

# SCRUM Projectplan INNER

## Project

### Inleiding

Het INNER-project richt zich op het ontwikkelen van een applicatie waarmee gebruikers de kwaliteit van EV-batterijen kunnen controleren door middel van röntgenbeelden. Hierbij wordt de gebruiker ondersteund door een Machine Learning (ML)-model dat fouten kan herkennen en de gebruiker kan helpen bij het sneller en nauwkeuriger beoordelen van de batterijstatus. Het project volgt de Scrum-methodologie, met een flexibele en iteratieve aanpak, om de gestelde doelen efficiënt te behalen.

---

## 1. Onderzoeksvragen

### Hoofdvraag

- Hoe kan de kwaliteit van een EV-batterij gecontroleerd worden door een gebruiker, geholpen door ML?

### Deelvragen

- Hoe kan een ML-model ontwikkeld worden om fouten te herkennen?
  - Hoe kan een gebruiker aangeven waar hij/zij fouten ziet in de batterij?
  - Hoe kan data over eerdere beoordelingen gebruikt worden om toekomstige controles te verbeteren?
  - Hoe kunnen historische data gebruikt worden om de ML-algoritmen te trainen?
  - Hoe kan de gebruikersinterface intuïtief ontworpen worden voor snelle interactie?
  - Hoe kunnen verschillende types schade aan batterijen automatisch worden herkend door het ML-model?
  - Wat zijn de technische eisen voor het opslaan van grote röntgenbeeldbestanden?
  - Hoe kan de nauwkeurigheid van het ML-model worden gemeten en verbeterd?
  - Hoe kunnen de resultaten van de batterijtest visueel worden weergegeven aan de gebruiker?
  - Welke beveiligingsmaatregelen moeten worden genomen om de gegevens van de batterijen te beschermen?
- 

## 2. Doel van het project

Het doel van dit project is om een intuïtieve, gebruikersvriendelijke applicatie te bouwen waarmee gebruikers röntgenbeelden van EV-batterijen kunnen beoordelen. Het ML-model leert van eerdere beoordelingen en helpt gebruikers door automatische foutsuggesties te doen. Deze functionaliteit moet bijdragen aan een snellere en accuratere beoordeling van batterijen.

---

# 3. User Story's / Requirements

## Gebruikers (User Stories)

1. Als gebruiker wil ik een eenvoudig te gebruiken interface hebben, zodat ik snel röntgenbeelden kan beoordelen.
2. Als gebruiker wil ik bevindingen op de batterijbeelden kunnen markeren, zodat ik fouten duidelijk kan identificeren.
3. Als gebruiker wil ik toegang hebben tot specificaties van de batterij, zoals type, leeftijd en schade.
4. Als gebruiker wil ik mijn bevindingen kunnen opslaan en later hervatten.
5. Als gebruiker wil ik verschillende kleurenmarkeringen kunnen gebruiken voor verschillende soorten fouten.
6. Als gebruiker wil ik een waarschuwing krijgen wanneer het ML-model een hoog risico op batterijschade detecteert, zodat ik extra aandacht kan besteden aan deze beoordeling.
7. Als gebruiker wil ik dat het systeem automatisch een samenvatting genereert van de meest kritieke fouten in de batterij, zodat ik snel een overzicht heb van de belangrijkste problemen.
8. Als gebruiker wil ik een rapport kunnen genereren en opleveren aan de eigenaar van de batterij, zodat deze de bevindingen kan inzien en opvolgen.
9. Als gebruiker wil ik röntgenbeelden kunnen roteren en inzoomen voor een beter perspectief.
10. Als gebruiker wil ik fouten kunnen corrigeren als het ML-model onjuiste suggesties doet.
11. Als gebruiker wil ik de mogelijkheid hebben om snel door eerdere beoordelingen te zoeken en fouten te vergelijken.

## Developers (Technical Requirements)

1. Als ontwikkelaar wil ik een schaalbare database opzetten voor opslag van röntgenbeelden en batterijdata.
  2. Als ontwikkelaar wil ik dat de applicatie foutloos draait op zowel desktops als tablets, waarbij een mobiele telefoon toegang krijgt tot de rapporten, maar niet de beoordelingsfunctionaliteit.
  3. Als ontwikkelaar wil ik de API efficiënt ontwerpen om snelle communicatie tussen frontend en backend te garanderen.
  4. Als ontwikkelaar wil ik ervoor zorgen dat de gegevens van gebruikers veilig en versleuteld worden opgeslagen.
  5. Als ontwikkelaar wil ik ervoor zorgen dat het ML-model regelmatig getraind wordt met nieuwe data om zijn nauwkeurigheid te verbeteren.
  6. Als ontwikkelaar wil ik het systeem zodanig opzetten dat het meerdere batterijmodellen kan ondersteunen.
  7. Als ontwikkelaar wil ik een logging systeem implementeren om fouten snel te kunnen identificeren.
  8. Als ontwikkelaar wil ik zorgen voor een modulaire codebasis om toekomstige updates eenvoudig te maken.
  9. Als ontwikkelaar wil ik geautomatiseerde tests draaien na elke code commit om bugs vroegtijdig te detecteren.
  10. Als ontwikkelaar wil ik zorgen voor snelle laadtijden van röntgenbeelden, zelfs bij lage internetverbindingen.
  11. Als ontwikkelaar wil ik de mogelijkheid hebben om op elke computer te kunnen ontwikkelen, zodat het team flexibel kan werken.
  12. Als ontwikkelaar wil ik dat mijn builds door een goed ingerichte CI/CD-pijplijn beoordeeld en gedeployed worden, zodat de kwaliteit van de code wordt gewaarborgd en het proces efficiënt verloopt.
-

## 4. Backlog

### Epic 1: Ontwikkeling van de gebruikersinterface voor het beoordelen van batterijen

1. Als gebruiker wil ik een eenvoudig te gebruiken interface hebben, zodat ik snel röntgenbeelden kan beoordelen.
2. Als gebruiker wil ik röntgenbeelden kunnen roteren en inzoomen voor een beter perspectief.
3. Als gebruiker wil ik verschillende kleurenmarkeringen kunnen gebruiken voor verschillende soorten fouten.
4. Als gebruiker wil ik bevindingen op de batterijbeelden kunnen markeren, zodat ik fouten duidelijk kan identificeren.
5. Als gebruiker wil ik mijn sessie kunnen opslaan en later hervatten.

### Epic 2: Integratie van het ML-model voor foutdetectie

1. Als gebruiker wil ik een waarschuwing krijgen wanneer het ML-model een hoog risico op batterijschade detecteert, zodat ik extra aandacht kan besteden aan deze beoordeling.
2. Als gebruiker wil ik dat het systeem automatisch een samenvatting genereert van de meest kritieke fouten in de batterij, zodat ik snel een overzicht heb van de belangrijkste problemen.
3. Als gebruiker wil ik dat het ML-model automatisch fouten herkent en suggesties doet voor reparatie of vervanging.
4. Als ontwikkelaar wil ik ervoor zorgen dat het ML-model regelmatig getraind wordt met nieuwe data om zijn nauwkeurigheid te verbeteren.
5. Als ontwikkelaar wil ik het systeem zodanig opzetten dat het meerdere batterijmodellen kan ondersteunen.

### Epic 3: Opslag en verwerking van batterijdata en röntgenbeelden

1. Als ontwikkelaar wil ik een schaalbare database opzetten voor opslag van röntgenbeelden en batterijdata.
2. Als gebruiker wil ik toegang hebben tot specificaties van de batterij, zoals type, leeftijd en schade.
3. Als gebruiker wil ik batterijdata kunnen exporteren naar PDF voor verdere analyse.
4. Als gebruiker wil ik kunnen filteren op specifieke fouten in eerdere batterijenrapporten.
5. Als ontwikkelaar wil ik ervoor zorgen dat de gegevens van gebruikers veilig en versleuteld worden opgeslagen.

### Epic 4: Samenwerkingstool voor gebruikers en rapportagefunctionaliteit

1. Als gebruiker wil ik een rapport kunnen genereren en opleveren aan de eigenaar van de batterij, zodat deze de bevindingen kan inzien en opvolgen.
2. Als gebruiker wil ik een overzicht van de batterijtesten kunnen zien, inclusief een samenvatting van de resultaten.
3. Als gebruiker wil ik automatisch meldingen ontvangen als nieuwe data beschikbaar is voor evaluatie.
4. Als gebruiker wil ik een geschiedenis van mijn eerdere beoordelingen inzien voor referentie.

5. Als gebruiker wil ik fouten kunnen corrigeren als het ML-model onjuiste suggesties doet.

## Losstaande user stories

1. Als gebruiker wil ik de mogelijkheid hebben om snel door eerdere beoordelingen te zoeken en fouten te vergelijken.
  2. Als gebruiker wil ik dat mijn vorige foutmeldingen automatisch gekoppeld worden aan batterijen met dezelfde ID, zodat ik geen dubbele meldingen krijg.
  3. Als gebruiker wil ik de mogelijkheid hebben om meerdere röntgenbeelden tegelijk te evalueren, zodat ik sneller meerdere batterijen kan beoordelen.
  4. Als gebruiker wil ik dat het systeem automatisch batterijen sorteert op basis van urgentie, zodat ik weet welke batterijen het meest aandacht behoeven.
  5. Als gebruiker wil ik notificaties ontvangen wanneer een nieuwe update van de applicatie beschikbaar is, zodat ik altijd met de laatste versie werk.
- 

## 5. Scrum Team

### Scrum Team Rollen:

- **Product Owner:** Wally van de Pas
  - **Scrum Master:** Bart van Hoven
  - **Developers:** Alle teamleden
  - **UX/UI Design:** Bart van Hoven en Stefan Jacobs
  - **DevOps:** Wally van de Pas en Piotr Tadralla
  - **ML-team:** Piotr Tadralla en Donovan Khoun
- 

## 6. Sprints en Planning

### Sprint Planning:

#### 1. Sprint 1 (2 weken):

- **Doelen:** Ontwerpen van de gebruikersinterface en opzetten van de database.
- **Verantwoordelijken:** UX/UI Designer voor de interface; Developers en DevOps Engineer voor de database.
- **Deliverables:** Werkende mock-up van de interface, eerste versie van de database.

#### 2. Sprint 2 (3 weken):

- **Doelen:** Basisintegratie van het ML-model voor  
foutdetectie en koppeling aan de database.

- **Verantwoordelijken:** Developers en ML-team.
- **Deliverables:** Werkend ML-model dat foutdetectie uitvoert op testdata en opslaat in de database.

### 3. Sprint 3 (3 weken):

- **Doelen:** Implementeren van foutmarkeringen in de interface en de communicatie tussen frontend en backend.
- **Verantwoordelijken:** UX/UI Designer en Developers.
- **Deliverables:** Gebruiker kan fouten markeren op de röntgenbeelden en deze worden in de database opgeslagen.

### 4. Sprint 4 (3 weken):

- **Doelen:** Beveiliging en optimalisatie van de applicatie, inclusief testen en eindgebruikersrapportage.
- **Verantwoordelijken:** Developers, DevOps Engineer en QA/Tester.
- **Deliverables:** Volledig werkende en beveiligde applicatie met rapportagefunctionaliteit voor gebruikers.

## Realistische tijdlijn voor 5 studenten

- **Week 1-2 (Sprint 1):** Focus op ontwerp en eerste infrastructuur (interface en database).
- **Week 3-5 (Sprint 2):** Start met ML-integratie en basisfunctionaliteiten.
- **Week 6-8 (Sprint 3):** Verbetering van gebruikersfunctionaliteit en foutmarkeringen.
- **Week 9-11 (Sprint 4):** Optimalisatie, beveiliging en laatste tests voor afronding van het project.

---

## 7. Definitie van Done

1. De gebruikersinterface is intuïtief en werkt vlekkeloos op alle apparaten.
2. Het ML-model kan batterijfouten met een nauwkeurigheid van 95% detecteren.
3. Gebruikers kunnen bevindingen markeren, rapporten genereren en fouten handmatig corrigeren.
4. Röntgenbeelden laden binnen 2 seconden, zelfs bij lage bandbreedte.
5. Alle gebruikersdata wordt veilig opgeslagen en voldoet aan de wettelijke privacyvereisten (bijv. GDPR).
6. De applicatie kan probleemloos honderden batterijen en röntgenbeelden opslaan en verwerken.
7. Alle functies zijn uitgebreid getest en goedgekeurd door het QA-team.
8. Het systeem genereert foutloze rapporten met een duidelijk overzicht van alle batterijtesten.
9. Gebruikers kunnen eenvoudig feedback geven en eerdere beoordelingen inzien.
10. De applicatie is volledig gedocumenteerd en klaar voor productie.

---

## Conclusie

Dit projectplan biedt een realistisch overzicht van de onderzoeksvragen, user stories, sprints en verantwoordelijkheden binnen het INNER-project. Met vier goed geplande sprints van in totaal 11 weken en een team van 5 studenten, kan de applicatie stapsgewijs ontwikkeld worden met een focus op gebruiksvriendelijkheid en schaalbaarheid, ondersteund door machine learning.