De **Unified Modeling Language (UML)** is een van de meest gebruikte methodes om de visies van systeemontwikkelaars begrijpelijk en gestandaardiseerd vast te leggen. Het laat je toe om diagrammen te tekenen die begrijpelijk zijn voor de klant

UML bestaat uit een aantal grafische elementen die tot diagrammen worden gecombineerd. Omdat het een taal is, bevat UML regels voor het combineren van deze elementen.

UML is ontstaan begin jaren ’90 door Grady Booch, James Rumbaugh en Ivar Jacobson en is in 1997 als standaard opgenomen.

Één van de grote voordelen van UML is de flexibiliteit. En dat het toegepast kan worden in zeer grote architectuursystemen of op een simpele kleine systemen. Voorbeelden hiervan zijn:

* Business model processen
* Het laten zien van een applicatie structuur
* Het omschrijven van een applicatie structuur
* Het gedrag van waarnemen van een systeem
* Moduleren van een data structuur
* Het schetsen van ideeën
* Een gedetailleerde specificatie bouwen van een systeem\
* Het generen van programmeercode

Van de onderstaande 10 UML diagrammen zullen op de opleiding van Inholland UML course, 5 van de 10 worden behandeld(rechtse kolom).

* Deployment diagram - activiteit diagram
* Component diagram - use-casses
* Communicatie diagram - klassen diagram
* Package diagram - toestand diagram
* State diagram - sequence diagram

Ook zal het 4+1 model worden behandeld, dit is een model die architectuursystemen afbreekt in 5 verschillende views. Dit zorgt ervoor dat je beter in staat bent om de rollen van de UML te begrijpen.

Elke view van het model heeft een bepaalde functie en UML diagrammen geassocieerd, hieronder ziet u een overzicht van de verschillende views met de bijbehorende functie en diagrammen.

**Logical view**

Logical view zorgt ervoor dat onderdelen van het architectuursysteem te zien is en de interactie daarvan. Ook vertegenwoordigd het concepten of ideeën. De UML diagrammen die voor deze view gebruikt worden zijn:

* klassen diagram
* toestand diagram
* object diagram
* sequence diagram
* Communicatie diagram

**Process view**

Process view beschrijft de processen van een architectuursysteem en de communicatie tussen deze processen. Ook verdiept process view zich naar wat er moet gaan gebeuren binnen het systeem, dit bij elkaar is zeer nuttig voor bij meerde processen. Het UML schema dat hiervoor wordt gebruikt is de activity schema.

**Physical view**

Physical view wordt de execution environment geregeld zoals een operating system of een database manage systeem. Hierbij wordt de deployment diagram van UML gebruikt.

**Development view**

Development view omschrijft de modules of componenten van het architectuursysteem, hieronder vallen packages, sub systemen en klasse bibliotheken hierdoor krijg je een bouwblok overzicht van het systeem, dit is nuttig om de verschillende systeem lagen te zien. Componenten en package schema’s worden hiervoor gebruikt.

**Use-case view**

Laat de functionaliteit zien van wat een systeem hoort te doen en hoe het bekeken wordt door de buitenwereld. De use-case dient als een guideline voor de andere 4 model views en vangt gebruikersdoelen op. En natuurlijk hoort hierbij de use-case diagram.

**Statische vs Dynamische benadering van systeemontwikkeling**

Een static model laat zien hoe een systeem in elkaar zit of een stuk van het syeteem. En zorgt voor de nadruk van deze onderdelen die het geheel maken tot een systeem. Ook geeft het de exacte mening over class names, attributes, method signatures en packages. De volgende UML diagrammen worden gebruikt voor static models: class diagram, object diagram en use case diagram.

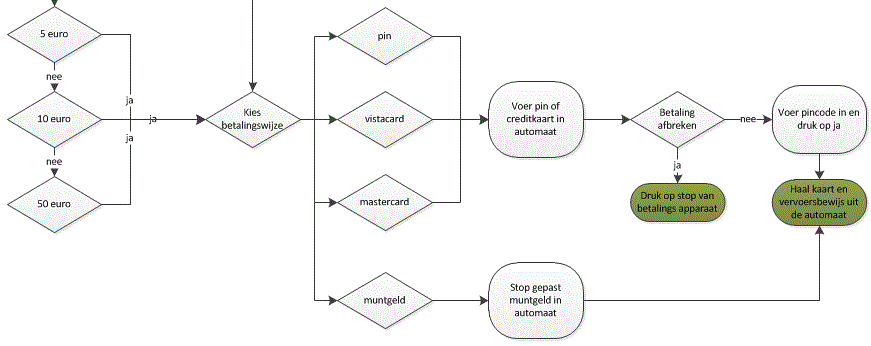
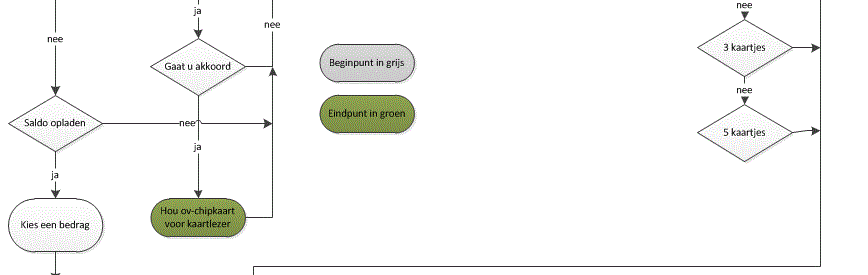
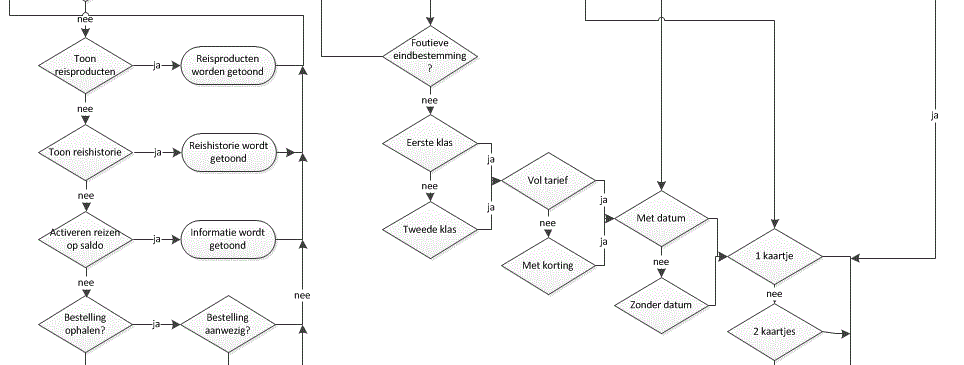
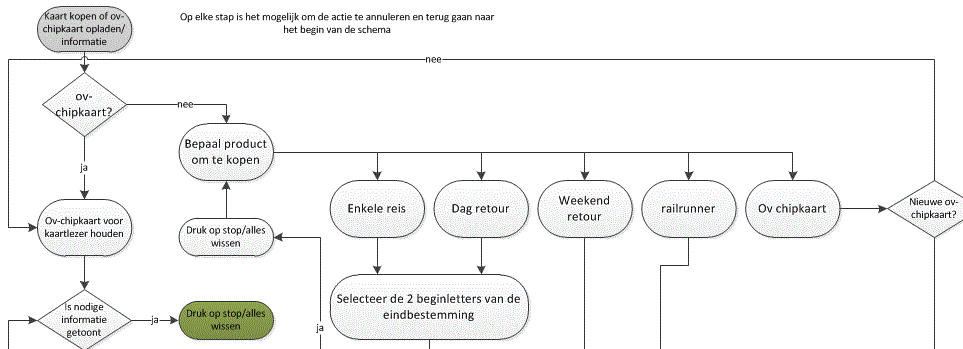
Een Dynamische model laat zien een systeem zich gedraagt, hierdoor vind je welke objecten je nodig heb en samen laten werken doormiddel van methodes en berichten. Een Dynamische model wordt gebruikt om de logica en gedrag van een systeem te ontwerpen. Dynamische model maakt gebruik van de sequence, communication, state en activity diagrammen.

Een implementation model is een gespecialiseerde onderdeel van static model dit laat het verschillen zien die een systeem nodig heeft voor het uitrollen. Ook laat het de fysieke componenten zien die nodig zijn zoals hardware en kabels, component en deployment diagrammen worden hier voor gebruikt.

Taakanalyse van NS automaat (papieren kaartje kopen enkele reis)

* Bepaal reisproduct
* Selecteer 2 begin letters van eindbestemming
* Kies eind bestemming
* Maak een keuze: eerste of tweede klas
* Maak een keuze: met korting of zonder
* Maak een keuze: met of zonder datum op het kaartje
* Maak een keuze voor het aantal kaartjes
* Bepaal op welke manier te betalen
* Betaal

Bij dit taakanalyse kan een activiteitenschema worden gemaakt hieronder ziet u er een voorbeeld van:



Ook het bestellen van een DVD bij bol.com moet je handelingen uitvoeren. Hieronder ziet u de taak analyse en de activiteitenschema die daar bij hoort.

Voordat je een DVD kan kopen zijn de volgende voorwaarden nodig:

1. Instaat om op internet te komen
2. Een web-browser

Taakanalyse:

1. Start browser
2. Type het www adres in van bol.com
3. Account aanmaken indien u deze nog niet heeft\
4. Log in met uw account
5. Zoek naar de DVD die u wil kopen
6. Voeg deze DVD toe aan winkelmand
7. Betaal

