

Proces 6: Testverslag en acceptatietest

Introductie

Een software developer is niet alleen maar bezig met programmeren. Er gaat veel tijd zitten in testen en debuggen (foutafhandeling). In deze module wordt aandacht besteed aan het testen van de applicatie die je hebt gemaakt.

Als je gaat testen, moet je jezelf een aantal vragen stellen:

- Wat wordt er getest?
- Wanneer wordt er getest?
- Hoe wordt er getest?
- Door wie wordt er getest?
- Wat zijn de stappen die we na het testen nog moeten nemen?

Tijdens het testen maak je gebruik van een testformulier waarin je de resultaten vastlegt. Dit levert een verslag op, waarin onder andere de vervolgstappen staan: wat ging er fout en hoe lossen we dat op?

De leerdoelen voor dit thema zijn:

Je kan....:

- ...code testen op afwijkingen.
- ...een testverslag opstellen.
- ...een acceptatietest opstellen en uitvoeren.
- ...conclusies trekken uit een acceptatietest.

Tijdsplanning

Week 1			
BASIS1A	½ uur	Uitleg Unit testen	
	½ uur	Oefeningen	
	1 uur	Opdracht	
Week 2			
BASIS1A	½ uur	Uitleg Acceptatietest	
	½ uur	Oefeningen	
	1 uur	Opdracht	

LES 1

OEFENING

We beginnen met een kleine oefening.

Vergelijk de volgende 2 websites met elkaar:

- Bol.com: <https://www.bol.com/nl/>
- Google.nl: <https://www.google.nl/>

Geef antwoord op de volgende vragen:

- Welke is beter? En waarom?
- Wat is er goed aan bol.com? En wat is er slecht aan?
- Wat is er goed aan google.nl? En wat is er slecht aan?

THEORIE

Je ziet dat het lastig is om de twee websites met elkaar te vergelijken. Het is alsof je appels met peren vergelijkt.

Eigenlijk weet je pas of een site “goed” is, als je weet wat de eisen van de site zijn: wat moet de website doen, en doet ‘ie dat ook? Dit zijn de **eisen van de klant**.

Naast de eisen van de klant moet een applicatie vaak ook nog voldoen aan een aantal **technische eisen**.

Denk aan:

- Is de site gebruikersvriendelijk (usability)
- Werken alle links, afbeeldingen en formulieren goed
- Is de website goed te vinden op internet (SEO)
-

Voor de technische eisen kunnen vaak standaardformulieren worden gebruikt.

Om deze technische eisen te testen, kan je gebruik maken van diverse online tools. Hier komen we later op terug.

Soorten tests

De vraag is nu: hoe gaan we testen?

Er zijn diverse soorten tests. Op Wikipedia vind je een (soort) overzicht van de diverse soorten softwaretests die je kan uitvoeren:

<https://nl.wikipedia.org/wiki/Categorie:Softwaretest>

We kunnen ze niet allemaal doen. Deze module ga je in ieder geval 2 tests uitvoeren:

- Unit test
- Acceptatietest

UNIT TEST

De **unit test** is een testmethode waarbij stukjes van de applicatie apart worden getest. Denk aan een formulier en de verwerkingen ervan. Of aan een landingspagina (indexpagina) met allemaal afbeeldingen, tekst en linkjes.

Zie: <https://nl.wikipedia.org/wiki/Unittesten>

Hoe voer je een unit test uit?

Dit kan heel uitgebreid. Vaak wordt er zelfs extra code geschreven om een unit test mee uit te voeren. Voorlopig houden we het wat eenvoudiger:

- Check de code:
 - o De code is netjes geordend.
 - o Naamgeving variabelen klopt.
Geen 'var1', 'var2', ... maar 'naam', 'adres', 'som',
 - o Alle functies en berekeningen werken en geven de verwachte uitkomst.
Test dit met verschillende waarden.
 - o TIP: Kijk eens op w3schools voor 'code-conventions':
https://www.w3schools.com/js/js_conventions.asp
- Check de werking in diverse browsers:
 - o Alle afbeeldingen en teksten zijn goed zichtbaar, ook op kleiner scherm.
 - o Alle hyperlinks werken en staan op de goede plaats.
 - o Alle formulervelden werken goed; het formulier wordt goed verwerkt.
 - o Alles is duidelijk voor de gebruiker (usability).
- Probeer de grenzen op te zoeken:
 - o Kan ik rare gegevens invullen in een formulier?
 - o Wat gebeurt er als ik de pagina heel smal maak?
 - o

Wie test er?

In principe kan je een unit test goed zelf uitvoeren. Maar vaak is het slim om ook iemand anders te laten testen. Zelf weet je namelijk hoe (dit deel van) de applicatie moet werken, terwijl iemand anders vaak andere dingen probeert.

Hoe leg je het vast?

Om te testen wordt een formulier gebruikt, waarin beschreven staat wat er wordt getest. Achter elke regel (eis) kan worden aangegeven of het onderdeel goed werkt of niet. Zo niet, dan wordt aangegeven wat er fout gaat.

Zo'n formulier kan er als volgt uitzien:

UNIT TEST		Datum:	
Uitgevoerd door:			
		In orde?	Opmerking
Code			
Commentaarregels gebruikt			
Juiste variabele namen gebruikt			
Goed ingesprongen			
Geen fouten in de code			
Code geeft de juiste uitvoer			

Werking		
Pagina laadt snel		
Alle onderdelen zijn zichtbaar en staan op de juiste plaats		
Structuur pagina blijft goed bij kleiner scherm		
Formulievelden werken goed		
Knoppen werken goed		
Uitvoer duidelijk op het scherm		
De pagina werkt goed in diverse browsers		
Overige opmerkingen:		

Dit formulier zal verder worden uitgebreid, afhankelijk van de eisen van het programmaonderdeel.

Na de test...

Nadat het formulier is ingevuld kan je een conclusie trekken: werkt dit onderdeel goed of niet? Wat moet er verbeterd worden? En wie gaat dat doen en wanneer moet dit af zijn? Dit wordt vastgelegd in een verslag.

Nadat de verbeteringen zijn doorgevoerd, zal er een nieuwe unit test moeten plaatsvinden om te controleren of alle onderdelen nu wel goed werken.

OEFENING

Onderstaande code checkt hoe laat het is. Afhankelijk van de tijd, krijgt de gebruiker de melding 'goedemorgen', 'goedemiddag', 'goedenavond' of 'goedenacht'. De melding verschijnt in een div op de pagina.

```
<!DOCTYPE html>
<html>
<head>
  <title>Proces6, oefening</title>
</head>
<body>
<h1>DE TIJD:</h1>
<div id="div1"></div>
<div id="div2"></div>
<script>
  let x = new Date();
  document.getElementById('div1').innerHTML = "Het is " + time;
  if (x.getHours() > 0 && x.getHours() < 6)
  {
    window.alert ("Goedenacht");
  }
  if (x.getHours() > 6 && x.getHours() < 12)
  {
    window.alert ("Goedemorgen");
  }
  if (x.getHours() > 12 && x.getHours() < 18)
  {
    window.alert ("Goedemiddag");
  }
</script>
</body>
</html>
```

Neem de code over in een HTML-pagina. Test of de code goed werkt. Geef aan of elk onderdeel goed werkt en wat er fout gaat. Maak een formulier waarin je de test vastlegt. Moet je wachten tot 12 uur 's nachts om te weten of de code 'goedenacht' laat zien? Hoe test je dat?

OPDRACHT

Deze opdracht maak je in groepjes van 2 personen.

Bij BASIS1B/BASIS1C heb je de eerste module van Javascript doorgenomen en de bijbehorende opdrachten gemaakt.

- Open één van de twee eindopdrachten ('De ziekte-identificatie app' of 'De begroting'). Als je deze opdrachten nog niet hebt gemaakt, dan mag je ook opdracht 6 ('prijs.html') gebruiken.
- Voer een unit test uit op deze opdracht.
- Check of de code netjes is en of alle eisen in de applicatie zijn verwerkt. Controleer of de applicatie goed werkt in de browser(s).
- Maak een volledig verslag van de test:
 - Wat was de opdracht? (Wat moet deze pagina doen?)
 - Maak voor de unit test een formulier en vul deze volledig in.
 - Geef aan wat de conclusie is.
 - Welke vervolgstappen moeten er genomen worden?

LES 2

ACCEPTATIE TEST

Bij een groot project zal er tussendoor een aantal keer een tussenoplevering zijn aan de klant (opdrachtgever). De klant kijkt dan of het opgeleverde deel voldoet aan de eisen. Aan het eind van het project, in de laatste testfase, vindt de laatste oplevering plaats. Dit gebeurt via een **acceptatietest**.

De acceptatietest is een onderdeel van het zogenoemde OTAP-proces: **O**ntwikkeling, **T**est, **A**ccceptatie en **P**roductie.

Dus: als de acceptatietest goed verloopt dan kan de applicatie in productie worden genomen.

Wat wordt er bij een acceptatietest allemaal getest?

In principe wordt de volledige applicatie getest op verschillende onderdelen. Welke onderdelen dit zijn, ligt aan de klant.

Allereerst zullen alle **eisen** van de klant worden gecheckt. Deze zijn natuurlijk per applicatie verschillend.

Daarnaast worden nog een aantal **algemene onderdelen** gecheckt, zoals usability, design, techniek en SEO. Op internet zijn verschillende checklists te vinden die je hiervoor kan gebruiken.

Kijk maar eens op: <https://www.rifffonlinemarketing.nl/blog/conversie-optimalisatie/25-criteria-om-de-kwaliteit-van-jouw-website-te-beoordelen/>

Hoe test je?

Daar moet natuurlijk staan: hoe test de klant? Allereerst zal de klant de applicatie testen op zijn/haar eigen device(s). Daarnaast zal de klant een aantal dingen uitproberen om te kijken hoe veilig en robuust de applicatie is. Dit laatste wordt ook wel **monkey testing** genoemd. Zie https://en.wikipedia.org/wiki/Monkey_testing.

Op Internet zijn er verschillende **online tools** te vinden om de applicatie/website te testen. Zo kan je jouw applicatie testen op snelheid, SEO, security...

Onderstaande site heeft een aantal tests verzameld:

<https://anotherconcept.nl/2019/02/15/de-top-4-beste-tools-om-jouw-website-te-testen/>

Als je op Google zoekt naar 'website test' zal je er nog meer vinden.

Probeer er een aantal uit. De uitkomsten zal je niet altijd begrijpen, maar je kan wel zien hoe jouw applicatie scoort.

Als laatste: **AVG**.

AVG staat voor *Algemene Verordening Persoonsgegevens* en beschermt de persoonsgegevens van Internetgebruikers.

Als jouw website gegevens bewaart van bezoekers, of als je cookies plaatst, dan moet je dit aangeven op je website. En je moet die gegevens natuurlijk veilig en niet langer dan noodzakelijk opslaan. Kijk maar eens op <https://www.heijteci.nl/avg-gdpr/> , of ga op zoek naar een andere site over AVG.

Check ook altijd of jouw website AVG-proof is!

Verslag

De acceptatietest levert een verslag op.

- Zorg voor een voorblad.
Hierop staat de naam van de applicatie, de datum, de namen van de makers (developers) en de naam van de opdrachtgever/klant.
- Allereerst wordt kort uitgelegd welke applicatie er wordt getest.
Waar staat de applicatie (URL) en wat moet de applicatie doen (eisen).
- Daarnaast staat het testformulier in het verslag. Als er meer mensen hebben getest, dan staan er dus meer formulieren in het verslag. Op elk testformulier staat wie getest heeft en wanneer.
- Wat zijn de conclusies die je kan trekken n.a.v. de tests?
Zijn er onderdelen die nog niet goed werken? Wat moet daaraan worden veranderd?
Wanneer (en door wie) wordt de website verbeterd?
En wanneer kan de website opnieuw opgeleverd worden?

OEFENING

Maak een formulier dat je kan gebruiken om een acceptatietest uit te voeren. Zorg dat de eisen van de klant erin staan en de algemene onderdelen die gecheckt moeten worden. Vergeet niet om bovenaan het verslag ruimte te maken voor de naam van de testpersoon, de datum en de URL van de applicatie.

OEFENING

Ga op Internet op zoek naar een onlinetest om een website te testen.

Bijvoorbeeld <https://nibbler.silktide.com/>

Vul een website in om te testen.

Neem bijvoorbeeld <https://www.glr.nl/> of <https://ict-lab.nl/>

Kijk hoe de website scoort en probeer hier conclusies uit te trekken: waar scoort de site goed op en wat kan beter?

OPDRACHT

Deze opdracht maak je in groepjes van 2 personen.

Bij BEROEPS1 hebben jullie een website voor een dierentuin moeten maken.

Deze website ga je uitgebreid testen.

- Maak een testformulier. Zorg dat het testformulier de onderdelen bevat die in deze reader zijn beschreven.
- Laat minimaal twee mensen een acceptatietest uitvoeren.
- Verzamel de tests en schrijf een volledig testverslag.

Het testverslag lever je in.

EXTERNE BRONNEN

Voor extra informatie kan je gebruik maken van de volgende links:

- Testen: [https://nl.wikipedia.org/wiki/Testen_\(software\)](https://nl.wikipedia.org/wiki/Testen_(software))
- Unit test: <https://www.guru99.com/unit-testing-guide.html>
- Acceptatietest: <https://www.applinet.nl/artikelen/acceptatietest.html>
- Pentest: <https://nl.wikipedia.org/wiki/Penetratietest>
- AVG: <https://www.yoursafetynet.com/nl/checklist-met-deze-10-stappen-maak-je-jouw-website-avg-proof/>

Online tests:

- Algemeen: <https://nibbler.silktide.com/>
- Speedtest: <https://www.uptrends.nl/tools/website-speed-test>
- SEO: <https://sitechecker.pro/app/main/project/1467580/audit/summary>
- Security: <https://pentest-tools.com/website-vulnerability-scanning/website-scanner>

Google analytics: <https://analytics.google.com/analytics/web/provision/#/provision>

BEOORDELING

Module	PROCES 6 Testverslag & acceptatietest
Naam beoordelaar	
Naam student	
Datum	
Leerdoelen:	O/V
Je test code op afwijkingen (opdr. 1)	
Je stelt een testverslag op (opdr. 1/2)	
Je kan een acceptatietest opstellen en uitvoeren (opdr. 2)	
Je kan conclusies trekken uit een acceptatietest (opdr. 2)	
Opmerkingen:	
Normering:	Score:
Leerdoelen voldoende	7
Extra onderdelen	1,5 (maximaal)
Vormgeving van de ingeleverde documenten	1 (maximaal)
Bij uitzonderlijk goed werk	0,5 (maximaal)
<i>Bij te laat inleveren</i>	<i>Maximum cijfer: 6</i>
<i>Niets inleveren</i>	<i>Cijfer: 1</i>
Cijfer:	
Werkproces B1-K1-W4: test software	O/V
Je voert snel, correct en adequaat de testactiviteiten uit.	
Je interpreteert de testresultaten en trekt logische conclusies.	
Je legt testresultaten en conclusies nauwkeurig, duidelijk en conform bedrijfs- of beroepsstandaarden vast.	
Opmerkingen:	