

Project portfolio Nick Wilschut.

Naam project: SBA-Werkbonnen
Datum: 23 februari 2021
Naam: Nick Wilschut
student nummer: 99048866

Inleiding:

Ik heb wegens omstandigheden niet deelgenomen aan een groepsproject op school, daarom heb ik het project wat ik bij SBA met medestudenten gedraaid heb gebruikt voor dit portfolio.

Hier een kort overzicht van het project SBA-Werkbonnen:

- In totaal zijn er 10 sprints geweest.
- Er is aan 99 stappen gewerkt in totaal binnen het project.
- Het project heeft bijna 4 maanden geduurd (2019-10-09 tot 2020-02-07).
- Het project is gebouwd in VueJs en Codelgniter + SBA backend architectuur.

Het doel van het project was om een applicatie te bouwen waarin werknemers van SBA werkbonnen konden gebruiken. Dus stel dat een werknemer een internet kabel aan zou moeten leggen bij een klant, zou deze werknemer een werkbbon kunnen aanmaken, hierin invullen hoelang hij/zij hier mee bezig gaat zijn en welke producten/gereedschap hij/zij hiervoor nodig heeft. Dit systeem zou er voor moeten gaan zorgen dat er geen producten of gereedschap onbetaald achter kunnen blijven bij de klant.

Project kick-off (van 2019-10-9 tot 2019-10-28):

Voordat wij het project konden gaan bouwen hebben wij eerst business processen gemaakt en deze in een groot document verwerkt waarin wij beschreven hoe wij de applicatie wilde gaan bouwen. Hierop hebben wij feedback gekregen van de klant (in dit geval Sander Brouwer). Deze feedback was bijvoorbeeld op de structuur die wij bedacht hadden, toen hebben wij dit verwerkt en werd uiteindelijk het document goed gekeurd.

Toen dit gebeurt was konden wij Wireframes maken. Dit zijn schetsen hoe de applicatie er uit komt te zien. Hierbij moest uiteraard rekening gehouden worden met de structuur die wij bedacht hadden in de business processen.

Iteratie plannen (elke week):

Elke week na de Iteratie oplevering (zie hieronder), hebben wij een planning gemaakt voor aankomende sprint. Dit deden wij in de SBA Customer Portal. Dit is het systeem wat bij SBA gebruikt wordt om sprints/stappen in te plannen. Voor het plannen van een sprint moesten wij stappen maken. In deze stappen hebben wij beschreven wat wij gingen doen. Ook moest hierin aangegeven worden welke dagen er aan de stap gewerkt kan worden, en hoeveel tijd dit ging kosten hierbij moest ook rekening met een test en een revisie dag gehouden worden. Tot slot moesten wij een testcase met het verwachte eindresultaat schrijven die de tester door kon lopen.

Een voorbeeld van zo'n stap zou kunnen zijn:

Naam stap: Login pagina.

Beschrijving: Het maken van een pagina waarin een gebruiker kan inloggen.....

Datum: 2019-10-28 tot 2019-11-1

Uren: 8

testcase:

ga naar www.voorbeeld.nl en kijk of u direct op de login pagina komt. Vul hier de volgende gegevens in:

username: verkeerdenaam

password: verkeerdwachtwoord

kijk of er nu een foutmelding verschijnt. Vul nu de volgende gegevens in:

username: testgebruiker

password: testgebruiker123

Kijk of u nu een success melding krijgt en naar de dashboard pagina doorgestuurd wordt.

Status: **concept**

verantwoordelijke: Nick Wilschut

De stap zou eerst door de scrum-master gecontroleerd worden en daarna goed of afgekeurd worden. In het geval dat de stap afgekeurd werd, moest de bijbehorende feedback verbeterd worden en de stap opnieuw gecontroleerd worden. In het geval dat de stap goedgekeurd werd, kon er aan gewerkt worden.

Iteratie opleveren:

Aan het eind van elke sprint hadden wij een oplevering aan de klant. Hiervoor moesten wij een dag van te voren een opleverings-document opsturen. In dit document moest staan welke stappen er af gemaakt waren, welke niet en welke bijzonderheden er waren (voor als iemand ziek was bijvoorbeeld). Hierin stonden ook alle testcases voor de klant + de test url van de applicatie. Ook werden punten als known issues en opgeloste bugs in het document verwerkt. Elke week zette wij dus ook een nieuwe release online op de test omgeving voor de klant. Op deze manier heeft de klant al tijd om de applicatie doorkijken voordat de meeting begint, en dus al tijd om feedback te bedenken die daarna in de meeting besproken werd.

Tijdens de meeting liepen we door de applicatie aan de hand van de stappen/features die voltooid waren. Als de klant hier dan feedback op had kon hij dit direct aangeven en werd dit opgeschreven door een van de andere medestudenten. Nadat de oplevering had plaats gevonden konden wij weer een planning maken voor de volgende sprint.

Code Review:

Als een stap af kwam tijdens de sprint moest deze eerst getest worden door een medestudent. De stap zou op zijn/haar naam gezet worden met de status Testing waardoor de stap in zijn/haar planning kwam te staan. Zodra de medestudent hier tijd voor had kon hij/zij de stap gaan testen, dit werd aan de hand van de eerder uitgeschreven testcase en de wireframes gedaan.

Als de stap werkte zoals beschreven, kon deze de status **completed** krijgen, en in het opleverings document bij de voltooide features toegevoegd worden.

Als de stap nog niet voldeed aan de eisen kreeg deze de status **revisie**, en werden de bugs of styling fouten in de testresults van de stap gezet. Ook werd de stap teruggezet naar de ontwikkelaar. Dit hield in dat hij/zij de feedback van de test moest verwerken totdat dit helemaal functioneert. Hierna kon de stap opnieuw getest worden. Onderhoud:

Als er bugs terug kwamen van de applicatie moesten deze verwerkt worden. De bugs konden ingeschoten worden in de customerportal, omdat bugs en issues altijd een hogere prioriteit hebben dan het verder ontwikkelen van de applicatie, moesten deze zo snel mogelijk verwerkt worden. Deze bugs werden dan als een stap behandeld, maar dan met hogere prioriteit, dit hield dus in dat er iemand de bug moest oplossen en vervolgens iemand moest testen of de bug goed opgelost was of niet. Als het goed opgelost was kon de bug de status **completed** krijgen en kon dit terug gekoppeld worden aan de klant. Over de release built is meer te lezen in het stuk Iteratie opleveren.

Beveiligingsrisico's:

In de standaard backend architectuur van SBA wordt al veel rekening gehouden met security risico's, er word veel met Authentication en Authorisation gedaan. In de meeste applicaties zijn daarom ook 4-6 gebruikers rollen. Het exacte aantal verschilt per applicatie omdat elke applicatie een anders functioneert. De Authentication is in de meeste applicaties het zelfde omdat dit gewoon een standaard login/registreer systeem is. In sommige applicaties is het registreer functionaliteit alleen mogelijk vanuit het admin systeem om ongewenste buitenstaanders tegen te gaan.