



SDLC – Software Development Lifecycle

IT Organisation 2

2020-2021

**Ann Das
Kim Gijbels
Britt Verschakelen**

Cursusinhoud

- Hoofdstuk 1: Ontwikkelomgevingen
- Hoofdstuk 2: Software development lifecycle
- Hoofdstuk 3: Systeemontwikkeling

- Hoofdstuk 1: Ontwikkelomgevingen
 - De student kan de ontwikkelomgevingen benoemen
 - De student kan de ontwikkelomgevingen in eigen woorden uitleggen
 - De student kan de ontwikkelomgeving toepassen op een casus
- Hoofdstuk 2: Software development lifecycle
 - De student kan SDLC in eigen woorden uitleggen
 - De student kan de domeinen van SDLC benoemen
 - De student kan de domeinen van SDLC in eigen woorden uitleggen
 - De student kan de domeinen van SDLC herkennen in een casus
 - De student kan de domeinen van SDLC toepassen in een casus
- Hoofdstuk 3: Systeemontwikkeling
 - De student kan de rol van systeemanalist in eigen woorden uitleggen
 - De student kan de kenmerken van een systeemanalist in eigen woorden uitleggen

Hoofdstuk 1

Ontwikkelomgevingen

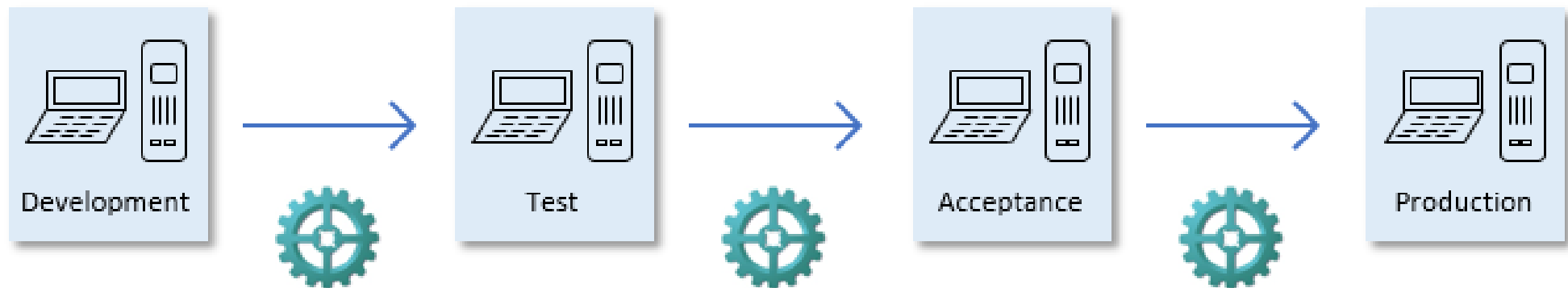


Ontwikkelomgevingen

- Idealiter wordt er met deze omgevingen gewerkt tijdens de ontwikkeling

- Development
 - Test
 - Acceptatie
 - Productie
- DTAP

- Afhankelijk van de organisatie worden al deze omgevingen gebruikt, of enkel een subset
 - Productie is er altijd!



Development

- HW/SW* platform en tools die de ontwikkelaar gebruikt voor ontwikkeling van de code
- Naast development, ook eerste testen en/of review mogelijk
- Meestal nog heel instabiel, omdat er nog geen uitgebreide testen zijn gebeurd



*HW/SW: Hardware/Software

Test

- Wanneer de development omgeving getest werd door de ontwikkelaars, worden alle functionaliteiten van de development omgeving 'gekopieerd' naar de testomgeving
- Stabieler dan de test omgeving, omdat er meer getest is

Acceptatie

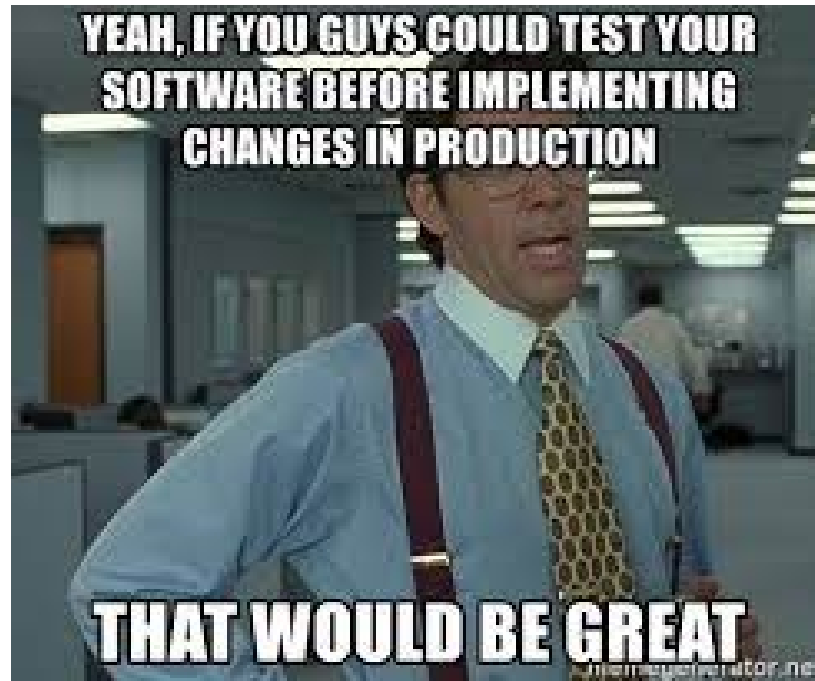
- Als alle noodzakelijke features op de testomgeving goed werken, wordt de testomgeving 'gekopieerd' naar de acceptatie omgeving
- Op deze omgeving gaat de klant zijn goedkeuring geven voor de gebouwde features
- De klant of een vertegenwoordiger van de klant, voert de testen uit op deze omgeving
 - Uitbesteding aan testers kan ook!
- Dit is de laatste stap voor het product toegankelijk is voor de eindgebruikers
- De acceptatie-omgeving is qua software, hardware en gegevens zoveel mogelijk gelijk aan de productie omgeving

Acceptatie

- Mogelijke testen die uitgevoerd kunnen worden
 - Functionele test
 - Testen of de functionaliteiten werken
 - Bv: Als de toepassing een webshop is, testen of alles werkt zoals verwacht
 - Piek- of stresstest
 - Testen of het systeem tegen piekbelasting kan
 - Bv: veel verkeer op de website, moeilijke berekeningen,...
 - Monkey-proof test
 - Is het systeem bestand tegen ongeoorloofde en onjuiste handelingen
 - Bv: testen met random inputs en random klikken door random gebruikers
 - Pen test of penetratietest
 - Proberen binnen te breken in de applicatie door kwetsbaarheden uit te buiten
 - Bv.: Wachtwoorden kraken, phishing, scannen van kwetsbaarheden

Productie

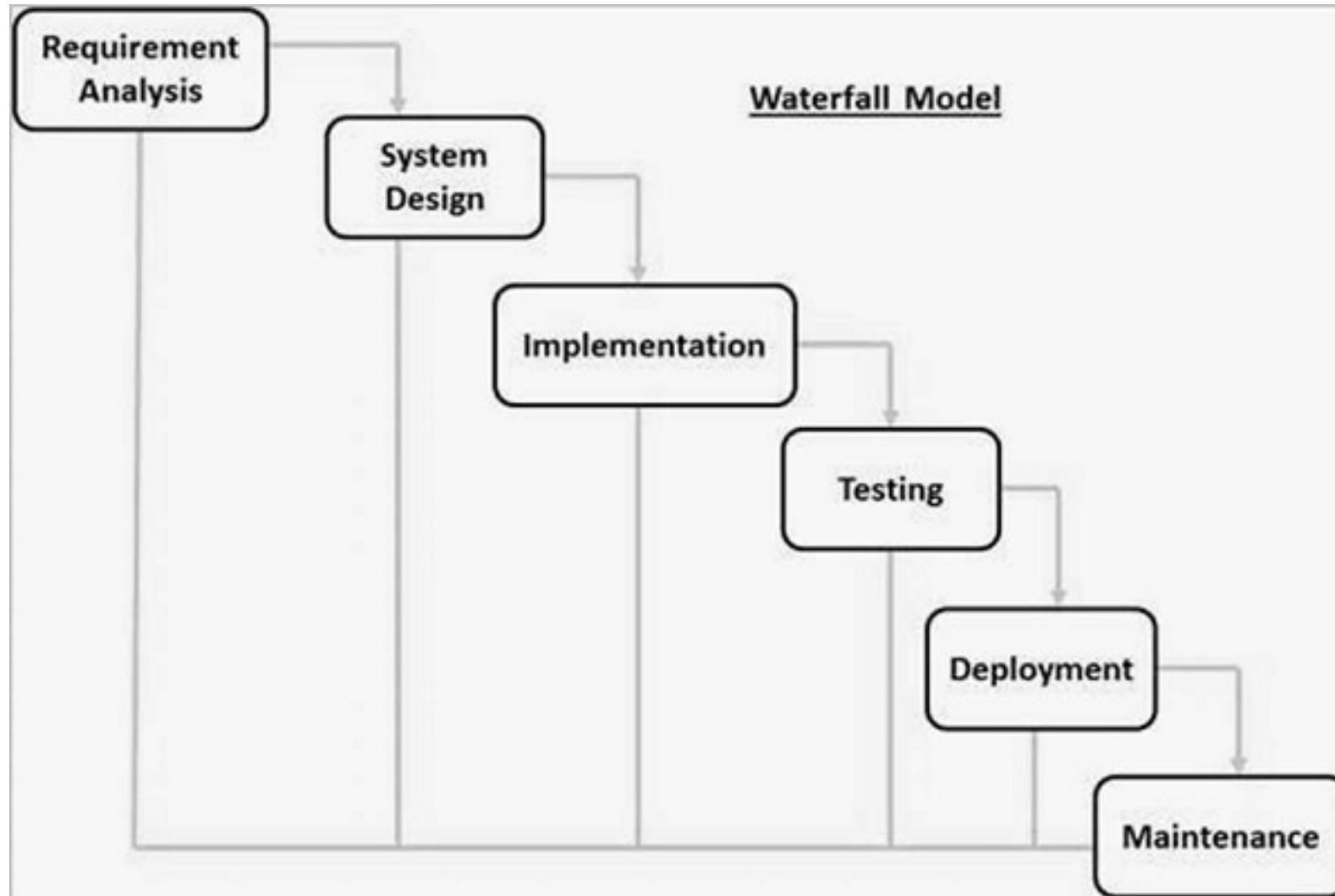
- Deze omgeving is toegankelijk voor de eindgebruiker
- Als de klant zijn akkoord gegeven heeft op de acceptatie omgeving (door te testen), wordt het project op deze omgeving gezet



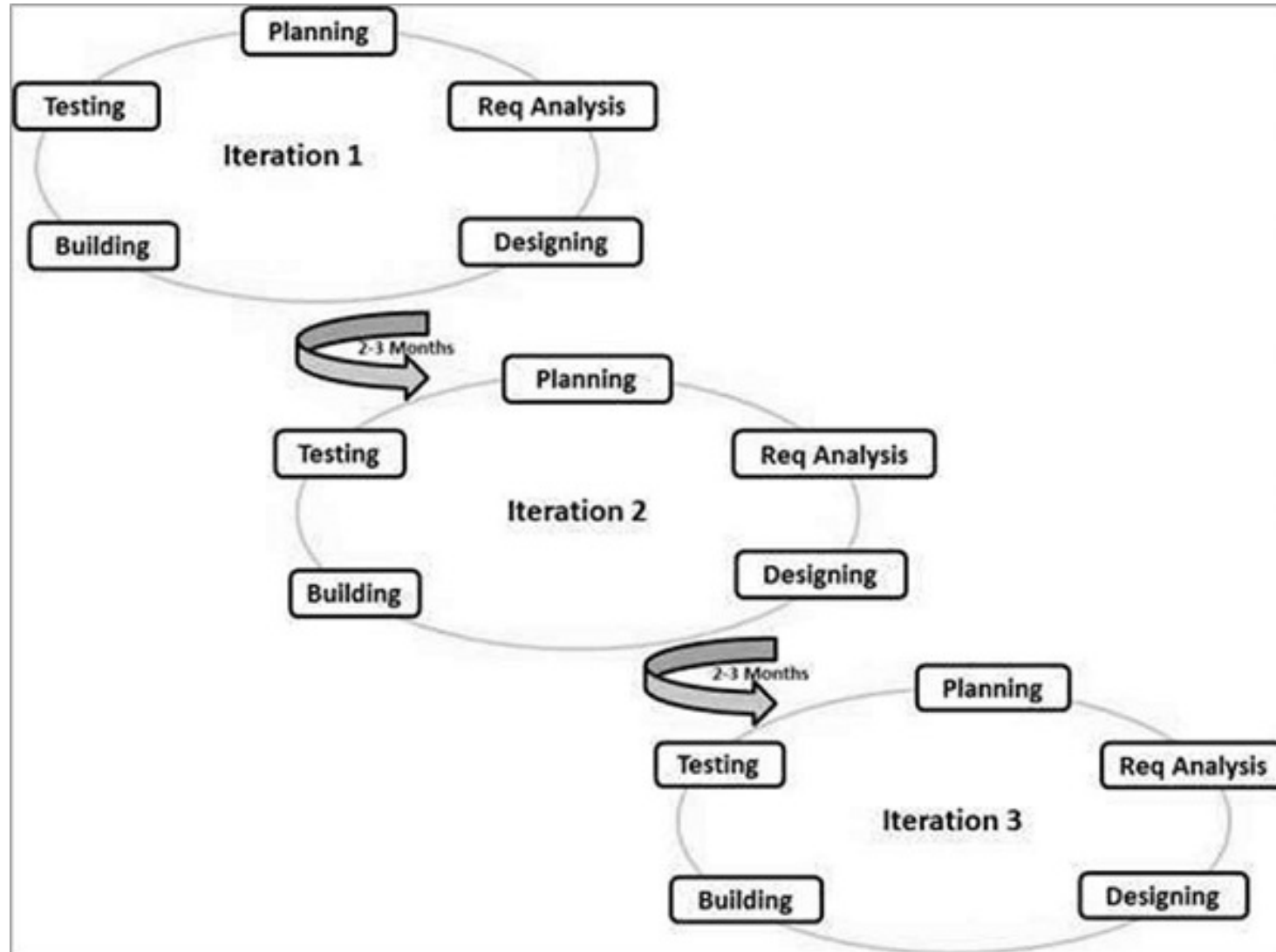
Hoofdstuk 2

Software development lifecycle

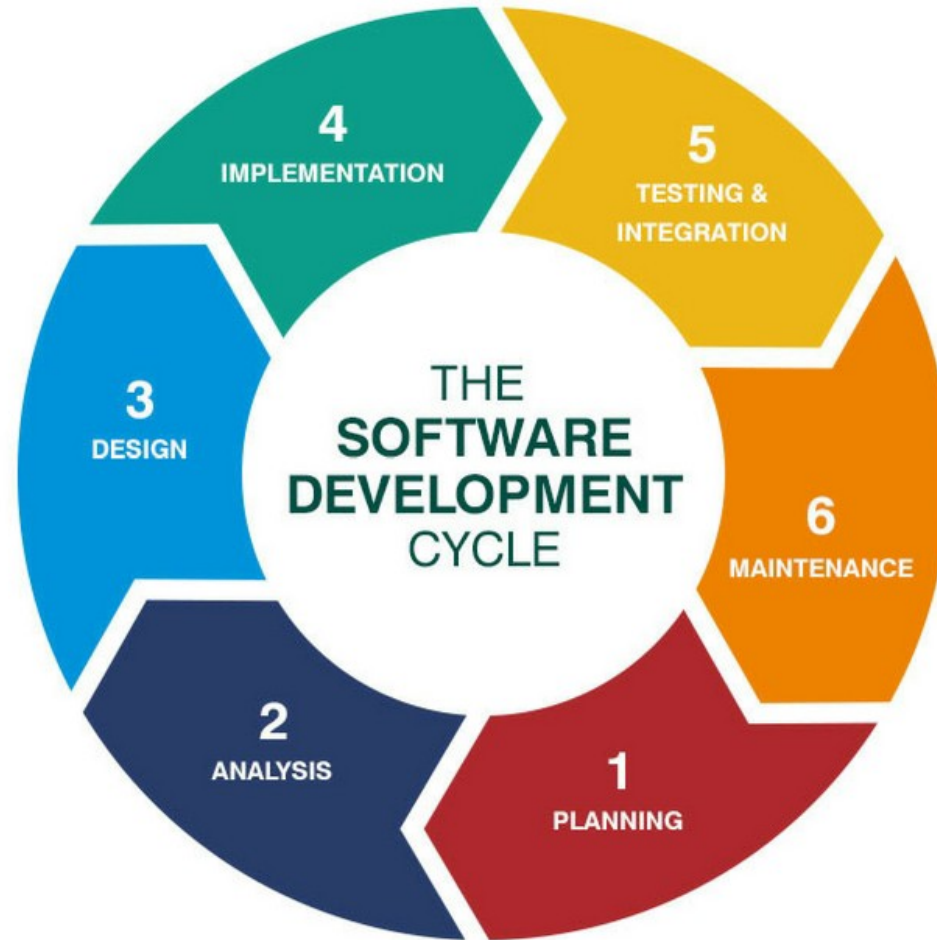
Wat zijn de fases van waterfall?



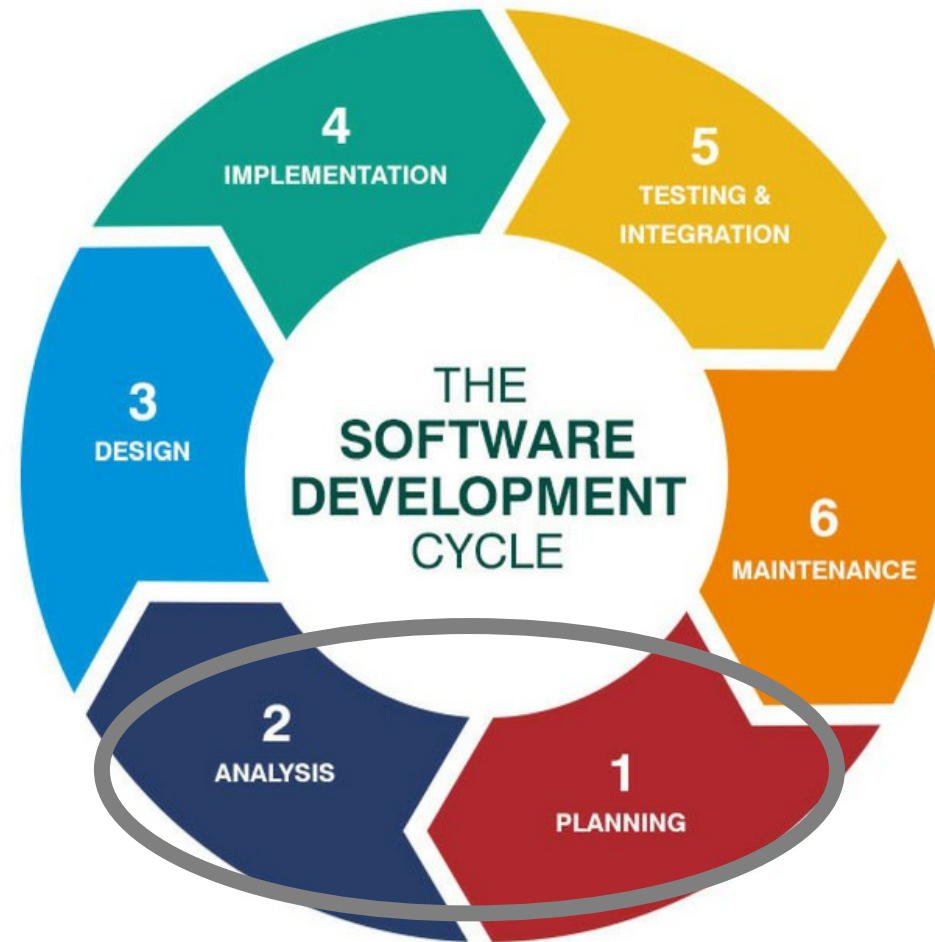
Herken je de verschillende fases van agile?



Software development lifecycle



Software development lifecycle



Software development lifecycle – Planning & Analyse

Planning

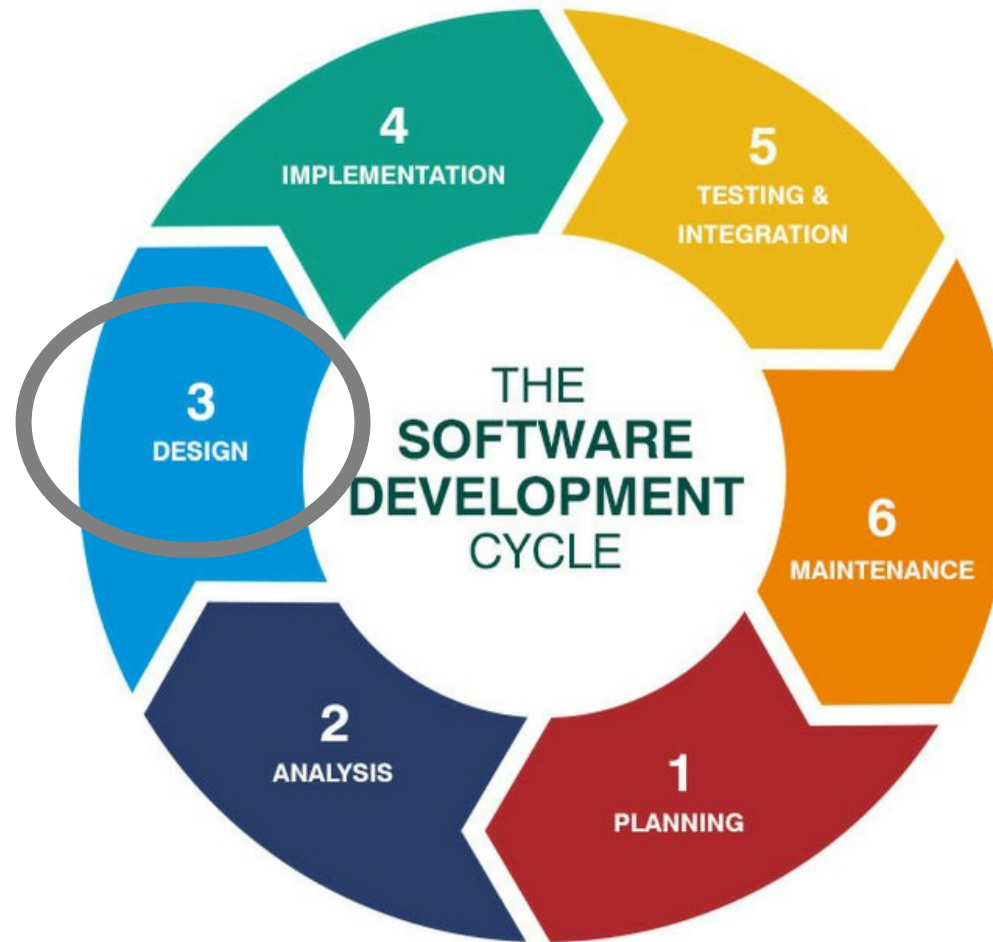
- Requirements analyseren
 - Wat moet de software kunnen?
 - Marktonderzoek
 - Kanten en eindgebruikers
 - Basis voor een haalbaarheidsonderzoek
- Taken die moeten gebeuren
 - Nadenken over hoe de kwaliteit gewaarborgd kan worden
 - Risico's van het project in kaart brengen



Analyse

- Al de requirements door de klant laten goedkeuren, duidelijk zodat er geen misverstanden zijn
- Een analyse wordt genoteerd in een SRS (Software Requirements Specification) : een document om alle functionele en niet-functionele vereisten van een product vast te leggen

Software development lifecycle



Software development lifecycle - Design

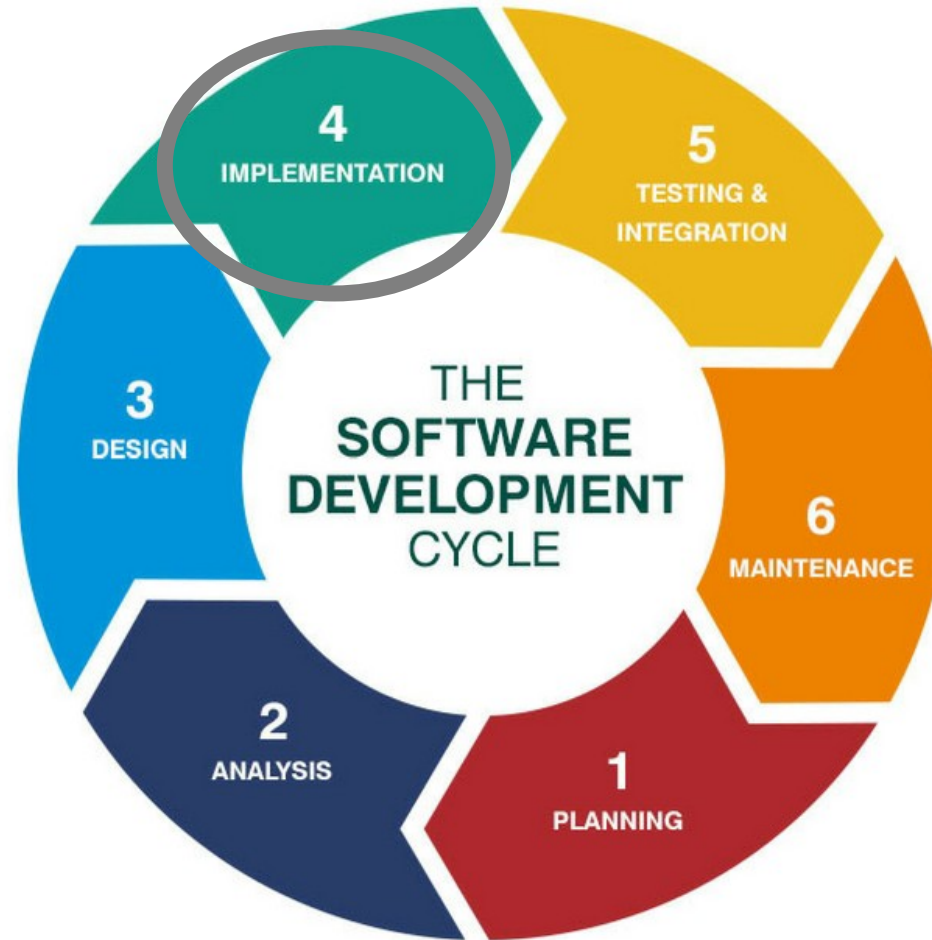
Design

- SRS vormt de basis voor de architecten
 - Meerdere mogelijke designs voorgesteld
 - Genoteerd in: Design Document Specification
 - Een ontwerpdocument is een meer gedetailleerde beschrijving van het product of de software (met beschrijvingen van schermen, benodigde data of tabellen, beschrijving van technische metingen om systeem security en performantie te verzekeren, ...)
- Bij agile werkvormen kunnen deze gemakkelijker gewijzigd worden



"It looks like you're the main stakeholder in this business."

Software development lifecycle

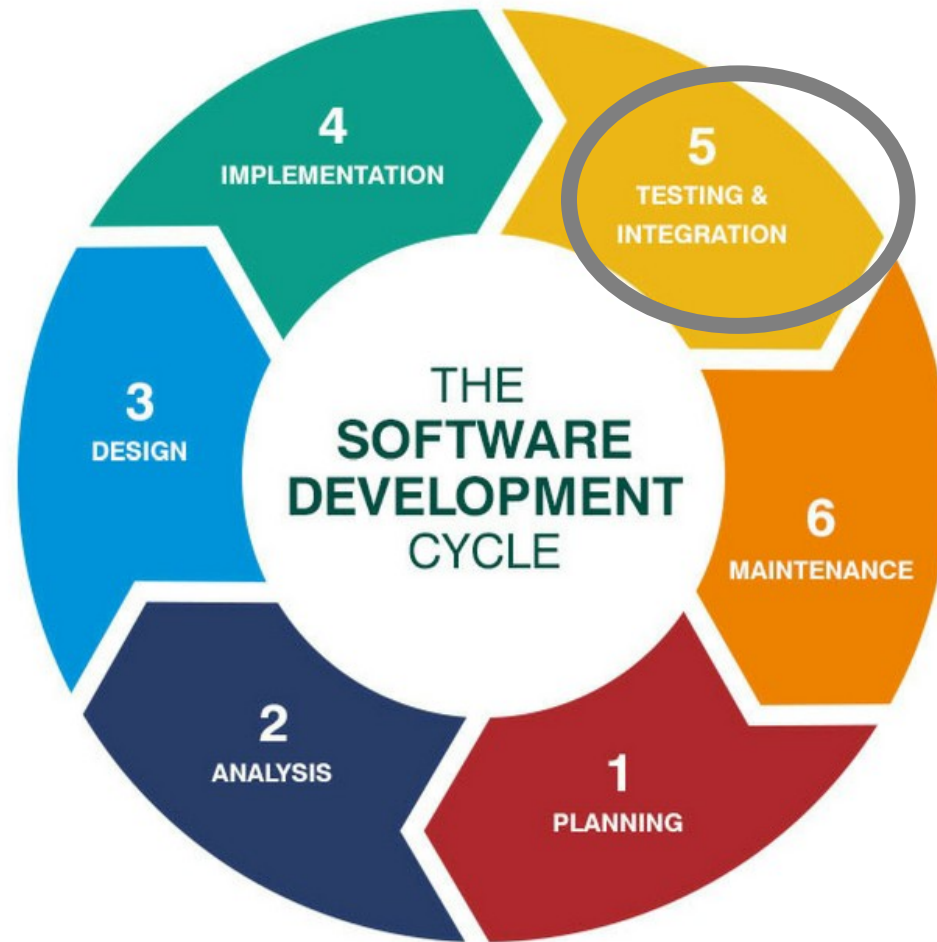


Software development lifecycle - Implementatie

Implementatie

- DDS vormt de basis voor de programmatie
 - Naast DDS kunnen er ook coding guidelines zijn
 - Naamgeving van klassen, naamgeving tests, structuur code, aantal code reviewers,,,,
- Hier wordt er gebruik gemaakt van de ontwikkelomgevingen
- Bij agile werkvormen is het de bedoeling om het einde van iedere sprint naar productie te gaan
 - Klant heeft dan een direct voordeel
 - Development -> Test -> Acceptatie -> Productie
 - In de praktijk gebeurt dit soms niet iedere sprint
 - Niet zuivere agile werkvorm
- Bij waterval gaat men op het einde van implementatie naar de testomgeving

Software development lifecycle

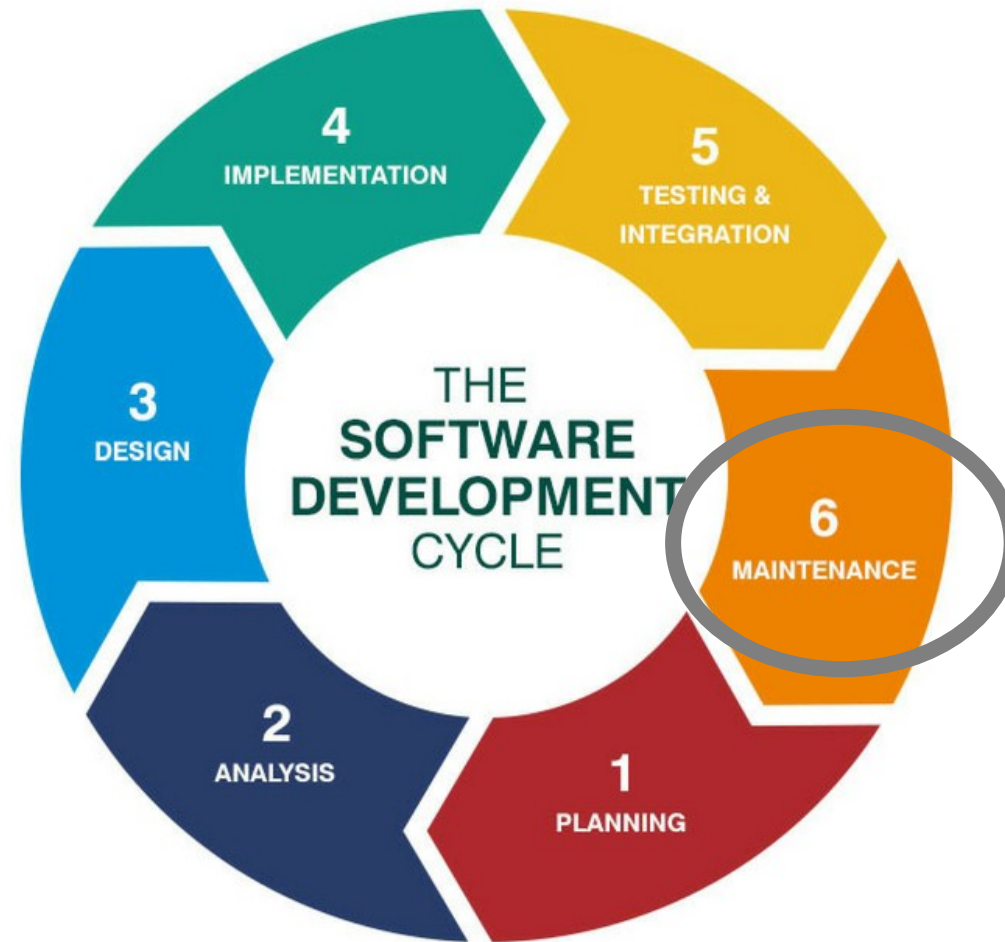


Software development lifecycle - Testen

- **Testen** na ontwikkeling is een belangrijke fase.
- Afhankelijk van of je met een klassieke projectmethode of agile werkt, kan testen op verschillende momenten ingepland worden in je project
 - Bij agile werken in elke iteratie
 - In klassieke ontwikkeling na de ontwikkelfase
- Doel van testen is kwaliteit na te streven
 - Testen of de requirements, opgesteld in analysefase, voldaan zijn (nakijken SRS).
 - Testen is functioneel (werkt alles zoals verwacht) zowel als technisch (bv performantie, ...)
 - Testen gebeurt best door verschillende mensen en vanuit verschillende invalshoeken.
 - Testen is een tijdsintensief en mag niet onderschat worden.
- Verschillende soorten testen:
zie hoofdstuk over omgevingen.



Software development lifecycle



Software development lifecycle - Maintenance

- Het product is klaar en gaat naar productie
 - Alle requirements uit SRS geïmplementeerd en getest
 - Alle processen en integratie met andere toepassingen is getest
 - Het product kan “in productie” gezet worden. Men zegt ook “gereleased” worden. Het wordt overgezet naar productie-omgeving.
- Soms wordt een product gefaseerd uitgerold
 - Bv. eerst naar een kleinere doelgroep. Men test zo of de in productiestelling goed werkt, kan feedback van eindgebruikers capteren en nog aanpassingen uitvoeren alvorens het voor een groot publiek open te stellen.
 - Bv. een deel van het product, om meer controle te hebben over de impact of het risico dat er zaken fout lopen
- Na een garantieperiode start dan de **maintenance**. Er worden afspraken gemaakt met de klant over het onderhoud van het product (bv. een jaarlijks budget voor het oplossen van fouten en kleine aanpassingen, of betalen per aanvraag, of een budget voor het oplossen van bugs maar niet voor nieuwe functionaliteiten...)

Ontwikkelomgevingen

Software development lifecycle

Systeemontwikkeling

Hoofdstuk 3

Systeemontwikkeling

Inleiding: wat is systeemontwikkeling

- Wat is een systeem?
 - Computersysteem
 - Informatiesysteem
 - Besturingssysteem
- Systeemontwikkeling
 - Ontwerpen en construeren van een nieuw systeem
 - Aanpassen van een bestaand systeem

Wat is een systeem?

- Definitie

“Een verzameling componenten die op een geordende manier samenwerken om een bepaald doel te bereiken.”

Bv.:

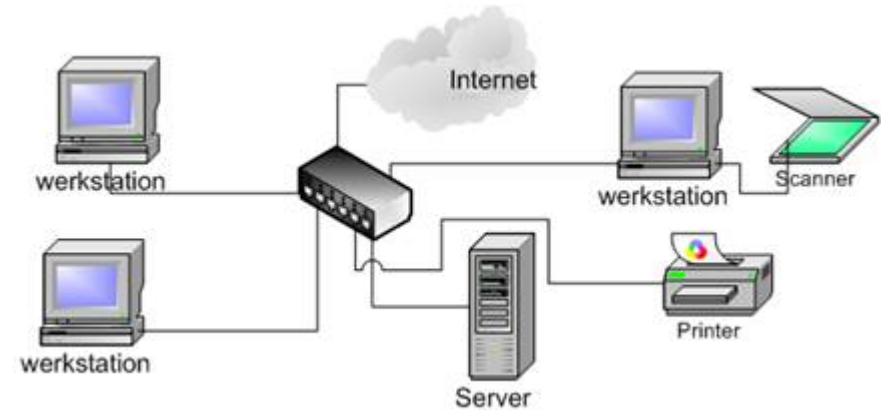
Camera
beveiligingssysteem



Een besturings-
systeem



Een LAN (Local Area Network)



Informatiesysteemanalyse en ontwerp = ?

- Het proces van ontwikkelen en onderhouden van een informatiesysteem
- Doel = verbeteren van organisatorische systemen door gebruik van software

De systeemanalist



Onderzoek vacatures in een organisatie
voor systeemanalist

Primaire rol van de systeemanalist:

- *Problemen en behoeften* van een organisatie bestuderen
- Uitzoeken hoe de organisatie kan *verbeterd* worden met de inzet van mensen, methoden en informatietechnologie
- De *eindgebruikers* in functionele afdelingen *helpen bij het definiëren van hun eisen* voor nieuwe of uitgebreidere informatiesystemen

Een systeemanalist is een teamworker

- Een ploeg met vertegenwoordigers van alle belangengroepen:
 - Eindgebruikers
 - Databasebeheerders
 - Opleidingsverantwoordelijken, ...
- Kenmerken van succesvolle teams:
 - Diversiteit
 - Tolerantie
 - Duidelijke en volledige communicatie
 - Vertrouwen gebaseerd op respect
 - Teambelang boven eigen ideeën kunnen stellen
 - Beloningsstructuur die gedeelde verantwoordelijkheid bevordert
 - Kwetsbaarheid



Vaardigheden van een systeemanalist

- Analytische vaardigheden:
 - Vooral systeemdenken
 - In detail → documenteren, uitwerken, passen
 - Rekening houden met alle aspecten van een systeem
- Technische vaardigheden:
 - De mogelijkheden en beperkingen van informatietechnologie verkennen
- Leidinggevende vaardigheden:
 - Nodig voor het beheren van projecten, middelen, risico's en veranderingen
- Communicatieve vaardigheden:
 - Eindgebruikers interviewen → **Durven doorvragen**
 - Tolk spelen tussen gebruikers en programmeurs
 - Goed naar anderen kunnen luisteren
 - Schriftelijk en mondeling rapporteren

De systeemontwikkelingomgeving – Andere analisten

- Welke andere analisten ben je tegengekomen in je zoektocht naar de functiebeschrijving van een systeemanalist?
- Wat is het verschil tussen deze soorten analisten?

EINDE

En nu herhalen en blokken... succes voor je examen!

