
Een Inleiding tot Software Engineering

Software engineering

- De economie is compleet afhankelijk van software.
- Meer en meer systemen zijn gecontroleerd door software
- Software engineering is bezig met:
 - theorieën,
 - methodes
 - tools voor professionele software ontwikkeling.

Wat is software?

- Computer programma's en geassocieerde documentatie zoals lastenboek, ontwerp modellen en user manuals.
- Software producten kunnen worden ontwikkeld voor een bepaalde klant of voor een algemene markt.
- Software producten zijn:
 - Generiek – ontwikkeld om te worden verkocht aan een waaier van verschillende klanten bv. PC software zoals Excel of Word.
 - Maatsoftware – ontwikkeld voor één enkele klant overeenkomstig zijn specificatie.
- Nieuwe software kan worden gecreëerd door het ontwikkelen van nieuwe programma's, het configureren van generieke software systemen of het hergebruiken van bestaande software.

Wat is software engineering?

- Software engineering is een discipline die bezig is met alle aspecten van software productie.
- Software engineers moeten werken volgens een systematische benadering en zullen gebruik maken van tools en technieken, afhankelijk van:
 - het op te lossen probleem
 - de development constraints
 - en de beschikbare resources

Wat is een software proces?

- Een reeks activiteiten met als doel de ontwikkeling of uitbreiding van software.
- Generieke activiteiten in alle software processen zijn:
 - Specificatie - wat moet het systeem doen en welke zijn de development constraints
 - Ontwikkeling – productie van het software systeem
 - Validatie – nagaan of de software is wat de klant wenst
 - Onderhoud – wijzigingen aan de software in respons op de vraag die wijzigt.

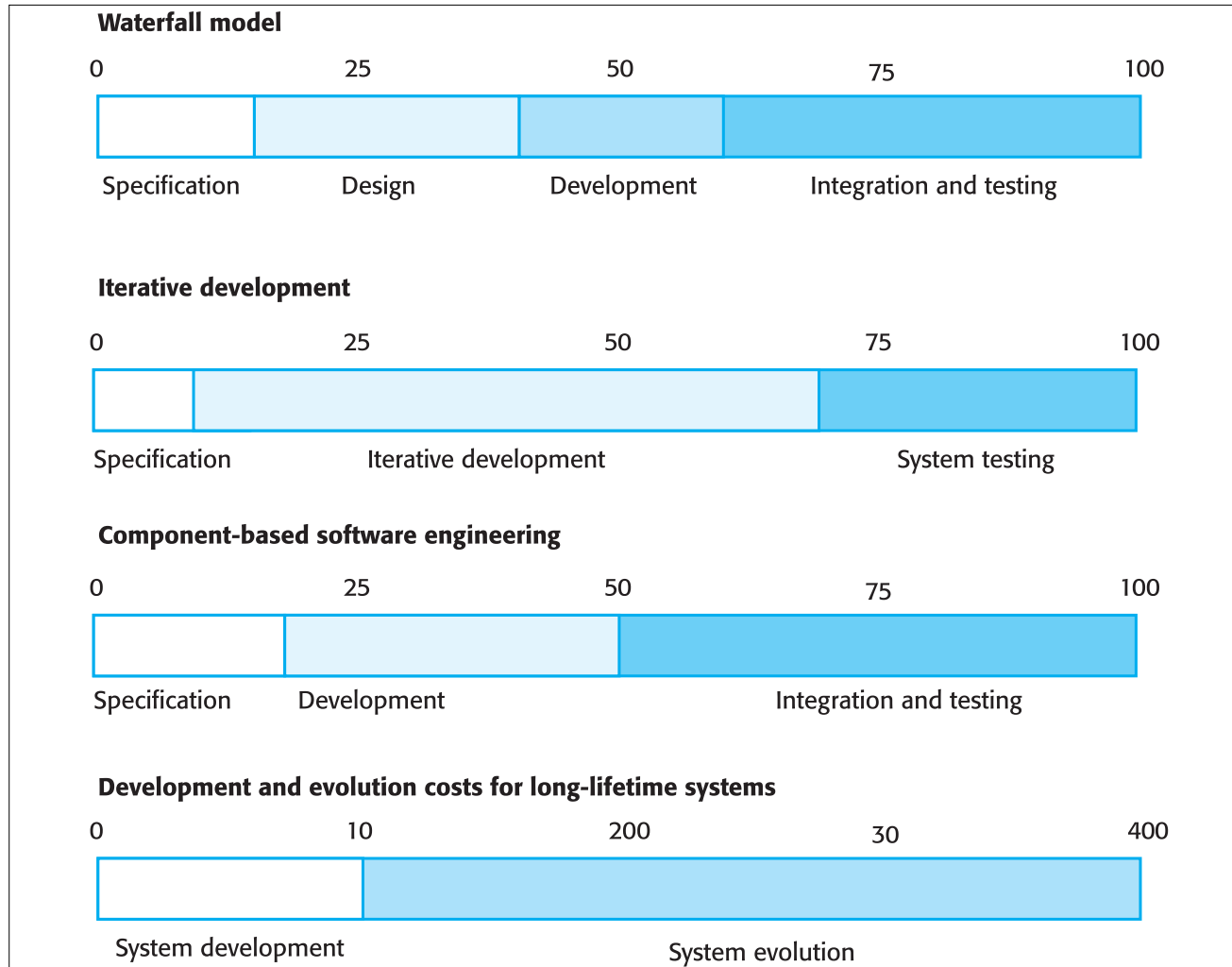
Wat is een software proces model?

- Een eenvoudige voorstelling van een software proces, getoond vanuit een specifiek perspectief.
- Voorbeelden van proces perspectieven zijn
 - Workflow perspectief – sequentie van activiteiten;
 - Data-flow perspectief - informatie stromen;
 - Rol/actie perspectief - wie doet wat.
- Generieke proces modellen
 - Waterval;
 - Iteratieve ontwikkeling;
 - Component-based software engineering.

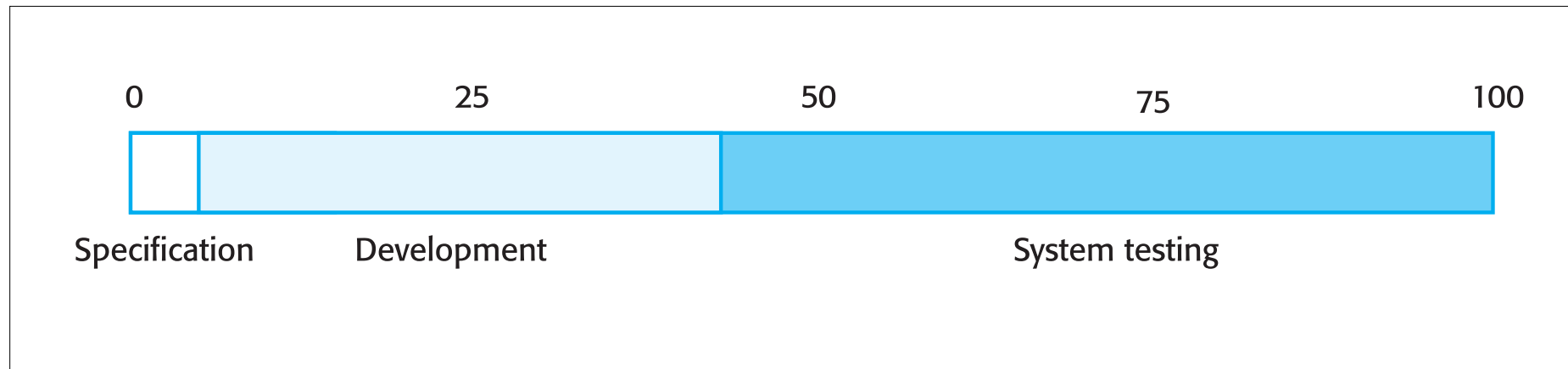
Wat zijn de kostenfactoren bij software engineering?

- Ongeveer 60% zijn ontwikkelingskosten, 40% gaat naar testen.
 - Voor maatsoftware, zijn de onderhoudskosten soms groter dan de ontwikkelingskosten.
- Kosten variëren afhankelijk van het type systeem dat wordt ontwikkeld, en de eisen gesteld aan systeemattributen zoals performantie en betrouwbaarheid.
- De kostenspreiding hangt af van het ontwikkelingsmodel dat wordt gebruikt.

Kostenverdeling



Productontwikkelingskosten



Welke methodes biedt Software engineering?

- Gestructureerde benaderingen voor software ontwikkeling met inbegrip van systeem modellen, regels, design advies en procesbegeleiding.
- Beschrijving van systeemmodellen
 - Beschrijving van grafische modellen die moeten worden geproduceerd;
- Regels
 - Beperkingen van toepassing op systeem modellen;
- Aanbevelingen
 - Advies ivm “good design practice”;
- Procesbegeleiding
 - Welke activiteiten worden uitgevoerd.

Wat is CASE (Computer-Aided Software Engineering)

- Software systemen die bedoeld zijn om geautomatiseerde ondersteuning te bieden voor software proces activiteiten.
- CASE systemen zijn ondersteunen meestal een bepaalde methode.
- Upper-CASE
 - Tools voor het ondersteunen van de eerste proces activiteiten van analyse of design;
- Lower-CASE
 - Tools voor het ondersteunen van de latere activiteiten zoals programmeren, debuggen en testen.

Wat zijn de activiteiten van goeie software?

- Software moet de vereiste functionaliteit en performantie leveren aan de gebruiker, en moet onderhoudbaar zijn, betrouwbaar en aanvaardbaar
- Onderhoudbaarheid
 - Software moet ontwikkeld zijn zodat gemakkelijk aanpassingen mogelijk zijn;
- Betrouwbaarheid
 - Software moet correct en consistent reageren;
- Efficiëntie
 - Software moet slechts de systeembronnen gebruiken die hij echt nodig heeft;
- Gebruiksvriendelijkheid
 - Software moet worden aanvaard door de gebruikers waarvoor hij werd ontworpen. Dit betekent dat hij gebruiksvriendelijk moet zijn en compatibel met andere systemen.

Wat zijn de uitdagingen voor software engineering?

- Heterogeniteit, opleveren en vertrouwen.
- Heterogeniteit
 - Ontwikkelen van software die heterogene platforms en omgevingen overdekt;
- Opleveren
 - Ontwikkelingstechnieken gebruiken die leiden tot een goed te plannen oplevering van de software;
- Vertrouwen
 - Ontwikkelingstechnieken die helpen bewijzen dat de software kan vertrouwd worden door zijn gebruikers.