

# Op weg naar HBO-ICT beroepstaken workshop

2023\_04\_06

.. voor S2

Teun: is een beroepstaak iets wat je in het bedrijfsleven zou kunnen doen?

hbo-I verhaal:

## Architectuurlagen:

- Gebruikersinteractie
- Organisatieprocessen
- Infrastructuur
- Software
- Hardware Interfacing

Dat alles kan in 4 Niveaus.

In S1 tm S3 dekken we in ieder geval Niveau 2 af.

Aan het eind van de studie delen van Niveau 3.

## Activiteiten:

- Analyseren
- Adviseren
- Ontwerpen
- Realiseren
- Manage & Control

## Voorbeeld: Hardware Interfacing:

- Niveau 1: Software schrijven voor een eenvoudig gegeven computersysteem, voorzien van sensoren en actuatoren.
- Niveau 2: Inrichten van eenvoudig computersysteem en via software de koppelingen met hardwarecomponenten realiseren.  
of  
Schrijven en testen van applicatiedriversoftware  
of  
Implementeren en testen van een protocol
- Niveau 3: Realiseren van een compleet computersysteem inclusief netwerk, hardware en systeemsoftware.  
of  
Opstellen en uitvoeren van een acceptatieprocedure, bijv in een virtuele omgeving, inclusief aspecten als timing, resourcegebruik en performance.
- Niveau 4: Realiseren van een compleet computersysteem waarbij gebruik gemaakt wordt van hardware synthese (vhdl) of artificial intelligence.

of

Realiseren van ...

**Verder zijn er nog professional skills:**

- Onderzoekend vermogen
- Zelflerend vermogen
- Organiserend vermogen
- Interactief vermogen

## Suggestie van Huib voor S2

Spelen met de hardware en leuke/nuttige dingen maken.

**TODO: herformuleren naar wat ze in de praktijk, in het bedrijfsleven zouden moeten kunnen doen als ze na het eerste jaar het bedrijfsleven in gaan (max 50 woorden).**

Mijn invulling:

- Gewenste functionaliteit **communiceren** met zijn opdrachtgever, **specificatie** opstellen en goed **gedocumenteerd** en **beargumenteerd** uitwerken. In de opdracht **combineert** de student typisch verschillende bestaande **deelsystemen**, zoals microcontrollers, sensoren, actuatoren en webserver tot een werkend geheel. Ook het schrijven van een **driver** hoort daarbij.

Andere keywords die genoemd werden:

- IOT, REST-API
- Binnen team over casus nadenken. Communiceren met stakeholders.
- Beperkte set software en hardware
- Kiezen uit meerdere microprocessors
- Testen op de functionele specificaties
- Onderzoek
- Gangbare ontwikkelmethoden
- Samenwerken
- Creatief
- Software
- Onder begeleiding werken
- Presenteren
- Testen
- Robot

Handig om met zijn alle met wordcloud te spelen: [whiteboard.office.com](https://whiteboard.office.com)

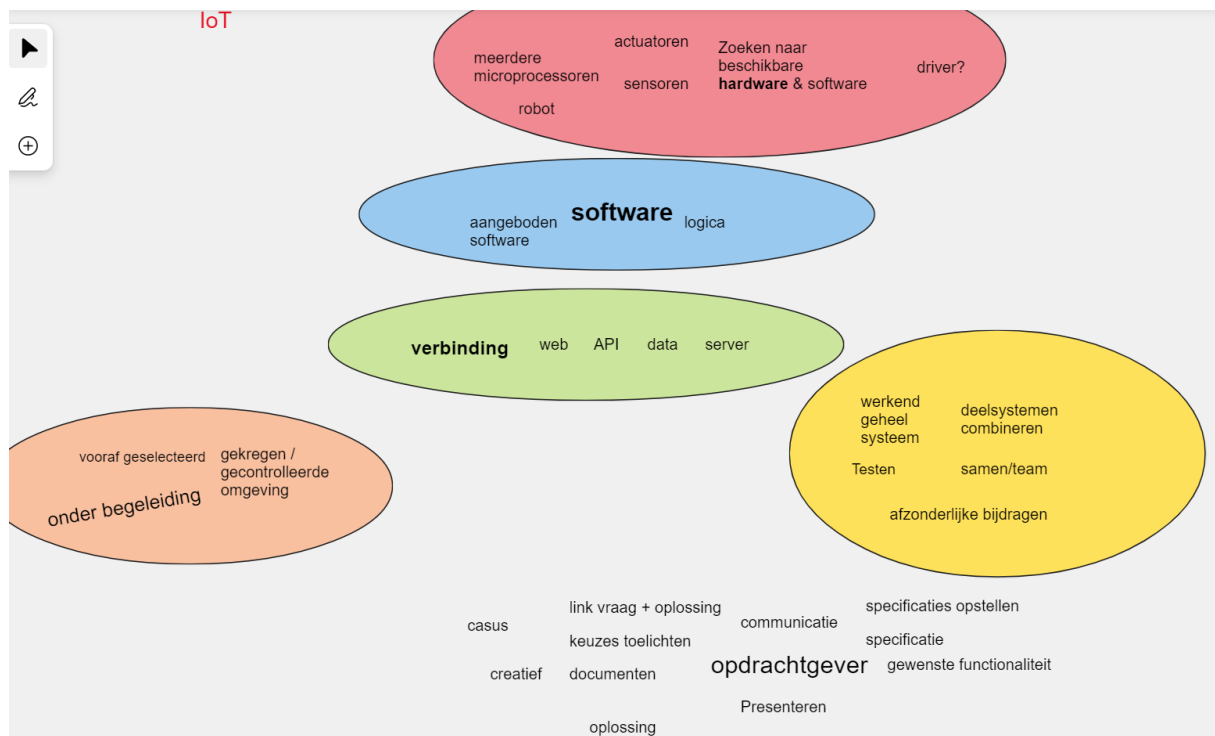
.. hoewel: de boards lijken compleet out of sync – werkt niet goed, zo lijkt het, je kunt niet handig mind-maps maken of (aanpasbare-) lijntjes tekenen.

Miro schijnt fijner te werken.

Groeperen van de word-cloud.

De gegroepeerde woorden kun je per groep vertalen naar een of meerdere leeruitkomsten.

(tussendoor: eerste jaar: opdrachtgever en begeleiding intern. Tweede jaar: opdrachtgever extern, begeleiding intern. Derde jaar: opdrachtgever en begeleiding extern)



**De beroepstaken die nu op het bord uit de groepsdiscussie kristalliseren, zijn:**

1. De student stemt samen met een team met een opdrachtgever de functionaliteit en de hoofdlijnen van de oplossing af.
2. De student integreert bestaande componenten tot een (extern-) connected embeded systeem d.m.v. software, realiseert de gewenste functionaliteit en test volgens een vooraf gespecificeerde testcase.
3. De student houdt de opdrachtgever en teamleden op de hoogte van de voortgang en neemt de feedback mee in de verdere ontwikkeling.
4. De student levert samen met het team de oplossing op aan de opdrachtgever.

Denk bij het laatste punt aan: git repo overdragen, overdrachtscommunicatie schrijven.

Verdere aantekeningen:

- Soorten testen benoemen in de leeruitkomsten

## Vervolg (2023-04-12): Op weg naar de leeruitkomsten

NB: er is een hierarchy: top level = de beroepstaken, wat ze in het bedrijfsleven doen.

Volgende niveau = leeruitkomsten: waarop gaan we toetsen – in algemenere zin.

Volgende niveau = leerdoelen: nog specifiek geformuleerd en concreter wat ze moeten kunnen.

In leeruitkomsten: I's : inhoudelijke leeruitkomsten en V's : vaardigheden (zoals professional skills)

Potentiele leeruitkomsten:

De student...

- Toont aan te kunnen samenwerken in een team binnen context van opdracht
  - Afstemmen wie wat doet

- Elkaar feedback geven / vragen
- Elkaar op dingen aanspreken
- Elkaar helpen
- Verantwoorelijkheid nemen / proactief
- Versiebeheer
- Planborden

NB: de sub-bullets specificeren de zg “indicatoren” van de leeruitkomst. De aanwezigheid van indicatoren maakt een eenvoudiger beschrijving van de leeruitkomst op.

De indicatoren leveren in het algemeen een of meerdere leerdoelen op.

Intermezzo: Arno: laat ze eerst tegen problemen aanlopen, en leer ze dan de oplossing.

- Schrijft toekomstbestendige code
  - Gestructureerd
  - Probleem decompositie
  - Interface aansluiten
  - Duurzaamheid van de oplossing
  - Toekomstbestendigheid
  - Modulair
  - Herbruikbaarheid
- Deelt de functionaliteit op en toont dit resultaat in een ontwerp van de oplossing.
- Student analyseert de gevraagde functionaliteit en hoofdlijnen van de oplossing, deelt deze op in verschillende hard-en- software componenten, en combineert deze tot een ontwerp.
- Student ontwikkelt voor een hardware component een software aansturing. (?)
- Student kan zelfstandig vanaf de grond software schrijven om een hardware component te gebruiken in een (eenvoudige) IOT oplossing (hulpmiddel? / indicator).
- Student toont aan dat is bijgedragen aan het combineren van verschillende hard- en software componenten tot een eenvoudige IOT oplossing. (Software / Hardware realiseren)
- Student richt samen met het team een beheersysteem voor softwareontwikkeling in en gebruikt deze voor voortgang, samenwerking en oplevering.
- Analooq voor Hardware interfacing, maar dan alleen gebruik.  
Vooral voor testen. (Hardware Managed Control 0+).  
(gebruik van ontwikkel en test platform voor hardware)
- Onderbouwing
  - Onderzoekend vermogens
- (Projectmatig) werken
  - Plannen van taken over de tijd
  - Doelstellingen? (waar werk je naar toe?) ... hoort meer bij ontwerp van hoofdlijnen, bij de docent in S2.
  - Incrementeel – under guidance
  - Uitleggen waar je mee bezig bent.
  - Zicht op proces (dus anderen laten zien waar je staat)
  - Check of op goede weg
  - Communicatie ->team, opdrachtgever, ??

- Documenteren
- Keuzes toelichten -> proces, inhoud
- Vastleggen van verkregen informatie.
- Samenwerken
- Reflectie
- Evaluatie

Van de rode delen is iffy of ze hierbij thuis horen.

Huiswerk: combineer groepen niet-rode sub-punten van “werken” tot 2,3 4 Leeruitkomsten.

**Misschien (cgpt, 3 stuks):**

### **Topic 1: Planning, doelen en taakverdeling**

- **Plannen van taken over de tijd**
- **Doelstellingen**
- **Incrementeel – onder guidance**

### **Topic 2: Communicatie, samenwerking en transparantie**

- **Uitleggen waar je mee bezig bent**
- **Zicht op proces (anderen laten zien waar je staat)**
- **Communicatie -> team, opdrachtgever**
- **Samenwerken**

### **Topic 3: Documentatie, reflectie en evaluatie**

- **Documenteren**
- **Keuzes toelichten -> proces, inhoud**
- **Vastleggen van verkregen informatie**
- **Reflectie**
- **Evaluatie**

Of met 2 stuks:

### **Topic 1: Planning, doelen en samenwerking**

- **Plannen van taken over de tijd**
- **Doelstellingen (waar werk je naar toe?)**
- **Incrementeel – onder guidance**
- **Samenwerken**
- **Check of op goede weg**

### **Topic 2: Communicatie, documentatie en evaluatie**

- **Uitleggen waar je mee bezig bent**

- Zicht op proces (anderen laten zien waar je staat)
- Communicatie -> team, opdrachtgever
- Documenteren
- Keuzes toelichten -> proces, inhoud
- Vastleggen van verkregen informatie
- Reflectie
- Evaluatie

2023-4-18

Eigen leeruitkomsten:

- De student kan plannen, doelen stellen en in taken opsplitsen.
- De student kan op transparante wijze communiceren.
- De student kan documenteren, reflecteren en evalueren.

Leeruitkomsten Jan Halsema:

- De student neemt **actief** de **verantwoordelijkheid** voor de uitkomst van de teamopdracht en de daarvoor benodigde onderdelen.
- De student houdt continue **overzicht** over de stand van het project dmv team **communicatie, plannen** over tijd en **versiebeheer**.
- De student **verbetert** zelf de **procesproducten** en het ontwikkelproces dmv peerreviews.
- De student **onderbouwt** de keuzes in documentatie ter ondersteuning van de opgeleverde producten.

Een leeruitkomst:

- Moet **toetsbaar** zijn.

Voor chat gpt4:

Ideeen voor leeruitkomsten voor procesmatige vaardigheden voor een informatica studie:

- De student neemt actief de verantwoordelijkheid voor de uitkomst van de teamopdracht en de daarvoor benodigde onderdelen.
- De student houdt continue overzicht over de stand van het project dmv team communicatie, plannen over tijd en versiebeheer.
- De student verbetert zelf de procesproducten en het ontwikkelproces dmv peerreviews.
- De student onderbouwt de keuzes in documentatie ter ondersteuning van de opgeleverde producten.
- De student kan plannen, doelen stellen en in taken opsplitsen.
- De student kan op transparante wijze communiceren.
- De student kan documenteren, reflecteren en evalueren.
- De student toont aan te kunnen samenwerken in een team binnen context van opdracht

Herformuleer het geheel van deze 8 leeruitkomsten tot 7 nieuwe leeruitkomsten van maximaal 20 woorden elk, die elkaar onderling zo min mogelijk overlappen.

Resultaat:

1. De student neemt verantwoordelijkheid voor teamopdrachten en bijbehorende onderdelen, met actieve betrokkenheid.
  2. De student behoudt overzicht in projecten door effectieve communicatie, planning en versiebeheer.
  3. De student verbetert procesproducten en ontwikkelprocessen via peerreviews en zelfreflectie.
  4. De student onderbouwt keuzes in documentatie ter ondersteuning van opgeleverde producten.
  5. De student plant, stelt doelen en deelt taken op in beheersbare stappen.
  6. De student communiceert transparant en werkt samen met teamleden binnen de opdrachtcontext.
  7. De student documenteert, reflecteert en evalueert prestaties voor continue verbetering en groei.
- 

Onderverdelen in kleine stukjes.

Daarna:

De student kan duur van taken (van de kleine stukjes) bepalen.

Vooraf maken van een dagplanning met taken en rapporteren over uitgevoerde taken en reflecteren op duur van de taken.

Gedrag: plannen van taken

Inhoud: duur van taken, dagplanning vooraf, rapporteer reflecteer achteraf.

Voorwaarden/context: dag. Rekening houdend met capaciteit van team en docenten.

Hulpmiddelen: vanuit een gegeven taak van een sprint.

Door Chat gpt samengevat in een leeruitkomst van 20 woorden:

- De student plant dagelijkse taken, rekening houdend met teamcapaciteit en docenten, en rapporteert en reflecteert achteraf.

Gedrag: (op tijd) komen

Inhoud: alle activiteiten

Voorwaarde: geen vooraf opgegeven geldige reden

Zowel leeruitkomsten t.a.v. inhoud als gedrag kunnen opnieuw geformuleerd worden vanuit bovenstaande opsplitsingen, en meer geconcretiseerd naar semester 2.

Richtlijn: max pakweg 10 leeruitkomsten.  
Probeer wel alle beroepstaken af te dekken.

Leerdoelen	Leeruitkomsten
Gewenste resultaten	Getoonde resultaten
Gekoppeld aan lessen	Gekoppeld aan toetsing
Specifiek onderdeel	Groter geheel / hele beroepstaak

### Huiswerk:

Bedenk wat je belangrijk vindt aan de reeds gevonden leeruitkomsten van I en V, Partitioneer ze zoals het voorbeeld van plannen en op tijd komen, en upload het naar het mapje "op weg naar leeruitkomsten" op teams.

Tip voor het vinden van common grounds met een groep: de consent-methode:

Neem nooit een beslissing zonder dat iedereen heeft gezegd "consent".

Betekent dat de betreffende persoon erachter kan staan. En dat voldoende is uitgesproken wat de bezwaren zijn.

Hallo, de volgende leeruitkomst:

- De student plant dagelijkse taken, rekening houdend met teamcapaciteit en docenten, en rapporteert en reflecteert achteraf.

Zou je (met enige fantasie) kunnen opsplitsen in "gedrag", "Inhoud", "voorwaarden/context", "hulpmiddelen", bijvoorbeeld zo:

Gedrag: plannen van taken

Inhoud: duur van taken, dagplanning vooraf, rapporteer reflecteer achteraf.

Voorwaarden/context: dag. Rekening houdend met capaciteit van team en docenten.

Hulpmiddelen: vanuit een gegeven taak van een sprint.

Kun jij hetzelfde doen voor de volgende leeruitkomsten:

1. De student neemt verantwoordelijkheid voor teamopdrachten en bijbehorende onderdelen, met actieve betrokkenheid.
2. De student behoudt overzicht in projecten door effectieve communicatie, planning en versiebeheer.
3. De student verbetert procesproducten en ontwikkelprocessen via peerreviews en zelfreflectie.
4. De student onderbouwt keuzes in documentatie ter ondersteuning van opgeleverde producten.
5. De student plant, stelt doelen en deelt taken op in beheersbare stappen.
6. De student communiceert transparant en werkt samen met teamleden binnen de opdrachtcontext.
7. De student documenteert, reflecteert en evalueert prestaties voor continue verbetering en groei.
8. Toont aan te kunnen samenwerken in een team binnen context van opdracht
9. Schrijft toekomstbestendige code
10. Deelt de functionaliteit op en toont dit resultaat in een ontwerp van de oplossing.
11. Student analyseert de gevraagde functionaliteit en hoofdlijnen van de oplossing, deelt deze op in verschillende hard-en- software componenten, en combineert deze tot een ontwerp.
12. Student ontwikkelt voor een hardware component een software aansturing. (?)



13. Student kan zelfstandig vanaf de grond software schrijven om een hardware component te gebruiken in een (eenvoudige) IOT oplossing (hulpmiddel? / indicator).
- 14 Student toont aan dat is bijgedragen aan het combineren van verschillende hard- en software componenten tot een eenvoudige IOT oplossing. (Software / Hardware realiseren)
15. Student richt samen met het team een beheersysteem voor softwareontwikkeling in en gebruikt deze voor voortgang, samenwerking en oplevering.
16. Analooq voor Hardware interfacing, maar dan alleen gebruik.

Hier zijn de opsplitsingen voor de 16 leeruitkomsten:

1. De student neemt verantwoordelijkheid voor teamopdrachten en bijbehorende onderdelen, met actieve betrokkenheid.
  - Gedrag: Verantwoordelijkheid nemen, actief betrokken zijn
  - Inhoud: Teamopdrachten, bijbehorende onderdelen
  - Voorwaarden/context: Teamverband
  - Hulpmiddelen: N.v.t.
2. De student behoudt overzicht in projecten door effectieve communicatie, planning en versiebeheer.
  - Gedrag: Communiceren, plannen, versiebeheer
  - Inhoud: Projecten, communicatie, planning
  - Voorwaarden/context: Projectverband
  - Hulpmiddelen: Versiebeheersystemen
3. De student verbetert procesproducten en ontwikkelprocessen via peerreviews en zelfreflectie.
  - Gedrag: Peerreviews uitvoeren, zelfreflectie
  - Inhoud: Procesproducten, ontwikkelprocessen
  - Voorwaarden/context: Projectverband, samenwerking
  - Hulpmiddelen: Review-tools
4. De student onderbouwt keuzes in documentatie ter ondersteuning van opgeleverde producten.
  - Gedrag: Documenteren, onderbouwen van keuzes
  - Inhoud: Documentatie, opgeleverde producten
  - Voorwaarden/context: Projectverband, oplevering
  - Hulpmiddelen: Documentatie-tools
5. De student plant, stelt doelen en deelt taken op in beheersbare stappen.
  - Gedrag: Plannen, doelen stellen, taken opdelen
  - Inhoud: Doelen, taken, stappen
  - Voorwaarden/context: Projectverband
  - Hulpmiddelen: Planning-tools
6. De student communiceert transparant en werkt samen met teamleden binnen de opdrachtcontext.
  - Gedrag: Communiceren, samenwerken
  - Inhoud: Transparante communicatie, opdrachtcontext
  - Voorwaarden/context: Teamverband
  - Hulpmiddelen: Communicatietools

7. De student documenteert, reflecteert en evalueert prestaties voor continue verbetering en groei.
  - Gedrag: Documenteren, reflecteren, evalueren
  - Inhoud: Prestaties, verbetering, groei
  - Voorwaarden/context: Individueel en teamverband
  - Hulpmiddelen: Documentatie- en evaluatietools
8. Toont aan te kunnen samenwerken in een team binnen context van opdracht
  - Gedrag: Samenwerken
  - Inhoud: Teamwerk
  - Voorwaarden/context: Opdrachtcontext
  - Hulpmiddelen: N.v.t.
9. Schrijft toekomstbestendige code
  - Gedrag: Coderen
  - Inhoud: Toekomstbestendige code
  - Voorwaarden/context: Softwareontwikkeling
  - Hulpmiddelen: Programmeertools
10. Deelt de functionaliteit op en toont dit resultaat in een ontwerp van de oplossing.
  - Gedrag: Functionele decompositie, ontwerpen
  - Inhoud: Functionaliteit, oplossingsontwerp
  - Voorwaarden/context: Projectverband
  - Hulpmiddelen: Ontwerptools
11. Student analyseert de gevraagde functionaliteit en hoofdlijnen van de oplossing, deelt deze op in verschillende hard-en- software componenten, en combineert deze tot een ontwerp.
  - Gedrag: Analyseren, decompositie, ontwerpen
  - Inhoud: Functionaliteit, hardware- en softwarecomponenten, ontwerp
  - Voorwaarden/context: Projectverband
  - Hulpmiddelen: Analyse- en ontwerptools
12. Student ontwikkelt voor een hardware component een software aansturing. (?)
  - Gedrag: Ontwikkelen
  - Inhoud: Hardware component, software aansturing
  - Voorwaarden/context: Projectverband, IoT-oplossing
  - Hulpmiddelen: Programmeertools
13. Student kan zelfstandig vanaf de grond software schrijven om een hardware component te gebruiken in een (eenvoudige) IOT oplossing (hulpmiddel? / indicator).
  - Gedrag: Zelfstandig software schrijven
  - Inhoud: Hardware component, IoT oplossing
  - Voorwaarden/context: Individueel, eenvoudige IoT-oplossing
  - Hulpmiddelen: Programmeertools

14. Student toont aan dat is bijgedragen aan het combineren van verschillende hard- en software componenten tot een eenvoudige IOT oplossing. (Software / Hardware realiseren)
- Gedrag: Combineren, bijdragen aan realisatie
  - Inhoud: Hardware- en softwarecomponenten, IoT oplossing
  - Voorwaarden/context: Projectverband, eenvoudige IoT-oplossing
  - Hulpmiddelen: Programmeer- en hardwaretools
15. Student richt samen met het team een beheersysteem voor softwareontwikkeling in en gebruikt deze voor voortgang, samenwerking en oplevering.
- Gedrag: Inrichten, gebruiken, samenwerken
  - Inhoud: Beheersysteem, softwareontwikkeling, voortgang, oplevering
  - Voorwaarden/context: Teamverband
  - Hulpmiddelen: Beheersystemen
16. Analooq voor Hardware interfacing, maar dan alleen gebruik.
- Gedrag: Hardware interfacing, gebruiken
  - Inhoud: Hardwarecomponenten
  - Voorwaarden/context: Projectverband
  - Hulpmiddelen: Hardware interfacing tools

Neem onderstaande leeruitkomsten alsjeblieft samen, en splits het vervolgens weer weer op in 8 zo intern zoveel mogelijk samenhangende maar onderling zo min mogelijk overlappende leeruitkomsten af, met maximaal 25 woorden per leeruitkomst:

Kun jij hetzelfde doen voor de volgende leeruitkomsten:

1. De student neemt verantwoordelijkheid voor teamopdrachten en bijbehorende onderdelen, met actieve betrokkenheid.
2. De student behoudt overzicht in projecten door effectieve communicatie, planning en versiebeheer.
3. De student verbetert procesproducten en ontwikkelprocessen via peerreviews en zelfreflectie.
4. De student onderbouwt keuzes in documentatie ter ondersteuning van opgeleverde producten.
5. De student plant, stelt doelen en deelt taken op in beheersbare stappen.
6. De student communiceert transparant en werkt samen met teamleden binnen de opdrachtcontext.
7. De student documenteert, reflecteert en evalueert prestaties voor continue verbetering en groei.
8. Toont aan te kunnen samenwerken in een team binnen context van opdracht
9. Schrijft toekomstbestendige code
10. Deelt de functionaliteit op en toont dit resultaat in een ontwerp van de oplossing.
11. Student analyseert de gevraagde functionaliteit en hoofdlijnen van de oplossing, deelt deze op in verschillende hard-en- software componenten, en combineert deze tot een ontwerp.
12. Student ontwikkelt voor een hardware component een software aansturing. (?)

13. Student kan zelfstandig vanaf de grond software schrijven om een hardware component te gebruiken in een (eenvoudige) IOT oplossing (hulpmiddel? / indicator).
- 14 Student toont aan dat is bijgedragen aan het combineren van verschillende hard- en software componenten tot een eenvoudige IOT oplossing. (Software / Hardware realiseren)
15. Student richt samen met het team een beheersysteem voor softwareontwikkeling in en gebruikt deze voor voortgang, samenwerking en oplevering.
16. Analooq voor Hardware interfacing, maar dan alleen gebruik.

1. De student neemt verantwoordelijkheid voor teamopdrachten, werkt actief samen en communiceert transparant met teamleden binnen de opdrachtcontext.
2. De student behoudt overzicht in projecten door effectieve communicatie, planning, versiebeheer en het inrichten van beheersystemen voor softwareontwikkeling en hardware interfacing.
3. De student verbetert procesproducten en ontwikkelprocessen via peerreviews, zelfreflectie en evaluatie van prestaties voor continue groei en ontwikkeling.
4. De student analyseert gevraagde functionaliteit, deelt deze op in hard- en softwarecomponenten en ontwerpt oplossingen op basis van deze analyse en onderbouwde keuzes.
5. De student plant, stelt doelen en deelt taken op in beheersbare stappen voor zowel hardware- als softwareontwikkeling, met aandacht voor het combineren van componenten.
6. De student schrijft toekomstbestendige code en ontwikkelt zelfstandig software aansturing voor hardwarecomponenten in het kader van eenvoudige IoT-oplossingen.
7. De student documenteert en onderbouwt keuzes ter ondersteuning van opgeleverde producten, zowel voor hardware- als softwarecomponenten, in het ontwerp.
8. De student toont aan te kunnen samenwerken in een team, past zelfstandig software toe voor hardwarecomponenten en evalueert het gebruik ervan binnen eenvoudige IoT-oplossingen.
9. De student draagt bij aan het combineren van verschillende hard- en softwarecomponenten tot eenvoudige IoT-oplossingen en reflecteert op zijn/haar bijdrage aan het project.

Beschouw de volgende leeruitkomst: • De student plant dagelijkse taken, rekening houdend met teamcapaciteit en docenten, en rapporteert en reflecteert achteraf. Deze leeruitkomst zou je (met enige fantasie) kunnen opsplitsen in "gedrag", "Inhoud", "voorwaarden/context", "hulpmiddelen", bijvoorbeeld zo: Gedrag: plannen van taken Inhoud: duur van taken, dagplanning vooraf, rapporteer reflecteer achteraf. Voorwaarden/context: dag. Rekening houdend met capaciteit van team en docenten. Hulpmiddelen: vanuit een gegeven taak van een sprint. Kun jij hetzelfde doen voor de volgende leeruitkomsten: (vervolgens de bovenstaande 9 gepaste)

#### Handmatige refactoring

1. Verantwoordelijkheid nemen voor teamopdrachten, actieve samenwerking en transparante communicatie
  - Gedrag: verantwoordelijkheid nemen, actief samenwerken, communiceren
  - Inhoud: teamopdrachten
  - Voorwaarden/context: opdrachtcontext
  - Hulpmiddelen: n.v.t.

2. Behouden van overzicht in projecten door effectieve communicatie, planning, versiebeheer en beheersystemen

- Gedrag: communiceren, plannen, beheersystemen inrichten
- Inhoud: versiebeheer, softwareontwikkeling, hardware interfacing
- Voorwaarden/context: projecten
- Hulpmiddelen: communicatiemiddelen, planningstools, versiebeheersystemen

3. Verbeteren van procesproducten en ontwikkelprocessen via peerreviews, zelfreflectie en evaluatie

- Gedrag: peerreviews uitvoeren, zelfreflectie, evalueren
- Inhoud: procesproducten, ontwikkelprocessen, prestaties
- Voorwaarden/context: continue groei en ontwikkeling
- Hulpmiddelen: feedbackformulieren, evaluatie-instrumenten

4. Analyseren van functionaliteit, opdelen in hard- en softwarecomponenten en ontwerpen van oplossingen

- Gedrag: analyseren, opdelen, ontwerpen
- Inhoud: functionaliteit, hard- en softwarecomponenten, oplossingen
- Voorwaarden/context: n.v.t.
- Hulpmiddelen: analysemethoden, ontwerpgereedschappen

5. Plannen, doelen stellen en taken opdelen in beheersbare stappen voor hardware- en softwareontwikkeling

- Gedrag: plannen, doelen stellen, taken opdelen
- Inhoud: hardware- en softwareontwikkeling, beheersbare stappen, combineren van componenten
- Voorwaarden/context: n.v.t.
- Hulpmiddelen: planningstools, taakbeheersystemen

6. Schrijven van toekomstbestendige code en ontwikkelen van software aansturing voor hardwarecomponenten

- Gedrag: code schrijven, software ontwikkelen
- Inhoud: toekomstbestendige code, hardwarecomponenten, IoT-oplossingen
- Voorwaarden/context: n.v.t.
- Hulpmiddelen: programmeertalen, ontwikkelomgevingen

7. Documenteren en onderbouwen van keuzes ter ondersteuning van producten in het ontwerp

1. Gedrag: documenteren, onderbouwen
2. Inhoud: keuzes, hardware- en softwarecomponenten, ontwerp
3. Voorwaarden/context: n.v.t.
4. Hulpmiddelen: documentatiegereedschappen

8. Samenwerken in een team, toepassen van software voor hardwarecomponenten en evalueren van het gebruik

- Gedrag: samenwerken, software toepassen, evalueren
- Inhoud: hardwarecomponenten, eenvoudige IoT-oplossingen
- Voorwaarden/context: teamverband
- Hulpmiddelen: softwaretools, evaluatie-instrumenten

9. Bijdragen aan het combineren van hard- en softwarecomponenten en reflecteren op eigen bijdrage aan het project

- Gedrag: bijdragen, combineren, reflecteren
- Inhoud: hard- en softwarecomponenten, eenvoudige IoT-oplossingen, projectbijdrage

- Voorwaarden/context: n.v.t.
- Hulpmiddelen: samenwerkingsplatforms, reflectie-instrumenten

#### Handmatige refactoring

1. Verantwoordelijkheid nemen voor teamopdrachten, actieve samenwerking en transparante communicatie
  - Gedrag: actief verantwoordelijkheid nemen, actief samenwerken, assertief communiceren, gezamenlijk plannen
  - Inhoud: teamprojecten
  - Voorwaarden: op komen dagen
  - Context: opdrachtcontext
  - Hulpmiddelen: scrum methode, trello, drawio, google docs
2. Plannen, doelen stellen en taken opdelen in beheersbare stappen voor hardware- en softwareontwikkeling
  - Gedrag: Plannen, doelen stellen, taken opdelen en prioriteren (evt duur plannen), beheersystemen inrichten
  - Inhoud: versiebeheer, softwareontwikkeling, hardware interfacing
  - Voorwaarden/context: zowel team als individuele projecten.
  - Hulpmiddelen: communicatiemiddelen, planningstools (drawio), versiebeheersystemen, journaliseren, scrum met punten-pokeren (voor team projecten)
3. Verbeteren van procesproducten en ontwikkelprocessen via peerreviews, zelfreflectie en evaluatie
  - Gedrag: peerreviews, zelfreflecties en evaluaties uitvoeren
  - Inhoud: procesproducten, ontwikkelprocessen, prestaties
  - Doel/context: continue groei en ontwikkeling
  - Hulpmiddelen: feedbackformulieren, evaluatie-instrumenten (canvas feedback), reflectieverslag, journaliseren
4. Analyseren van functionaliteit, opdelen in hard- en softwarecomponenten en ontwerpen van oplossingen
  - Gedrag: analyseren, opdelen, ontwerpen
  - Inhoud: functionaliteit, hard- en softwarecomponenten, oplossingen
  - Voorwaarden/context: projecten
  - Hulpmiddelen: analysemethoden (UML), ontwerp gereedschappen (drawio, pspice, fritzing)
5. Schrijven van toekomstbestendige code en ontwikkelen van software aansturing voor hardwarecomponenten
  - Gedrag: code schrijven, software ontwikkelen
  - Inhoud: toekomstbestendige code, hardwarecomponenten, IoT-oplossingen
  - Context: projecten
  - Voorwaarden: OO kennis, IoT kennis (Batterijen, Sleep, internet, REST)
  - Hulpmiddelen: programmeertalen (C++), ontwikkelomgevingen (hwlib), Visual Studio.

6. Documenteren en onderbouwen van keuzes ter ondersteuning van producten in het ontwerp

- Gedrag: documenteren, onderbouwen
- Inhoud: keuzes, hardware- en softwarecomponenten, ontwerp
- Voorwaarden/context: projecten
- Hulpmiddelen: documentatiegereedschappen (Word, Excel), journaliseren.

7. Professionele werkhouding

- Gedrag: (op tijd) komen, 40 uur per week aan de opleiding besteden.
- Inhoud: alle activiteiten
- Context: de opleiding is voltijds. Er kan bij wet niet worden ingekort op het aantal te besteden uren per SE.
- Voorwaarde: geen verstek zonder vooraf opgegeven geldige reden. Aan de huiswerkopdrachten is duidelijk de verwachte tijd besteed. Tracking van uren. Tracking van (peerreview-) resultaten van formatieve inleveringen. Feedback daarop (in eerste lijn via SLB?)

2023-05-09

Vaste structuur kd