

# Hoofdstuk 1

## Introductie

### 1.1 Over softwareontwikkeling

In de meeste gevallen zijn er veel mensen betrokken bij de ontwikkeling van software: niet alleen de klant die de opdrachtgever is en de programmeurs die het eigenlijke werk doen, maar ook mensen die deskundig zijn op het gebied waar de applicatie over gaat, analisten, systeembeheerders, mensen met financiële verantwoordelijkheid et cetera.

Een ontwikkeltraject kan maanden of zelfs jaren duren. Omdat er vaak veel mensen bij betrokken zijn, is er in feite sprake van een *project*. In zo'n geval moet je een projectmatige aanpak hebben om alles in goede banen te leiden. Meestal is er dan ook een projectmanager die de gang van zaken bewaakt. Over projectmanagement gaat dit boek niet.

Behalve een algemene projectmatige aanpak heb je ook een meer inhoudelijke *methode* nodig om binnen redelijke tijd tot het gewenste product te komen. In het verleden zijn er veel methoden bedacht en beschreven om op een systematische manier te komen tot softwareontwikkeling. Bekende methoden zijn *watervalmethode*, *Rational Unified Process* (RUP) en *eXtreme Programming* (XP). Vaak geven deze methoden aanwijzingen hoe je het ontwikkelproject moet inrichten.

#### 1.1.1 De watervalmethode

Bij de watervalmethode verloopt het ontwikkelproces als een stroom van fasen die je achtereenvolgens doorloopt om het eindproduct te krijgen. De fasen waaruit deze methode bestaat zijn de volgende:

- vastleggen van de eisen waaraan de applicatie moet voldoen;
- maken van een ontwerp;
- coderen van diverse gedeelten van de applicatie (implementatie);
- samenvoegen van de verschillende onderdelen (integratie);
- testen van de applicatie en het aanbrengen van verbeteringen;
- afleveren en installeren bij de klant;
- onderhoud na aflevering.

Deze fasen vormen een vrij natuurlijke route waarlangs je een willekeurig product (niet alleen een applicatie) tot stand kunt brengen. Immers, voordat je gaat ontwerpen moet je weten wat de eisen zijn die de klant aan het product stelt. Het ontwerp is natuurlijk nodig om te kunnen coderen, zeker als je met een team van programmeurs werkt. De geschreven code moet je uiteraard testen voor je het aflevert en installeert bij de klant.

Er is vrij veel kritiek op de watervalmethode, waarschijnlijk vooral omdat deze in het verleden nog wel eens rigoureuus is toegepast: elke fase moest volledig zijn afgerond alvorens je aan een volgende fase kon beginnen. Dat betekent bijvoorbeeld dat de klant in het begin van het proces, aan het einde van de eerste fase, zijn handtekening zet onder de eisen waaraan de applicatie moet voldoen. Wijzigingen in de eisen zijn niet mogelijk, of alleen ten koste van veel mankracht en geld.

Maar in de praktijk komt het vaak voor dat de klant weliswaar van tevoren een aantal eisen heeft, maar dat deze eisen gaandeweg de ontwikkeling van de applicatie kunnen veranderen, dat niet alle eisen even zwaar blijken te wegen, dat de klant sommige oplossingen waar de programmeurs mee komen niet heeft kunnen voorzien en om die reden sommige eisen kan laten vallen, of juist met nieuwe eisen komt.

Voordeel van de watervalmethode is dat de fasen helder zijn afgebakend, waardoor iedereen weet waar hij aan toe is; nadeel is dat het model geen ruimte laat om later terug te komen op eerder genomen beslissingen.

Er zijn andere methoden, waarin de flexibiliteit als het ware is ingebouwd.

### 1.1.2 Rational Unified Process

Het Rational Unified Process (RUP) is een methode ontworpen door Rational Software (nu IBM). De methode onderscheidt in principe vier fasen:

- *inception*, de startfase van het project waarin gekeken wordt naar de haalbaarheid, de financiële aspecten en de risico's van het project;
- *elaboration*, de uitwerkingsfase waarin een analyse wordt gedaan van het probleem en een zogeheten ontwerpmodel wordt gemaakt op grond van de resultaten van de analyse;
- *construction*, de fase van het maken van de software;
- *transition*, de fase waarin de software wordt overgedragen aan de klant.

RUP heeft expliciet een aantal momenten ingebouwd waarop het proces kan worden afgeblazen, bijvoorbeeld wanneer aan het eind van de startfase blijkt dat de financiële consequenties te groot zijn.

RUP is een *iteratief* proces, voornamelijk in de constructiefase. Iteratief wil zeggen dat de software niet in één keer tot stand komt, maar in betrekkelijk kleine, min of meer afgeronde componenten. Bij de oplevering van een component wordt deze geëvalueerd, de eisen eventueel bijgesteld, de analyse en het model eventueel veranderd en de opgedane

ervaring meegenomen bij de ontwikkeling van de volgende component. Bij dit alles maakt RUP nadrukkelijk gebruik van grafische weergaven van aspecten van de software met behulp van UML omdat RUP veel waarde hecht aan goede en uitvoerige documentatie, in zowel geschreven als grafische vorm.

Over RUP valt veel meer te zeggen dan ik hier heb gedaan. Op internet zijn veel bronnen te vinden die uitvoerig ingaan op deze methode.

### 1.1.3 eXtreme Programming

Een betrekkelijk nieuwe methode van softwareontwikkeling is eXtreme Programming (XP) met een paar opvallende kenmerken. XP houdt er niet van om de eisen voor een applicatie eerst uitvoerig op papier te formuleren, omdat dit veel tijd kost en de eisen later toch vaak blijken te veranderen. In plaats daarvan wordt de klant nauw bij het XP-ontwikkelp proces betrokken.

Als uitgangspunt voor het proces gelden zogeheten *user stories*, een tamelijk globale omschrijving door de gebruiker van een deel van de functionaliteit die het systeem moet krijgen. Het mag niet langer dan een tot drie weken kosten om een user story te implementeren. Tijdens de analyse en het ontwerp van een user story en tijdens de implementatie is de klant continu aanwezig om vragen over details te beantwoorden en om feedback te geven op het geleverde werk.

XP hecht veel waarde aan het in een vroeg stadium testen van kleine stukjes code (*unit testen*) en het regelmatig verbeteren van de structuur van de code (*refactoring*). Het programmeerwerk wordt gedaan door programmeurs die in tweetallen aan een pc zitten (*pair programming*). Het voordeel hiervan is, eenvoudig gezegd, dat twee meer weten dan een en dat programmeurs elkaar kunnen stimuleren en aanvullen. Terwijl de een code tikt, kan de ander nadenken. Regelmatig wisselen de programmeurs van rol. Als een bepaalde user story geïmplementeerd is, voert de klant een test uit om te zien of hij het gemaakte deel accepteert.

XP maakt op een betrekkelijk informele manier gebruik van UML en alleen als het nodig is.

### 1.1.4 Gemeenschappelijke aspecten van de methoden

Er bestaan nog veel meer methoden voor softwareontwikkeling, en vaak vind je daarin aspecten van de drie hier genoemde methoden terug. De verschillen zitten vooral in de nadruk die op sommige aspecten wordt gelegd of juist niet, op de volgorde van de werkzaamheden en op de organisatie van het hele project.

Ondanks deze verschillen is een groot aantal aspecten in alle methoden terug te vinden:

- onderzoek naar de eisen waaraan het systeem moet voldoen;
- analyse van de eisen en van het probleemgebied;
- het maken van een model;
- het schrijven van code met het model als uitgangspunt;

- het testen van het systeem;
- de oplevering van het systeem.

### 1.1.5 Welke methode gebruikt dit boek?

Dit boek gebruikt geen vastomlijnde methode, maar richt zich op de algemene aspecten van softwareontwikkeling die in de vorige paragraaf staan genoemd en die, onafhankelijk van de methode die je kiest, belangrijk zijn voor elke programmeur.

## 1.2 Samenvatting

- Bij softwareontwikkeling is vaak sprake van teamwerk gedurende langere tijd.
- Er zijn veel verschillende methoden voor de ontwikkeling van applicaties die alle tot doel hebben het ontwikkelproces in goede banen te leiden.
- De watervalmethode, RUP en XP zijn bekende ontwikkelingsmethoden.
- Naast grote verschillen zijn er ook veel overeenkomsten tussen de verschillende methoden.

## 1.3 Vragen

- 1 Geef een korte omschrijving van de watervalmethode.
- 2 Geef een korte omschrijving van RUP.
- 3 Geef een korte omschrijving van XP.

## 1.4 Oefeningen

### Oefening 1.1

Zoek meer informatie over XP en geef hiervan een beschrijving in je eigen woorden. Welke voor- en nadelen zie je in deze methode?

### Oefening 1.2

Zoek meer informatie over RUP en geef hiervan een beschrijving in je eigen woorden. Welke voor- en nadelen zie je in deze methode?