen van je BPV en daarop reflecteren én het verdiepen op het gebied van diverse testen.

Inhoud

Hoe ging dat op je BPV-plek?

In je BPV-bedrijf heb je natuurlijk veel ervaringen opgedaan. Samenwerken in een team, kennismaken met nieuwe syntaxis en frameworks, versiebeheer, misschien een vrijdagmiddagborrel.

Ook heb je ongetwijfeld een beeld gekregen hoe de software developers de geproduceerde software en tussenliggende deelproducten testten.

Wie weet ben je zelf wel bezig geweest met het testen van je eigen of andermans software.

Het kan natuurlijk ook zijn dat je een groot deel van je BPV thuis hebt moeten werken en zo wat minder meekrijgt hoe het testen en verbeteren ging.

Daarnaast zijn er ook BPV-bedrijven die dusdanig groot zijn dat ze aparte software testers hebben: mensen die gespecialiseerd zijn in software testen en dat op diverse manieren en levels doen. Deze mensen moeten dan ook weer terugkoppelen naar de developers.

The screenshot shows a job listing for a **Software Tester (senior)** at Relatics. The listing includes the company name, location (Postcode 2984 in Ridderkerk), and a salary range of €3.000 - €5.500 per maand. It also mentions that the employer works via Indeed and provides a brief description of the role: 'Van nature kijk je verder dan enkel de eisen die rond een feature op papier staan; juist 'buiten de lijntjes' ga je na of onze software goed werkt.' The listing was posted 30+ days ago.

Deze processen, developen en testen, kunnen gehaal apart van elkaar worden uitgevoerd of juist wel samen omdat bijvoorbeeld de developers en de testers onderdeel zijn van één team.

Je merkt al dat testen dus op veel verschillende manieren kan gebeuren: door jezelf, door een collega, door een ander team, misschien zelfs door mensen in een ander gebouw. Hoewel er veel verschillen zijn heeft jouw BPV bedrijf daar waarschijnlijk wel een eigen keuze in gemaakt.



Je kunt nu **oefening 1.1** maken.



Vergelijk het met het KD

Voordat we verder gaan met de inhoud van Testen en Verbeteren en de diverse methodieken die daarbij komen kijken gaan we eens kijken wat jij zelf al weet en hebt ervaren van Testen en van Verbeteren. Maar wanneer we toch bezig zijn kunnen we ook beter gelijk het hele KD tegen het licht houden – en vergelijken met wat we al kunnen. Want misschien zonder dat je het door hebt: best wel wat zaken heb je al wel eens ervaren.

Aan de andere kant moeten we ons ook realiseren dat veel opgedane kennis en vaardigheden vanuit school of vanuit de BPV misschien ook anders wordt getoetst met de PvB. Hoe zit dat dan? Hebben we dan bepaalde vaardigheden aangeleerd die we niet gaan gebruiken? Is dat dan allemaal nutteloos geweest?

Wij, SD van het KW1C, hebben ervoor gekozen om jullie kennis te laten maken met diverse ontwikkel- en testtechnieken. Een deel daarvan komt terug bij het examen, het andere deel zul je ook in je werk later (of studie later) veelvuldig nodig hebben.

Maar vanaf dit thema en met name ook vanaf thema 8 zal de werkvorm en de gebruikte methodieken en tools steeds meer gaan aansluiten op het examen – maar ook zeker op de praktijk van veel softwareonderneming.

Doe dit nu zelf:

→ Ga naar de ELO > Thema 7 > TV > Oefeningen > Bijlagen en download het bestand Software_developer.pdf. Blader hier eens doorheen.

Door het KD tegen het licht te houden en te kijken wat jij al hebt gedaan en wat nog niet krijg je een beeld van hoe je al hebt ontwikkeld. Tegelijkertijd krijg je een beeld van het KD en wat je zoal nog zou moeten gaan oefenen / leren.

Een afkorting uitgelegd

Wanneer we kijken naar een afkorting dan kan deze bijvoorbeeld zijn: B1-K1-W4. Dit staat voor: Basisdeel 1, Kerntaak 1, Werkproces 4.

Een omschrijving van een werkproces

Zoals je misschien al wel hebt gezien bevat een werkproces een Omschrijving, een Resultaat en een Onderdeel gedrag. Het werkproces B1-K1-W4: Test Software heeft de volgende omschrijving:

De beginnend beroepsbeoefenaar maakt testscenario's voor het testen van de gerealiseerde software. Hij/zij kiest een passende testvorm, zoals eigen test van software, unit tests, integratietest, acceptatietest en kiest een passende testmethodiek. Hij/zij bepaalt wat nodig is voor het testen, zoals bijvoorbeeld de testomgeving, middelen en test data. Vervolgens voert de beginnend beroepsbeoefenaar de test(s) uit. Hij/zij interpreteert de bevindingen en trekt conclusies.



Nou, vertaal dat maar naar een concrete manier van werken. En toch is het te doen. Niet voor niets word je geëxamineerd op deze werkprocessen: ze zijn immers (mogelijke) afspiegelingen van realistische manieren van werken in de praktijk.

Om je Proeve van Bekwaamheid te behalen moet je voor ieder werkproces een voldoende halen. Dit zijn dus de werkprocessen (zoals ook weergegeven op pagina 2):

- B1-K1-W1: Plant werkzaamheden en bewaakt de voortgang
- B1-K1-W2: Ontwerpt software
- B1-K1-W3: Realiseert (onderdelen van) software
- B1-K1-W4: Test software
- B1-K1-W5: Doet verbetervoorstellen voor de software

- B1-K2-W1: Voert overleg
- B1-K2-W2: Presenteert het opgeleverde werk
- B1-K2-W3: Reflecteert op het werk

Misschien heb jij tijdens je BPV je heel veel met het ontwerpen van software – grafisch, maar ook qua database en architectuur – bezig gehouden en met het realiseren, maar niet met het testen. Het kan ook zijn dat je juist vooral aan het realiseren en testen bent geweest. Heb je veel of weinig overleg gevoerd en op welke manier? Heb je eens een keer je werk moeten presenteren aan collega's of opdrachtgever? Allemaal vragen die voor iedere student weer anders worden beantwoord.

Doe dit nu zelf:

→ Pak eens je verslaglegging van de BPV erbij en kijk eens of je werkzaamheden hebt genoteerd die mogelijk aansluiten bij het KDz.



Je kunt nu **oefening 1.2** maken.



Verschillende soorten testen

Software testen zijn onderdeel van de QA, ofwel: Quality Assurance. Niet alleen moet er software gemaakt worden. Het moet ook goed werken. En dat is een heel breed begrip.

Want software kan bijv. goed werken wanneer het door een paar mensen wordt gebruikt maar haperen wanneer duizenden mensen het gebruiken.

Of het werkt prima met bepaalde gegevens maar met andere gegevens niet.

Het kan perfect werken, maar het is toch niet volgens de eisen van de opdrachtgever.

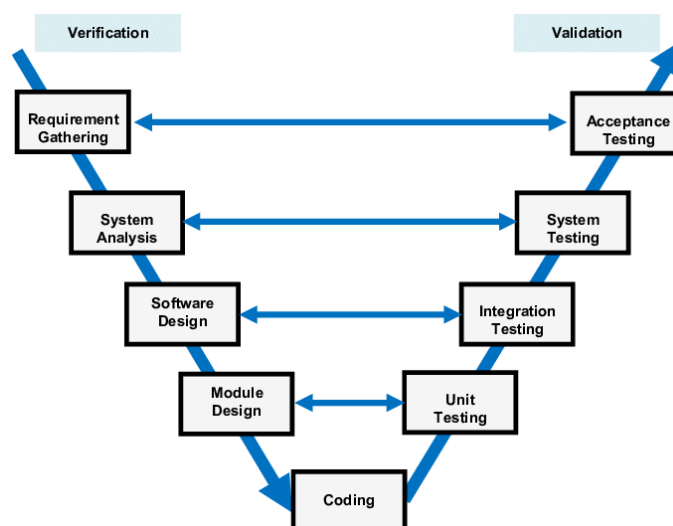
Wie weet heb je software dat goed werkt maar zo lek als een mandje is.

Het kan ook zijn dat je software goed werkt en andere onderdelen ook goed, maar samen werken de verschillende onderdelen niet goed.

En zo kunnen we nog wel even doorgaan . . .

Wat blijkt is dat er op heel veel verschillende levels en met verschillende doelen kan worden getest. En... door verschillende mensen met verschillende rollen.

Het V-model (een soort uitbreiding van de Waterfall Method – hier komen we later op terug) laat al diverse levels van testen zien gekoppeld aan verschillende fasen van software produceren:



Dit model laat zien dat een acceptatietest bijvoorbeeld wordt uitgevoerd naar aanleiding van de requirements van de software, ofwel: wat de opdrachtgever wil. Vandaar dat dit dan ook vaak mét de opdrachtgever wordt uitgevoerd. In Thema 8 gaan we het ook verder hebben over de acceptatietest.

Naarmate je verder naar beneden gaat worden de stukken software steeds kleiner en technischer, de samenhangende testen ook. Zo zie je ook dat Unit Testing (ook in Thema 8) samenhangt met softwaremodules die compleet los kunnen staan van andere softwareonderdelen binnen het project.

Doe dit nu zelf:

→ Bekijk [deze video](#).



Quality Assurance

Naast dat je software natuurlijk 'goed moet werken' probeer je ook de QA te beheren. Verschillende aspecten met bijbehorende vragen zijn:

- Effectiviteit – Doet de software wat het moet doen?
- Betrouwbaarheid – Blijft de prestatie goed wanneer er bijv. veel gebruikers zijn?
- Gebruiksvriendelijkheid – kan de software gemakkelijk gebruikt worden?
- Flexibiliteit – Kan het goed worden gewijzigd en goed worden verplaatst binnen systemen?
- Onderhoudbaarheid – Kan het goed onderhouden worden?
- Beheerbaarheid – Kan het goed operationeel gebracht worden en gehouden worden?
- Beveiliging – Zijn zaken als privacy, diefstal en moedwillige verstoringen geborgd?
- Efficiency – Is er een balans tussen de prestatieniveau en de hoeveelheid middelen?

Twee benaderingen

Er zijn van oudsher al twee benaderingen:

- **De positieve benadering** – Voldoet het systeem of product aan de belangrijkste eisen van gebruik?
- **De negatieve benadering** – Kunnen fouten aangetoond worden in het systeem of product?



Je kunt nu **oefening 1.3** maken.