Leeruitkomsten Software Engineering

Hierbij de beschrijving van de concept leeruitkomsten van HBO-ICT Software Engineering. De leeruitkomsten zijn geheel conform HBO-i en Tuning. Ook is een koppeling met de SSDLC. De [data is online](https://github.com/Windesheim-HBO-ICT/Leeruitkomsten/tree/main/SE/Analyse) beschikbaar. Ook de [gevolgde methodiek](https://4cid.nl/4c-id-leeruitkomsten/) is beschikbaar. Niveau 1 zijn niet opgenomen in dit overzicht, maar de gegevens zijn wel in de brondata aanwezig.

In een volgende fase kunnen meer standaarden opgenomen worden. Bepaald worden of de toegevoegde processtappen vanuit de analyse gewenst zijn (kolom Bron in de brondata). En natuurlijk de koppeling en nadere uitwerking van Professional Skills.

# Software Engineering - Niveau 2

HBO-i schrijft als context voor niveau 2 voor: Software systeem bestaand uit subsystemen.

**Adviseren**De student valideert requirements.  
De student adviseert over de onderdeelselectie.  
De student adviseert over de architectuur.  
De student gaat geen testproces adviseren.

**Analyseren**De student definieert het probleemdomein.   
De student verzamelt requirements. En past ASVS, RRM en ISO25010 toe.  
De student analyseert requirements.   
De student specificeert requirements.   
De student voert een analyse uit.

**Manage & Control**De student managet het ontwikkelproces. En past Documentbeheer WHM en APA toe.  
De student beheert requirements.   
De student beheert de ontwikkelstraat.   
De student gaat geen ontwikkelstraat evalueren.   
De student gaat geen configuratie-, change- en releasemanagement uitvoeren.

**Ontwerpen**De student maakt gebruik van een gegeven opgestelde teststrategie. En past TDD, mutation testing, E2E, unittesten, integratietesten en het V-model toe.  
De student stelt een ontwerp op. En past HBO-ICT richlijnen ontwerpen, wiskunde basis, UML, wireframes, C4, SOLID, DRY, Clean Architecture, Threat Model, evil userstories en KISS toe.  
De student stelt de kwaliteit vast.   
De student stelt een testontwerp op.

**Realiseren**De student richt de ontwikkelomgeving in.   
De student bouwt een softwaresysteem. En past Code standaarden, OO, Clean Code, DAST en SAST toe.  
De student gaat geen applicatie refactoren.   
De student beredeneert bouwkeuzes.   
De student stelt een test op.   
De student voert een test uit. ZAP  
De student evalueert een test.   
De student gebruikt het beheersysteem.   
De student gaat geen software beschikbaarstellen.

# Software Engineering - Niveau 3

HBO-i schrijft als context voor niveau 3 voor: Schaalbaar software systeem en aansluitend bij bestaande systemen.

**Adviseren**De student valideert requirements.  
De student adviseert over de onderdeelselectie.  
De student adviseert over de architectuur.  
De student gaat geen testproces adviseren.

**Analyseren**De student definieert het probleemdomein.  
De student verzamelt requirements.  
De student analyseert requirements.  
De student specificeert requirements.  
De student voert een analyse uit.

**Manage & Control**De student managet het ontwikkelproces.De student beheert requirements.De student beheert de ontwikkelstraat.De student evalueert de ontwikkelstraat.De student voert configuratie-, change- en releasemanagement uit.

**Ontwerpen**De student stelt een teststrategie op.  
De student stelt een ontwerp op.  
De student stelt de kwaliteit vast.  
De student stelt een testontwerp op.

**Realiseren**De student richt de ontwikkelomgeving in.   
De student bouwt een softwaresysteem.   
De student refactort een applicatie.   
De student beredeneert bouwkeuzes.   
De student stelt een test op.   
De student voert een test uit.   
De student evalueert een test.   
De student gebruikt het beheersysteem.   
De student stelt software beschikbaar. En past CI/CD toe.