**Group 3 DevOps Document**



**Gemaakt door:**

**Renas Khalil**

**Sadek Al Mousawi**

**Sam Maijer**

**Daniel Pustjens**

**Module: DevOps | WFSDAD.DO.23**

**Versie: 2.0**

**Datum: 05-05-2024**

**Inhoudsopgave**

- Projectkeuze

- DevOps

- Praktijken & technieken

- Aandachtspunten & verbeterstrategieën

- Actieplan

**Projectkeuze**

We hebben ervoor gekozen om **SL2B** te gebruiken voor ons projectkeuze, omdat we veel van de voorbeelden die voorkomen in lessen van **SL2B & DevOps** toegepast kunnen worden in het project zelf. Op die manier begrijpen we niet alleen de lesstof beter, maar kunnen we ook sneller en efficiënter werken, waarbij we meer kennis opbouwen.

Daarnaast biedt SL2B handige ingebouwde tools aan om onze code te testen. Door het gebruik van deze tools kunnen wij ervoor zorgen dat onze applicatie vanaf het begin stabiel is en dat eventuele fouten vroegtijdig worden ontdekt en opgelost. Hierdoor kunnen we een sterke applicatie bouwen en ervoor zorgen dat alles soepel blijft draaien.

Verder maken we gebruik van het volgende voor het project:

**Sprints:** waar wij elke **twee weken** een demo opleveren. Deze demo gaat hand in hand met de werkprincipes van **DevOps**.

**Unit tests en build tests:** Docker build images en geautomatiseerde testen.

**CI/CD-pipelines:** Git branches die automatisch worden getest en gemerged als ze de testen met succes voltooien

**SL2B** voldoet ook aan alle eisen om gehost te worden op de Kubernetes cluster en we kunnen het project monitoren qua performances.

Dit zijn de algemene redenen die ons hebben overtuigd om **SL2B** te gebruiken voor het **DevOps** traject.

**DevOps**

1. Plan: Trello board
2. Code: Github
3. Build: Docker
4. Test: C# unit test + Docker unit test
5. Release: Kubernetes cluster
6. Deploy: Docker
7. Operate: Kubernetes cluster
8. Monitor: Grafana

A diagram of software development

Description automatically generated

**Praktijken & Technieken**

1. **Plan: Trello board**

Wij hebben ervoor gekozen om Trello te gebruiken omdat het een gebruiksvriendelijke gebruikersinterface heeft en we er veel meer ervaring mee hebben vergeleken met Azure. Het geeft ons de vrijheid om de board te designen naar onze keuze. Hierdoor is de trelloboard zelf overzichtelijk en fijn om mee te werken.

1. **Code: Github**

Ons team heeft al veel ervaring met GitHub, dus de keuze was snel gemaakt om ermee te gaan werken. Het zorgt ervoor dat wij branches kunnen maken en hierdoor continu aan het project kunnen werken zonder dat wij elkaar in de weglopen.

1. **Build: Docker**

Docker is een tool die ervoor zorgt dat wij onze applicatie op elke operating system kunnen runnen.

1. **Test: C# unit test + Docker unit test**

C# unit test maakt gebruik van de ASP.NET core systeem, wat ervoor zorgt dat er al zelf veel testen aanwezig zijn en wij dit kunnen gebruiken voor het automatiseren van onze testen.

Docker unit test zorgt ervoor dat wij onze builds kunnen testen en configureren voor elke operating systeem.

1. **Release: Kubernetes cluster**

In ons traject wordt er van ons verwacht dat wij ons project releasen en operate op de Kubernetes cluster die aanwezig zou zijn tijdens onze lessen.

1. **Deploy: Docker**

We maken gebruik van docker om onze build image te creëren en aan de hand van deze image het project te operaten.

1. **Operate: Kubernetes cluster**

In ons traject wordt er van ons verwacht dat wij ons project releasen en operate op de Kubernetes cluster die aanwezig zou zijn tijdens onze lessen.

1. **Monitor: Grafana**

We hebben onderzoek gedaan voor monitoring en hieruit kwam Grafana omdat het opensource is en het ons een mogelijkheid geeft om ene dashboard te creëren.

**Aandachtspunten & Verbeterstratagieën**

1. Overzichtelijker maken van epics en user story's. Voorheen wisten we niet perfect hoe je epics en user story's moest maken/koppelen, waardoor het onoverzichtelijk was welke user story bij welke epic hoorde. Verder wisten we niet precies wanneer de user story helemaal klaar zou zijn, omdat we niet veel acceptatiecriteria hadden genoteerd.

**Oplossing:** Elke epic heeft een nummer en bijhorende user story met daarbij ook een nummer. De branches worden bijvoorbeeld vernoemd ”1.2-inlog-systeem”. Verder wordt er voor elke user story acceptatiecriteria gemaakt, zodat de persoon die eraan werkt weet wanneer het af zou zijn.

1. Er wordt gelet op uitstelgedrag. Voorheen hadden we in het begin moeite om meteen aan de eindopdrachten te werken, wat aan het einde van de periode tot veel werk en veel stress leidde.

**Oplossing**: We gaan het verminderen doormiddel van Sprints, Daily stand-ups en een planning bijhouden.

1. Er wordt gelet op taakverdeling. Voorheen hadden we moeite met het verdelen van taken, waardoor één persoon veel meer deed dan de rest.

**Oplossing:** We letten op dat iedereen aan het werk is en actief meedoet aan elke fase.

1. Er zal meer onderzoek worden gedaan naar de concepten van DevOps zodat wij een beter begrip krijgen over de verschillende fases van DevOps en wellicht verandering van de tools die wij gebruiken in het project.

**Toelichting:** Voor vrijwel iedereen in ons projectgroepje zijn de devops-concepten nieuw, dit betekent dat wij als groepje gezamenlijk nog veel onderzoek moeten doen om erachter te komen hoe wij het willen aanpakken voor ons project.

**Actieplan**

**1. Structuur epics en user story’s**

Actie 1.1: Nummeringssysteem implementeren. Elk epic en user story krijgt een uniek nummer. Voorbeeld: Epic 1 kan hebben User Stories 1.1, 1.2, enz.

Actie 1.2: Standaard branch-namen aanhouden. Gebruik het nummeringssysteem in de naam van de branches om duidelijkheid en vindbaarheid te verbeteren, zoals 1.2-inlog-systeem.

Actie 1.3: Acceptatiecriteria definiëren. Duidelijke en meetbare acceptatiecriteria voor elke user story op te stellen, zodat het duidelijk is wanneer een taak als voltooid beschouwd kan worden.

**2. Verminderen van uitstelgedrag**

Actie 2.1: Sprints inplannen. Een sprintduur van twee weken aanhouden en strikt aan de geplande sprint reviews en planningssessies houden. Er worden daar o.a. verantwoordelijken toegewezen, deadlines vastgesteld en eventuele aanpassingen gemaakt.

Actie 2.2: Dagelijkse stand-ups implementeren. Elke schooldag een korte stand-up meeting houden om de voortgang te bespreken en problemen/valkuilen snel aan te pakken.

Actie 2.3: Voortgang visueel bijhouden. Gebruik maken van Trello board om taken te visualiseren en de status van elk item bij te werken.

**3. Verbetering van taakverdeling**

Actie 3.1: Rollen en verantwoordelijkheden definiëren. Elk teamlid moet duidelijke verantwoordelijkheden hebben en deze evenwichtig verdeeld moeten zijn.

**4. Verdieping in DevOps concepten**

Actie 4.1: Gezamenlijke leersessies organiseren. Regelmatige sessies plannen waarin teamleden onderzoek doen en hun resultaten over DevOps concepten delen.

Actie 4.2: Feedback en evaluatie van gebruikte tools.