

# Scrum-aanpak voor het project "Slimme Serre"

---

## Teamleden

- Leerling 1: **Woud De Waegenaere** – Verantwoordelijk voor de Sensor Klasse
- Leerling 2: **Lars Ysebaert** – Verantwoordelijk voor de Actuator Klasse
- Leerling 3: **Vincent Junior Goemaere** – Verantwoordelijk voor de Tuinbeheer Klasse

## Scrum-werkwijze

Voor het project "**Slimme tuin**" werken wij volgens de Scrum-methode, een flexibele manier van samenwerken met korte iteraties, duidelijke communicatie en frequente evaluaties.

## Weekly Stand-ups

Elke week houden we een korte Scrum stand-up meeting (10–15 minuten) waarin we de volgende drie vragen beantwoorden:

1. Wat heb ik gedaan sinds de vorige stand-up?
2. Wat ga ik doen tot de volgende stand-up?
3. Waar heb ik problemen
4. Code van elkaar overlopen en verbeteren

Deze stand-ups helpen ons de voortgang te bewaken en obstakels vroeg te signaleren.

## Scrum-bord

We gebruiken een digitaal Scrum-bord in **Monday** met drie kolommen:

1.  To Do
  - Hier plaatsen we alle taken die nog moeten worden gestart.
2.  In Progress
  - Taken waar we momenteel aan werken.
3.  Done
  - Voltooide en gecontroleerde taken.

Voorbeeld taken:

- - Sensor klasse ontwerpen en testen
- Actuator klasse programmeren (LED & pomp)
- Logica schrijven voor tuinbeheer
- Interface bouwen voor datavisualisatie
- GitHub repository structureren en onderhouden
- Tests uitvoeren en verbeteren

## Tools

- GitHub: Voor versiebeheer, samenwerking en code review.
- Iedereen werkt op een eigen branch en maakt pull requests.
- Regelmatig committen en duidelijke commit messages zijn verplicht.
- Trello / GitHub Projects: Voor het beheren van ons Scrum-bord.
- (Optioneel) Google Drive / Docs: Voor documentatie.

## Verantwoordelijkheden

- Iedere leerling schrijft één of meerdere klassen en documenteert zijn werk goed.
- Samenwerking gebeurt via regelmatige communicatie, stand-ups en het Scrum-bord.
- De interface en integratie worden gezamenlijk aangepakt na afronding van de basisfunctionaliteiten.

## Iteraties

We werken in korte iteraties waarbij we na elke fase:

- Testen of de code werkt
- Feedback geven aan elkaar
- Aanpassingen doorvoeren indien nodig

## Conclusie

Door Scrum toe te passen werken we gestructureerd, met goede samenwerking en zicht op de voortgang. Dit zorgt ervoor dat we het project "Slimme Serre" efficiënt en in teamverband succesvol kunnen uitvoeren.

## **Product backlog:**

De Product Backlog bevat alle taken die nodig zijn om het slimme monitorsysteem te realiseren. Voorbeelden van backlog-items zijn:

<b>Nr.</b>	<b>User Story (taak)</b>	<b>Prioriteit</b>	<b>Verantwoordelijke</b>
1	Als gebruiker wil ik dat de sensor temperatuur, luchtvochtigheid en bodemvochtigheid meet, zodat ik data kan verzamelen.	Hoog	Leerling 1 (Sensor)
2	Als systeembeheerder wil ik dat de waterpomp automatisch geactiveerd wordt bij een droge bodem, zodat planten niet uitdrogen.	Hoog	Leerling 2 (Actuator)
3	Als gebruiker wil ik feedback krijgen via een LED als de pomp actief is, zodat ik zie dat het systeem werkt.	Midden	Leerling 2 (Actuator)
4	Als systeem wil ik data van sensoren ophalen en hier beslissingen op baseren (bijv. wanneer de pomp aan moet), zodat het automatisch werkt.	Hoog	Leerling 3 (Tuinbeheer)
5	Als gebruiker wil ik een eenvoudige interface zien waar ik de actuele waardes van de tuin kan volgen, zodat ik inzicht heb in de status.	Midden	Gezamelijk
6	Als ontwikkelaar wil ik dat de code in GitHub netjes verdeeld is over branches, zodat we veilig kunnen samenwerken.	Hoog	Iedereen
7	Als team willen we wekelijks stand-ups houden om de voortgang te bespreken, zodat we kunnen bijsturen waar nodig.	Midden	Iedereen
8	Als ontwikkelaar wil ik dat er tests zijn voor sensor en pomp, zodat ik zeker weet dat alles werkt.	Laag	Iedereen
9	Als team willen we de logica iteratief testen en verbeteren, zodat de slimme serre betrouwbaar werkt.	Midden	Leerling 3

## **Benodigdheden:**

- **Woud:**
  - o Temperatuursensor
  - o Luchtvochtigheidssensor
  - o Bodemvochtigheidssensor
- **Lars:**
  - o Waterpomp
  - o Waarschuwing LED/LCD

## **To Do:**

- **Woud:**
  - o Temperatuursensor programmeren
  - o Luchtvochtigheidssensor programmeren
  - o Bodemvochtigheidssensor programmeren
  - o Waarden van alle sensoren kunnen uitlezen en kunnen omzetten naar leesbare en bruikbare data
- **Lars:**
  - o Waterpomp programmeren
  - o Waarschuwing LEDs programmeren
- **Junior:**
  - o Communicatie tussen sensoren en actuators programmeren
  - o Drempelacties programmeren
  - o Interface
  - o Github beheren

## Sprintplanning

### Week 1 (13/05):

- **Lars:** motor, leds aanschakelen
- **Vincent:** algemeen alles voor de main + documentatie
- **Woud:** sensoren proberen uitlezen

### Week 2 (20/05):

- **Vincent:** Visuele interface, webpagina grotendeels
- **Lars:** DC motor en code

### Week 3 (27/05):

- **Lars:** led aansturen, actuator klasse afwerken
  - Problemen motor en transistor
- **Vincent:** Webpagina en code samenkoppelen, main code, en afspreken functies
- **Woud:** Temperatuur en bodemvochtigheidssensor werkende krijgen

### Week 4 (3/06):

- **Lars:** Code finetunen en becommentariëren, schema en berekeningen,
- **Vincent:** Main afgewerkt, functies afgesproken, website op ESP32 met sensorwaarden afwerken. Flowchart
- **Woud:** bodemsensor afwerken
- **Lars en Woud:** schakelingen toevoegen