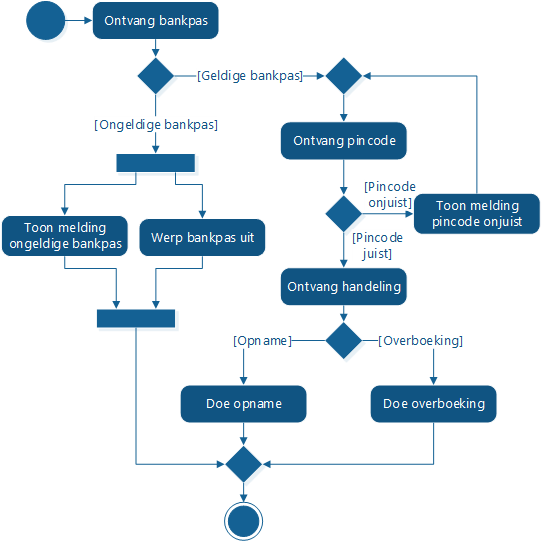
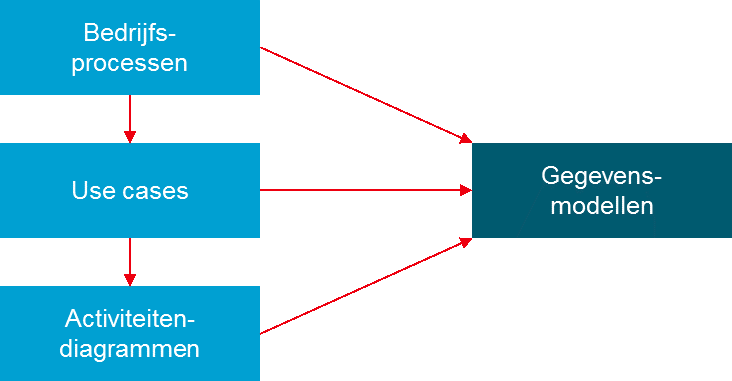
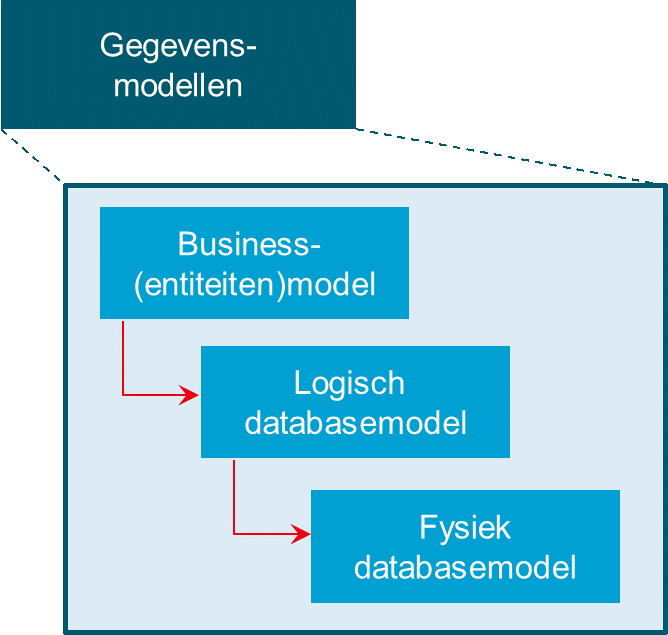
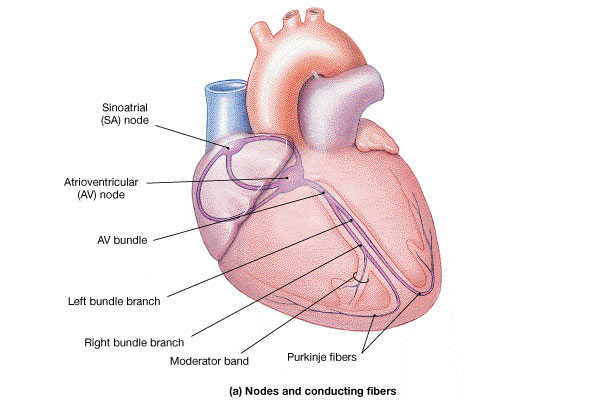
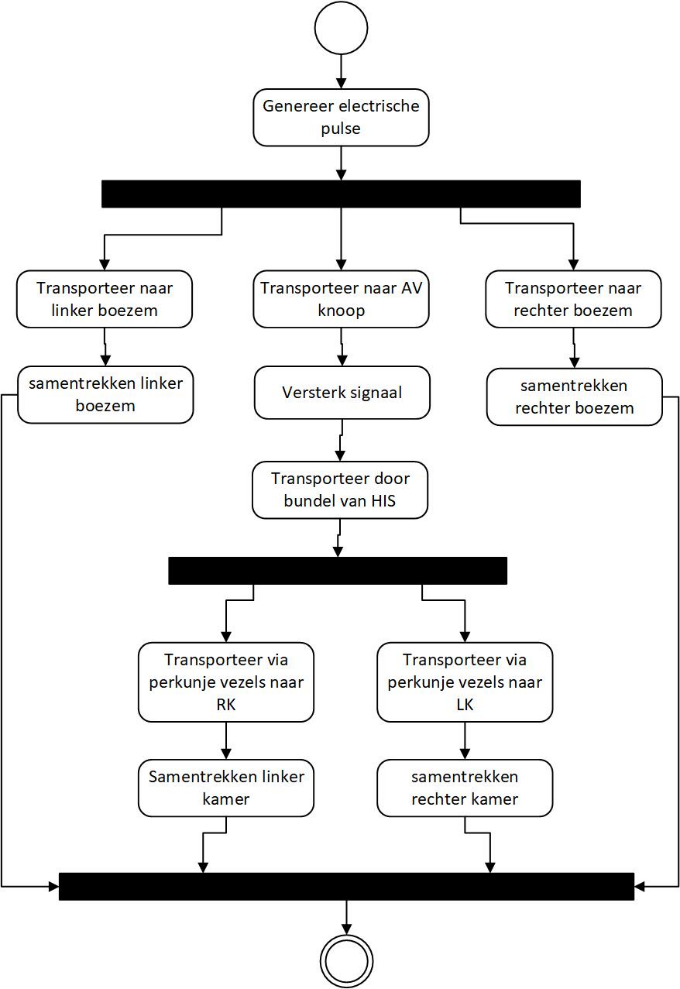
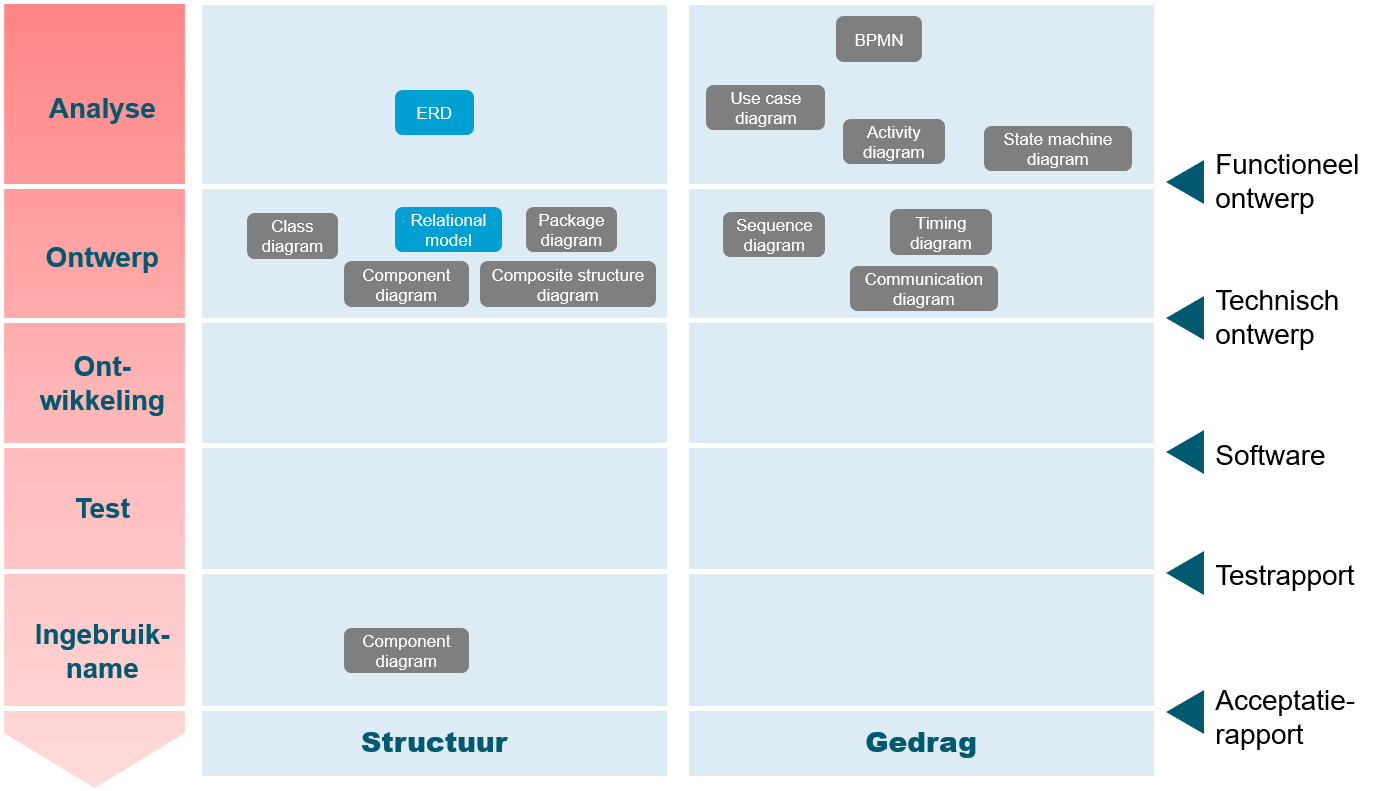
Samenvatting theorie MOD1, periode A  
  
1.1 inleiding  
1.1.1  
Een model is een schematische weergave van de werkelijkheid.  
Bij modelleren gaat het om het maken van modellen.  
  
Voorbeeld plaatje:   
  
ICT’er maakt modellen om te communiceren over zijn ontwerp, net zoals een architect maquettes en tekeningen maakt. Biedt structuur, vergelijken van opties gaat makkelijker, leggen afspraken vast.  
  
Twee modellen die belangrijk zijn: structuur en gedragsmodellen.  
  
- Structuurmodel: Statisch, beschrijft de samenhang tussen onderdelen en geven inzicht in de structuur van een systeem.  
  
- Gedragsmodellen: Zijn dynamisch, beschrijven gedrag en geven inzicht in hoe iets werkt.  
  
ICT modellen weergegeven met een diagram.  
Iedere diagram heeft een:  
- Syntax: Welke symbolen er worden gebruikt.  
- Semantiek: Wat de betekenis van de symbolen is.  
  
  
1.1.2  
Doel van dit vak: modellen die in de ICT gebruikt worden te leren kennen en toe te passen.  
Focus op modelleren van data/gegevens.  
We leren een database te ontwerpen, te maken, te vullen met data en de database te bevragen.  
  
Volgende onderwerpen komen aanbod:  
Entity relationship diagram (ERD)  
Relational database management system (RDBMS)  
Conceptueel datamodel  
Logisch en fysiek databasemodel  
SQL  
1.1.3  
- Context onderzoeken en het probleem afbakenen (bedrijfsprocessen)  
Verdiepen in de omgeving waarin de oplossing wordt gebruikt  
  
- Te ondersteunen activiteiten vaststellen (use case en activiteitendiagram)  
Hoe de oplossing gebruikt zal worden.(use case) Welke activiteiten hebben ondersteuning nodig van een ICT’er z’n oplossing. (activiteitendiagram)  
  
- Benodigde gegevens inventariseren en structureren (gegevensmodellering)  
Welke gegevens er opgeslagen moeten worden, structuur brengen aan de gegevens zodat ze efficiënt en doelgericht in een database kunnen worden opgeslagen.  
  
Plaatje als voorbeeld:  
  
Producten (modellen) die de aanpak van de ICT’er oplevert bij het ontwikkelen van een ICT-oplossing.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Gegevensmodellen hebben drie verschillende soorten, zoals te zien is op de volgende afbeelding:  
  
  
- Business(entiteiten)model (conceptueel datamodel)  
Beschrijven van alle begrippen (entiteiten) die relevant zijn voor de ICT oplossing.  
Gedefinieerd in begrijpelijke taal voor iedereen.  
  
- Logisch datamodel  
Formele notatie hoe verschillende entiteiten met elkaar samenhangen.  
Soort relaties, kenmerken van entiteiten en de eigenschappen van de kenmerken.  
Iedere entiteit opgeslagen in een eigen tabel, kenmerken vormen de kolommen van de tabel.  
Specificeert hoe unieke rijen in de tabel worden geïdentificeerd. (Primaire sleutel)  
Kan worden geïmplementeerd op ieder willekeurige (relationeel) databaseplatform.  
  
- Fysiek datamodel  
Vertaling van het logische datamodel naar één specifieke databaseplatform.  
Omschrijft hoe de gegevens van de ICT-oplossing op dit specifieke platform kunnen worden opgeslagen.  
Bijvoorbeeld: Hoeveel bytes er voor een tekstveld worden gereserveerd, welke datatype er voor een numeriek veld wordt gebruikt.  
  
Verschillende soorten relationele database (formeel: relational database engines)   
Bekende voorbeelden: MySQL, PostgreSQL, SQLite, SQL server en Oracle.  
Welk platform wordt gekozen hangt af van veel factoren.  
  
  
  
  
  
  
  
  
  
  
1.1.4  
In dit onderdeel van het vak leren we te modelleren voor ICT-oplossingen.   
We krijgen een introductie over wat een database is.  
We leren over verschillende gegevensmodellen.  
Daarna leren we ook de querytaal (SQL) om met de database te kunnen ‘praten’  
  
  
1.2 Wat is een model?  
1.2.3  
Abstracties worden gebruikt om de belangrijkste aspecten te benadrukken.  
Bijvoorbeeld een tekening vereenvoudiging van een auto of gebouw, er worden bepaalde details weggelaten die degene die de tekening leest niet hoeft te weten.  
Het gaat hier om een abstractie, een model van een auto of gebouw. Pas als het gebouwd is wordt het abstracte model concreet.   
  
Niet alle aspecten van een model van een gebouw staan op een en dezelfde tekening. Voor ieder doel wordt een aparte tekening gemaakt in de ‘taal’ van dat doel.  
Een elektricien wil bijvoorbeeld weten waar meters, stopcontacten en bliksemafleiders zijn geplaatst.  
Voor degene die beton komt storten is dat minder van belang, die wil weten hoe dik muren moeten worden en waar de betonvloer gestort moet worden.

Een model beschrijft dus die aspecten van de werkelijkheid die belangrijk zijn voor een specifiek gebruiksdoel.  
  
  
1.3 Waarom modelleren?  
  
- Grip krijgen op het onderzoeksdomein  
Opdelen van complexe problemen in deelaspecten. Zo wordt het probleem in behapbare onderdelen opgedeeld.  
  
- Structureren van een aspect.   
Bij data-modellering worden bijvoorbeeld entiteittypen, relaties tussen entiteittypen en attributen onderscheden.  
  
- Stepwise refinement.  
Elke fase van systeemontwikkeling is een iteratief proces. Een model is uitermate geschikt om te wijzigen ivm nieuwe informatie en voortschrijdend inzicht.  
  
- Communicatie  
Modellen geven overzicht over aspecten. Ze kunnen met enige uitleg besproken en beoordeeld worden.   
  
- Compacte vorm van documentatie  
Via plaatjes kun je veel specifieke informatie kwijt zoals het eerste voorbeeld met de bankpas.  
  
  
- Eenvoudig om alternatieven aan te geven  
Twee modellen gebruiken om te vergelijken is makkelijker dan het gebruiken van twee tekstuele verhalen.  
  
- Eenduidig vastleggen van afspraken.  
Modellen forceren, mits juist gebruikt, consistentie en eenduidigheid en zijn daarom geschikt.  
  
  
1.3.1  
Voorbeeld van een structuurmodel zoals in 1.1.1 wordt besproken.  
Structuurmodel  
  
Voorbeeld van een gedragsmodel zoals in 1.1.1 wordt besproken.  
Gedragsmodel  
  
  
1.3.2  
Een model heeft alleen betekenis als het wel-gevormd is zoals een zin in de Nederlandse taal.  
  
De diagrammen die in de ICT worden gebruikt hebben altijd een eenduidige semantiek.  
De taal van dergelijke diagrammen noemen we een formele taal.  
  
De factostandaard voor modellen in de ICT is UML (unified modelling language)  
  
  
1.3.3  
Wanneer een team de opdracht krijgt een applicatie te ontwikkelen wordt een aantal fasen doorlopen.  
  
Ten eerste wat de applicatie precies moet doen, de analyse fase, het functioneel ontwerp wordt beschreven.  
Vervolgens wordt er een ontwerp gemaakt waarbij de nadruk ligt hoe de functionaliteit wordt gerealiseerd.  
Daarna vindt de realisatie plaats, worden de producten uitvoerig getest en mits goedgekeurd in productie genomen.  
  
Plaatje is een overzicht van modellen per fase.  
  
  
De blauw gekleurde modellen worden behandeld in het onderdeel modelling.  
  
  
  
  
  
  
  
  
1.4 Introductie database  
1.4.1  
Een database is een systeem dat functioneert als opslagplaats voor informatie.  
  
1.4.2  
Een database is een opslagplaats voor data.  
De voornaamste taak van een database is om grote hoeveelheden gegevens efficiënt, consistent en duurzaam op te slaan en deelverzamelingen op de gewenste manier te verstekken aan gebruikers of toepassingsprogramma’s   
  
Een databasesysteem bestaat uit gegevensopslag (één of meerdere databases) en een beheersysteem (het database management systeem, DBMS)  
  
1.4.3  
“Een database is een collectie van persistente data.”

Het woord persistent geeft hierbij aan dat de data gedurende een zekere tijd worden opgeslagen in het permanente geheugen van een computersysteem. Daarmee bedoelen we het geheugen waarvan de inhoud niet verloren gaat bij het uitschakelen van het systeem.”  
  
1.4.4  
“Een DBMS is een softwaresysteem voor het beheer van databases.”

Alle interactie met een database gebeurt via het DBMS: als een gebruiker data wil opzoeken, toevoegen, wijzigen of verwijderen 2 of een database wil aanmaken of wijzigen.  
  
1.4.5  
Databases zijn een essentieel onderdeel van de informatiemaatschappij.  
Gewoonlijk gebruiken toepassingsprogramma’s databases om de gegevens op te slaan. Hier een paar toepassingen uit de praktijk die je misschien al bent tegen gekomen.

* Enterprise Resource Planning Systemen (ERP), zie vak BIM (periode B).
* Customer Relationship Management Systemen (CRM), zie vak BIM (periode B).
* Business Intelligence (BI) applicaties, zie vak BIM (periode B).
* OSIRIS, het onderwijsinformatiesysteem van de Hogeschool Utrecht (HU). De naam staat voor Onderwijs en Studenten Informatie, Registratie en Inschrijf Systeem.
* Studielink, het landelijke informatiesysteem van de Nederlandse hogescholen en universiteiten voor aanmeldingen voor een opleiding.
* DigiD, het systeem voor inloggen op websites van de overheid en in de zorg. Je DigiD bestaat uit een gebruikersnaam en een wachtwoord die je allebei zelf kiest.
* Op de website www.verlorenofgevonden.nl/ kun je zoeken als je iets verloren hebt of melden dat je iets gevonden hebt. De gegevens worden in een database opgeslagen.

1.4.6  
De belangrijkste functies van een databasesysteem zijn:

1. een grote hoeveelheid gegevens opslaan, wijzigen en verwijderen
2. zeker stellen dat gegevens correct en compleet zijn
3. werkzaamheden van meerdere gebruikers tegelijk mogelijk maken
4. gegevens zo snel mogelijk zoeken en aan gebruiker teruggeven
5. structuren aanmaken om gegevens goed op te slaan
6. datazekerheid en dataprotectie garanderen
7. beheer en uitbreiding van de database faciliteren

1.4.7

De zes pijlers van databases zijn:

1. beschikbaarheid – per gebruiker, maar eveneens per systeem  
   het nieuwe doel: “anywhere, anytime, anyone”
2. beheerbaarheid – overzicht van de database prestatie, proactieve waarschuwing als er b.v. niet meer genoeg opslagruimte is, inspelen van software updates bij lopende database
3. herstelbaarheid – bij database of server crashes, ook herstel van menselijke fouten
4. consistentie – onderling verband van data afdwingen door regels
5. prestaties – sterkere hardware ingezet voor centraal gebruikte toepassingen, configureerbaar voor speciale toepassingen (big data, GIS)
6. schaalbaarheid – scale up (“dikkere server”)/scale out (“groter serverpark”)