

CURRICULUM

SOFTWARE ENGINEERING LOW-CODE

SOFTWARE ENGINEERING LOW-CODE

In het omscholingstraject Software Engineering low-code leer je softwaresystemen ontwerpen, realiseren en testen. Je werkt met verschillende programmeertalen en ontwikkelmethoden. Hiernaast leer je ook werken met een low-code software programma, waarin je met weinig code volwaardige software kunt ontwikkelen.

Je leert projectmatig te werken en je richt je op het beheren, analyseren, adviseren, ontwerpen en realiseren van softwaresystemen. In de ontwikkeling van nieuwe softwaresystemen speelt de gebruiker een steeds belangrijkere rol. Daarom ontwikkel je ook communicatieve vaardigheden om de wensen van de klant te kunnen achterhalen en die te vertalen in een gebruiksvriendelijk eindproduct. Het programma Software Engineering low-code heeft grote overeenkomsten met het programma Software Engineering. Het laatste project verschilt. Je bouwt een web-applicatie met behulp van een low-code systeem in plaats van de web-applicatie volledig te programmeren.

Profiel Software Engineer

Wereldwijd is er momenteel behoefte aan goed opgeleide IT-specialisten. Na het volgen van dit programma kun je aan de slag als Software Engineer in verschillende programmeertalen waaronder low-code.

OVERZICHT SOFTWARE ENGINEERING

De opleiding is ingedeeld in drie leerlijnen namelijk vakinhoud, persoonlijke vaardigheden en projecten. De vakinhoudelijke leerlijn bestaat uit modules. Het gehele curriculum bestaat uit 2.240 studie-uren. Deze studie-uren worden verdeeld over de opleidingsduur van elf maanden.

CURRICULUM

De omscholing tot Software Engineer low-code is gericht op het aanleren van de benodigde skills en theorie die nodig zijn bij het succesvol ontwerpen en programmeren van software. Software zoals apps, backend applicaties, websites en database-applicaties.

Het traject van vijf maanden scholing gecombineerd met de werkperiode van zes maanden bij de werkgever bereidt je voor op een diversiteit aan functies binnen het vakgebied Software Engineering. Voorbeelden van programmeertalen waar cursisten bij hun bedrijf software in ontwikkelen zijn Java, PHP, Mendix, Oracle en .Net. Ook werken cursisten aan SAP integratieprojecten en Hippo CMS-uitbreidingen.

CURRICULUM

Het onderwijsprogramma van Software Engineering low-code bestaat uit twee semesters.

- In het eerste semester (4 maanden) krijg je vakinhoudelijke thema's die relevant zijn voor het vakgebied Software Engineering. Na een basisfase met drie vakken werk je aan twee projecten waarin je tegelijkertijd workshops en cursussen volgt.
- In het tweede semester (6 maanden) ga je fulltime aan de slag bij de werkgever. Tijdens dit semester zijn er vier terugkomsessies bij de HvA voor het uitwisselen van ervaringen en het aansnijden van interessante thema's.

Semester 1:

- Basisfase:
 - Programming
 - Databases
 - Object Oriented Programming
- Project 1 Quizmaster:
 - Workshop: Scrum
 - Workshop: Git
 - Workshop: Model View Control
 - Workshop: NoSQL
 - Workshop: Testing
 - Workshop: Teamwork
- Training en coaching

• Project 2 Webapplicatie:

- Cursus: Advanced Programming
- Cursus: Algoritmes & Datastructures
- Cursus: Design Patterns
- Cursus: Requirements Engineering
- Cursus: Software Maintenance
- Cursus: Software Testing
- Cursus: UML
- Cursus: WEB applications
- Workshop: Scrum
- Workshop: Professionele vaardigheden

Semester 2:

- Fulltime werken bij werkgever
- Vier terugkomsessies
- Coaching

SEMESTER 1 - BEGINNFASE

- Programming
- Databases
- Object Oriented Programming

BEGINFASE - PROGRAMMING

Binnen de module Programming leer je hoe een programmeertaal toegepast kan worden voor het oplossen van eenvoudige programmeerproblemen. Tevens leer je bestaande, kleine programma's begrijpen en aanpassen.

Na afloop van de module Programming kun je:

- gebruikmaken van variabelen, datatypes en operatoren
- expressies samenstellen en berekenen
- control structures (if, switch, for, while, ...) construeren en bepalen welke wanneer nodig is
- functies/methoden construeren en gebruiken
- arrays (container) construeren en toepassen
- uitleggen wat een Class/Object is en een eenvoudige Class gebruiken

BEGINFASE - DATABASES

Het ontwerpen van databases leer je met behulp van kleine casussen waarin je de informatiebehoefte leert herkennen en vertalen naar een conceptueel gegevensmodel. De complexiteit van de casussen neemt toe en leidt tot recursieve relaties, subtypering en meerdere constraints. Je leert hoe je dit gegevensmodel kunt omzetten in een tabelstructuur van een relationele database. Daarnaast wordt aandacht besteed aan de implementatie van een database. Je leert vele mogelijkheden van de relationele taal Structured Query Language (SQL) gebruiken om allerlei soorten gebruikersvragen te vertalen naar een SQL query. En je leert hoe je informatie automatisch kan controleren en verwerken met behulp van triggers. Je maakt ook kennis met de problematiek van een multi-user omgeving en met andere oplossingen dan een relationele database om gegevens op te slaan.

BEGINFASE - DATABASES

Na afloop van de module Databases 1 kun je:

- de principes van genormaliseerde relationele databases uitleggen
- een database definiëren m.b.v. de SQL: tabellen, sleutels, integriteitsregels, actieregels, checks
- een conceptueel gegevensmodel opstellen in de vorm van een Entiteit Relatie Diagram (ERD), m.b.v. recursieve relaties en subtypering
- een relationele database ontwerpen en implementeren en daarbij de juiste informatie automatisch controleren en verwerken m.b.v. triggers
- gegevens in een relationele database bevragen en manipuleren m.b.v. SQL, gebruikmakend van Group by, Subqueries, Views, Stored Procedures
- database indexen gebruiken om performance te verbeteren
- de principes van databases in een multi-user omgeving benoemen en uitleggen en weet wat transactions en locking strategies inhouden.

BEGINFASE - OBJECT ORIENTED PROGRAMMING

Binnen de module Object Oriented Programming leer je over de volgende onderwerpen: Classes en Objecten, Complexe datatypes, Overerving (Inheritance), Polymorfisme, Programmaontwerp, Documentatie en Coderingsstandaard.

Na afloop van de module Object Oriented Programming 1 kun je:

- classes ontwerpen, implementeren en gebruiken
- complexe datatypes implementeren en gebruiken
- overerving implementeren en toepassen
- polymorfisme toepassen
- abstracte classes en interfaces implementeren en gebruiken
- exceptions implementeren en gebruiken
- text files lezen en schrijven
- connecties met een SQL databases maken en gebruiken d.m.v. JDBC
- duidelijke documentatie maken die voldoet aan de coderingsstandaard

SEMESTER 1 – PROJECT 1 QUIZMASTER

De volgende workshops worden gegeven:

- Workshop: Scrum
- Workshop: Git
- Workshop: Model View Control
- Workshop: NoSQL
- Workshop: Testing
- Workshop: Teamwork

PROJECT 1 - QUIZMASTER

In dit project van drie weken bouw je in teamverband de backend van een JavaFX applicatie. Op basis van gegeven JavaFX-schermen bouw je een Model-View-Controller-applicatie die met behulp van JDBC alle benodigde gegevens opslaat in en ophaalt uit een SQL database. Een deel van de gegevens wordt opgeslagen en opgehaald uit een NoSQL (documentstore) database. Tijdens het project krijg je aanvullende en ondersteunende workshops.

Na afloop van deze projectopdracht kun je:

- een IT-systeem ontwerpen m.b.v. UML
- de realisatie van een IT-systeem plannen m.b.v. Scrum
- versiebeheer toepassen m.b.v. Git
- een (deel van een) IT-systeem realiseren
- vanuit een Java-programma communiceren met een relationele database d.m.v. JDBC
- vanuit een Java-programma communiceren met een NoSQL database m.b.v. Json-bestanden
- de functionaliteit van een eenvoudig IT-systeem evalueren en testen

SEMESTER 1 – PROJECT 2 WEBAPPLICATIE

De volgende cursussen en workshops worden gegeven:

- Cursus: Advanced Programming
- Cursus: Algoritmes & Datastructures
- Cursus: Design Patterns
- Cursus: Requirements Engineering
- Cursus: Software Maintenance
- Cursus: Software Testing
- Cursus: UML
- Cursus: WEB applications
- Workshop: Scrum
- Workshop: Professionele vaardigheden

PROJECT 2 – WEBAPPLICATIE

In het tweede project, dat acht weken duurt, ga je een een gelaagde web applicatie modelleren; het bouwen van de presentatie-, business-, en data laag; instellingen kunnen lezen en schrijven naar bestand, data naar database; passende foutafhandeling; SOLID principes toepassen. Bij het ontwerpen worden de bijbehorende ontwerppatronen en Collections/Datatype en Unittests gebruikt: JavaScript, AJAX, html/css. De webapplicatie wordt ontwikkeld met behulp van een low-code platform namelijk Mendix developer.

Het project is ingedeeld in drie sprints.

- [Sprint 1 richt zich op de backend](#)
- [Sprint 2 richt zich op de backend en de frontend](#)
- [Sprint 3 richt zich geheel op de frontend.](#)

Na elke sprint vindt een review en retrospective plaats. In de laatste week vindt een individueel techno assessment plaats. Dit gebeurt op basis van een portfolio dat opgesteld wordt door de cursisten zelf op basis van voorgeschreven rubrics.

Tijdens dit project leer je nieuwe technieken die je nodig hebt voor het bouwen van de webapplicatie. Je krijgt verdiepende cursussen op het gebied van programmeren en je krijgt cursussen en workshop die je nodig hebt om goede, gestructureerde en onderhoudbare code te schrijven.

SEMESTER 2

- Fulltime werken bij werkgever
- Vier terugkomsessies
- Coaching



WWW.IT-OMSCHOLING.NL