



Praktijkleren

samen sterk onderwijs

Examenafspraken

Proeve van Bekwaamheid

| Algemene informatie | | |
|----------------------------------|--|----------------|
| Examenvorm | Proeve van bekwaamheid – Praktijkexamen in de beroepspraktijk | |
| Kwalificatiedossier en cohort | Software development | 2020 en verder |
| Profiel, niveau en crebocode | P1: Software developer, niveau 4 | 25604 |
| Examencode | SD_SD20-PE1_B1-K1-2_1v1 | |
| Kerntaak | B1-K1: Realiseert software B1-K2: Werkt in een ontwikkelteam | |
| Werkprocessen en opdrachtnummers | 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 | Opdracht 1 |
| Vaststellingsdatum | 27-10-2020 | |

| Persoonsinformatie | |
|--------------------|---------------|
| Datum | 13/02/2024 |
| Naam kandidaat | Shibin Pan |
| Studentnummer | 220617 |
| Klas/groep | SEITO-21A |
| Beoordelaar 1 | Frans de Boer |
| Beoordelaar 2 | Marcel Kanon |

Algemeen

Dit is het document *Examenafspraken*. Gebruik voor het concretiseren van dit document de *Invulinstructie voor examenafspraken*.

Referentieopdrachten

Mijn examenopdracht wordt het bouwen van een tool die van PDF naar ProdistERP XML-bestand omzet. De bedoeling is dat deze tool een/meerdere PDF-bestand kan analyseren en de inhoud zo goed mogelijk zien te vertalen naar een XML-bestand, zodat ProdistERP-applicatie uiteindelijk kan inlezen en importeren.

Concrete invulling examenopdrachten

Beschrijf hier de gekozen invulling van de opdrachten, de context waarbinnen het examen plaatsvindt en de afdnamecondities.

[Examenopdracht 1]

Het bouwen van een tool die PDF-bestand omzet naar XML-bestand, zodat de ERP-applicatie uiteindelijk kan inlezen. ERP staat voor Enterprise Resource Planning, met onder andere de relatiebeheer, voorraadbeheer, calculatiemodule, facturatie. Het pakket wordt voornamelijk gebruikt door grafische industrie, maar ook in Displaybouwers, Signbedrijven, Natuursteen en botenbouwers, enzovoort, waarbij maatwerk steeds vaker wordt toegepast. Op dit moment gaat het zo als mensen een bestelling hebben geplaatst, moeten de bedrijven zelf de orders handmatig invullen op ProdistERP applicatie om voorraad bij te werken, of een productieproces te beginnen, of überhaupt de facturatie klaar te zetten. Dat kost natuurlijk veel tijd. Het stagebedrijf heeft daar een idee bedacht, waarbij ik een tool ga maken dat die een PDF-bestand analyseert en de inhoud van die PDF zo goed mogelijk zien te vertalen naar een XML-bestand zodat de ERP-applicatie uiteindelijk kan inlezen. Dat bespaart veel tijd.

De tool hoeft niet voor alle willekeurige PDF-bestanden te gelden. De examenopdracht is beperkt tot de PDF-bestanden van een hotelketen. Dus de opmaak van de PDF-bestanden ziet er hetzelfde uit. Maar omdat een PDF-bestand een geprint bestand is, zijn alleen de vaste gegevens makkelijk te herleiden, zoals het klantnummer, de afzender, etc. Maar voor de rest zijn er nog variabele gegevens, want je hebt soms een bestelling van maar 1 artikel, en soms heb je ook een bestelling van 20 verschillende artikelen. Daar moet ik ook nog rekening mee houden.

Het doel van de tool is om bedrijven het proces voor facturatie sneller, makkelijker en foutloos te maken. Ze hoeven niet meer de orders handmatig in te vullen. Met de tool kunnen ze in een keer meerdere PDF-bestanden laten verwerken, en foutloos ook.

Mijn stagebegeleider wil graag dat proces automatisch gebeurt. Nadat het bedrijf de PDF-bestanden in een map plaatst, worden de bestanden automatisch uitgelezen, en geëxporteerd als een XML-bestand.

Ik ben van plan om het eerste een prototype te ontwikkelen. In de eerste week heb ik al een opwarmopdracht gekregen dat ik een tool moet maken waarbij de gebruikers bestanden kunnen uploaden naar een specifieke map(en de map kunnen ze ook zelf selecteren). Deze tool kan ik dus goed gebruiken om het prototype te maken.

1. De gebruiker krijgt een scherm waarbij hij een pad kan selecteren(waar hij het Xml bestand uiteindelijk naartoe wil), en dat hij de PDF bestanden kan uploaden.
2. De tool moet PDF-profiel kunnen herkennen(gaat het om een specifieke hotelketen dan doorgaan, anders overslaan).
3. Xml bestand wordt aangemaakt met dezelfde naam in de map die de gebruiker eerder heeft geselecteerd
4. PDF uitlezen en omzetten naar dat correct opgebouwd XML bestand

Het uitlezen van de PDF bestanden worden bepaalde teksten uit specifieke coördinaties gehaald, vervolgens kunnen deze teksten gekoppeld worden aan relevante XML tags. Verder moet diverse onderdelen herkend worden zoals de afleverdatum, afleveradres en de bestelde artikelen etc.

Zodra het XML bestand is gegenereerd en het ingelezen kan worden in ProdistribERP. Dan is in principe het prototype klaar.

Het maken van het eindproduct gebeurt niet met een user interface(*automatisering). Dus een specifieke map wordt constant in de gaten gehouden, zodra er wijzigingen worden aangebracht, wordt mijn tool geactiveerd.

B1-K1 & B1-K2

Examenopdracht 1

Je gaat in dit examen een applicatie ontwikkelen. Je werkt als een projectmedewerker in een projectteam. Dit projectteam werkt op een iteratieve manier. Er wordt minimaal 1 iteratie (sprint) uitgewerkt. Je bent verantwoordelijk voor je eigen uitgewerkte user stories¹.

Opstarten project

1. Ontvang de opdracht van de opdrachtgever.
2. Verzamel eventuele aanvullende informatie. Bijvoorbeeld door internet of de klant te raadplegen.
3. Formuleer de uitgangspunten, eisen en wensen voor het project in een document (bijvoorbeeld Definition of done). Doe dit volgens de geldende bedrijfsprocedures. Dit omvat bijvoorbeeld het volgende.
 - a. Projectdoel: Wat wil de klant bereiken met het product?
 - b. Doel van het team en eigen doelen (bijvoorbeeld security-/performance-eisen die niet binnen een user story passen).
 - c. Te gebruiken technieken en codeconventies
4. Stem af met de betrokkenen over het document met uitgangspunten, eisen en wensen.
5. Stel op basis van het document de user stories op. Zet deze user stories op de Product backlog.

Uitvoeren iteratie (sprint)

1. Plan en plaats de user stories. Schat onder andere de tijd en de prioriteit/volgorde in (bijvoorbeeld met de MoSCoW-methode). Plaats de user stories op de iteratie/sprint backlog.
2. Verdeel binnen het team de eerste op te pakken user stories. Zorg voor een evenwichtige verdeling.
3. Werk de eigen user stories uit in taken, criteria en een ontwerp. Stel hierbij, passend bij de opdracht, de volgende documenten op of voer de volgende activiteiten uit, zoals:
 - a. een ERD (datamodel);
 - b. een use case;
 - c. een klassendiagram;
 - d. normaliseren;
 - e. wireframes;
 - f. mock-ups;
 - g. een activiteitendiagram;
 - h. een keuze framework.
4. Realiseer de eigen user stories. Stel hierbij, passend bij de opdracht, de volgende documenten op of voer de volgende activiteiten uit, bijvoorbeeld het volgende.
 - a. Programmeren
 - b. (Unit)testen

¹ Overall waar 'user stories' staat, kan ook '(deel)functionaliteit' worden gelezen als er niet met SCRUM wordt gewerkt.

- c. Versiebeheer bijhouden
 - d. Documenteren (in de code of apart)
 - e. Een database inrichten/aanpassen/aanvullen (indien nodig en afhankelijk van de situatie)
 - f. Overleggen met andere betrokken disciplines (bijvoorbeeld het designteam of de security officer) (indien nodig en afhankelijk van de situatie)
 - g. Een codereview (onderling)
 - h. Een contactmoment met het team om desbetreffende story te kunnen realiseren (optioneel en afhankelijk van ontwikkelmethode. Bijvoorbeeld een stand-up meeting)
 - i. Een planning bijhouden (bijv. in/op DevOps/Trello/whiteboards/flip-over)
5. Voeg met het team de verschillende user stories samen.
6. Voer een test uit voor alle user stories uit de sprint die samen het product maken.
 - a. Stel een scenario (inclusief testdata) op voor de test.
 - b. Voer de test volgens het scenario op.
 - c. Stel conclusies en verbetervoorstellen op voor je eigen user stories.
 - d. Neem deze verbetervoorstellen mee naar de review met de klant, indien nodig.

Opleveren

1. Lever het product op aan de klant (review).
 - a. Houd bijvoorbeeld een demo of een presentatie. Ieder lid van het projectteam presenteert hierbij een of meer user stories.
 - b. Stel naar aanleiding van de oplevering verbetervoorstellen voor het product op indien nodig.
 - c. Bundel de verbetervoorstellen van de integratietest en review van jouw user stories in een document.

Reflecteren

1. Houd een reflectiemeeting (retrospective).
 - a. Stel naar aanleiding van de reflectiemeeting verbetervoorstellen op voor het proces in de volgende iteratie/sprint indien nodig.
 - b. Verwerk de verbetervoorstellen in nieuwe epics/user stories en zet deze op de Sprint backlog.

Voeren eindgesprek

1. Individueel technisch inhoudelijk gesprek over je opgeleverde werk (max. 10 min).

Resultaat

Als resultaat van deze opdracht lever je de volgende producten en/of diensten op.

- Document uitgangspunten, eisen en wensen
- Notulen reflectiemeeting
- Applicatie/user stories
- Document met verbetervoorstellen

Aanvullende afspraken

Mijn examenopdracht is natuurlijk niet een compleet project als een website of applicatie. Het is een tool waarmee je iets kan bereiken. Wel wordt elke week een update gegeven wat ik de afgelopen week heb gedaan en wat ik volgende week ga doen. Er wordt trello bord bijgehouden, en een github omgeving waar ik alle documentaties en codes online kan zetten.

Functioneel ontwerp: huidig- gewenst situatie, onderzoek PDF uitlezen en inlezen naar XML, stappen van het proces. (pdf, controleer, uitlezen, exporteren...)

Research analyse: Er wordt een research analyse gemaakt om beter inzicht te krijgen voor de beste aanpak, in de analyse komt huidige oplossingen, bestaande technologieën, schaalbaarheid etc.

Technische ontwerp: In dit document wordt duidelijk gemaakt welke programmeertaal en welke tools er wordt/worden gebruikt.

Evaluatierapport: hoe dit project ging, hoe het proces ging, reflectie op mezelf.

Periode, beschikbare tijd en afgesproken beoordeelmomenten

Week 7: Kennismaking en opwarmopdracht. Eerste versie Examenopdracht opleveren

Week 8: Eventueel Examenopdracht verbeteren, Beginnen met afgesproken documentaties

Week 9: Functioneel ontwerp, Research analyse, Technisch ontwerp

Week 10: Documentaties,

Week 11: Beginnen met eerst prototype,

Week 12: Ontwikkelen,

Week 13: Ontwikkelen, Tussentijdsbeoordeling

Week 14: Ontwikkelen,

Week 15: Ontwikkelen,

Week 16: Ontwikkelen

Week 17: Ontwikkelen,

Week 18: Ontwikkelen,

Week 19: Ontwikkelen,

Week 20: Testen, Tussentijdsbeoordeling, Evaluatierapport

Week 21: Testen, Presentatie maken, Presentatie voorbereiden

Week 22: Voor 27 mei alles inleveren

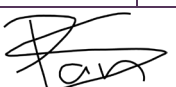

Week 23: 4 juni 2024 8:30 10:00 presentatie

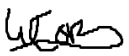
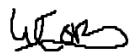
...


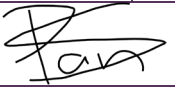
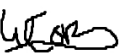
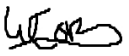
Week 26: Laatste tussentijdsbeoordeling.

Week 27: Afronding

Start en einde examen

| B1-K1 | | | | |
|---------------------------|---|------|---|-------|
| Examenduur | 12/02/2024 | 9:00 | 12/07/2024 | 17:00 |
| Handtekening Kandidaat |  | |  | |

| | | |
|-------------------------------|---|---|
| Handtekening Beoordelaar 1 |  |  |
| Handtekening Beoordelaar 2 | | |

| B1-K2 | | | | |
|-------------------------------|---|------|---|-------|
| Examenduur | 12/02/2024 | 9:00 | 12/07/2024 | 17:00 |
| Handtekening Kandidaat |  | |  | |
| Handtekening Beoordelaar 1 |  | |  | |
| Handtekening Beoordelaar 2 | | | | |