

Opdracht: van tekst naar schema

De watervalmethode

Het ontwikkelen van software is meer dan alleen het schrijven van een computerprogramma (meer dan alleen programmeren). We gaan aan de slag met de *watervalmethode*, een gestructureerde manier om applicaties te ontwikkelen.

Je gaat in tweetallen of drietallen een paar opdrachten uitvoeren om bekend te raken met de software-ontwikkelmethode.

Leerdoelen

- Je kunt de fasen van de watervalmethode benoemen.
- Je kunt uitleggen wat er tijdens de analyse-fase gebeurt en wat voor een informatie er allemaal in een analysedocument moet komen te staan.
- Je kunt vanuit een gegeven casus zelfstandig een analysedocument schrijven.

Tweetallen formeren

Deze opdracht wordt uitgevoerd in tweetallen.

Opdrachtbeschrijving

Voer in tweetallen onderstaande opdracht uit. Lever het resultaat van de opdracht in via Canvas door een foto of video van je schema te maken.

Klap je laptop dicht want bij deze opdracht is het niet de bedoeling dat je internet gebruikt. Bij deze opdracht maak je kennis met de watervalmethode van software-ontwikkeling.

Opdracht: bestudeer onderstaande tekst die gaat over de watervalmethode, een gestructureerde manier om software te ontwikkelen. Maak na bestudering van de tekst een schematische weergave van de watervalmethode. Lever je schema in via Canvas.

De watervalmethode

Na de oplevering van een applicatie is er altijd sprake van onderhoud. De laatste fase van een softwareontwikkeltraject is altijd de onderhoudsfase, soms ook wel beheerfase genoemd. In deze fase wordt het systeem operationeel gehouden. Soms kunnen er kleine patches (updates) worden doorgevoerd als er tijdens het werken met de software problemen optreden.

Met een projectmethodiek als Prince2 kan de watervalmethode gestuurd en gemanaged worden. Vanaf de eerste stap van het project tot en met de laatste stap. Een belangrijke fase bij waterval is de ontwerpfase. Tijdens de ontwerpfase wordt er een ontwerp (bijvoorbeeld een algoritme – een stappenplan *hoe* de computer een ingewikkelde berekening moet uitvoeren) gemaakt en gedocumenteerd in een document. Het ontwerpdocument dient als input voor de volgende fase, de implementatiefase. In die fase begint men met programmeren.

Voorafgaand aan de ontwerpfase wordt eerst nog bepaald *wat* nu precies het probleem is dat moet worden opgelost. Het vastleggen en documenteren wat het probleem is vindt plaats in de

analysefase. Voorafgaand aan de analyse is er soms nog een pre-analyse, een fase waarbij diverse vooronderzoeken kunnen worden uitgevoerd met vaak als doel om kennis op te doen van het probleem van de klant.

Tijdens de analysefase worden onder meer de user requirements opgesteld. Dat is een lijst van functionaliteiten die de uiteindelijke applicatie moet gaan uitvoeren. Het is belangrijk om te weten dat deze lijst alleen functionele eisen bevat, dus geen technische details.

De lijst van functionele eisen wordt vaak geprioritiseerd. Elke functionaliteit krijgt een prioriteit. Bij de ontwikkeling van een game bijvoorbeeld heeft de functionaliteit "Het karakter moet door het doolhof kunnen bewegen" een hogere prioriteit dan "De haarkleur van het karakter moet door de gebruiker in te stellen zijn". Ook zonder dat de gebruiker de haarkleur kan uitkiezen zal het game werken.

In een ontwerpdocument staat veel technische details die uitleggen hoe de functionele eisen van de analyse geprogrammeerd moeten gaan worden. De stappen die het computerprogramma moet uitvoeren staan daarin beschreven.

Nadat de software is geschreven in bijvoorbeeld een programmeertaal als C# dient deze getest te worden. Dit kan de programmeur zelf doen, maar vaker zie je dat er aparte functionele testers zijn die de testfase uitvoeren. De programmeur kan tijdens de testfase worden bijgeschakeld om fouten in de software op te lossen. Het testen gebeurt altijd aan de hand van een testplan. Dit testplan kan al in een vroeg stadium (bijvoorbeeld als onderdeel van de analysefase) worden geschreven. Na het uitvoeren van de testen wordt door de tester(s) een testrapport geschreven met de bevindingen.

Zowel het analysedocument als het ontwerpdocument worden door de programmeur gebruikt om te coderen. De implementatiefase in de software engineering is hiermee anders dan bij bijvoorbeeld de realisatie van een ERP-systeem (bedrijfsbreed softwarepakket) waar de implementatiefase meer overeen komt met onze installatiefase (installeren van de opgeleverde software). De implementatiefase wordt ook wel eens realisatiefase genoemd. Het realiseren bestaat uit coderen (programmeren) in bijvoorbeeld C# maar ook tegelijkertijd het oplossen van typefouten in de code. Dit gebeurt allemaal door een team van programmeurs. De programmeurs bekijken soms ook elkaars code en geven elkaar daar feedback over. Dit noemen ze code reviews.