**Week 1**

**1. Wat is UML?**

* UML is een modelleertaal die gebruikmaakt van diagrammen waarbij standaard afspraken zijn gemaakt waardoor iedereen het kan begrijpen.

**2. Waar bestaat UML uit?**

* Notaties (elementen die samen werken: symbolen, connectors, notes, values, etc).
* Diagrammen (plaatjes van een proces of systeem).

**3. Waar komt UML vandaan? Noem de personen die aan de basis van UML hebben gestaan.**

* UML is in 1997 ontworpen en uitgegeven door de Object Management Group (OMG) om een standaard te maken.
* UML is bedacht door Grady Booch, James Rumbaugh en Ivar Jacobson.

**4. Wat is de sterke kant van UML?**

* UML is flexibel en schaalbaar.

**5. Wat bedoelen ze met “UML is schaalbaar” ?**

* UML kan meegroeien wanneer processen uitgebreider worden.

**6. Waar gebruik je UML voor. Noem er 8.**

* Bedrijfsprocessen modelleren.
* applicatiestructuur in kaart brengen.
* systeemarchitectuur beschrijven.
* systeemgedrag vastleggen.
* modeldata structureren.
* ideeën uitschetsen.
* gedetailleerde systeemspecificatie bouwen.
* programmeercode genereren.

**7. Noem de 10 soorten diagrammen en geef van elke een korte uitleg.**

* **Use case diagram :** Toont wat het systeem doet met behulp van actors.
* **Class diagram :** Definieert classes, objecten en relaties onderling.
* **Object diagram :** Laat systeemstructuur zien op een bepaald moment in de tijd.
* **Package diagram :** Bundelt elementen om een hoger level view van het systeem te laten zien.
* **State diagram :** Dynamisch beeld van het systeemgedrag.
* **Activity diagram :** Dynamisch beeld van het systeemgedrag maar dan de workflow, bedrijfsprocessen en procedurelogica.
* **Sequence diagram :** Interactie diagram, hoe objecten met elkaar praten met nadruk op tijd en de volgorde van berichtgeving.
* **Communicatie diagram :** Hoe objecten praten met elkaar, nadruk op links tussen de objecten.
* **Component diagram :** Beeld van de componenten in een systeem. Elk component vormt één of meerdere klassen.
* **Deployment diagrammen :** Brengen software in kaart hoe het met hardware praat om een fysieke lay-out van het systeem uit te beelden.

**8. Welke 5 diagrammen gaan wij gebruiken?**

* Use case diagram
* Activity diagram
* Sequence diagram
* State diagram
* Class diagram

**Vragen bij het 4+1 Model**

**1. Hoe heet het model waar men in de video over spreek?**

* Het 4 + 1 Model

**2. Leg uit waar het model betrekking op heeft.**

* Het geheel in gedeeltes opbreken om alle onderliggende diagrammen en relaties goed en overzichtelijk te kunnen bekijken.

**3. Geef een opsomming van de verschillende views.**

* **Logical** : Toont onderdelen die het systeem bevat en de interactie er tussen.
* **Process** : Toont systeemprocessen en de onderlinge communicatie.
* **Physical** : Toont hoe software de hardware gebruikt.
* **Development** : Beschrijft de modules en componenten(package, subsystems en class libraries) van het systeem. Geeft een overzicht van hoe het systeem opgebouwd is.
* **Use Case** : Toont systeem functionaliteit vanuit het buitenwereldperspectief.

**4. Welke diagrammen horen bij de verschillende views. Noem ze en bedrijf ze kort.**

* **Logical**: Class, state, object, sequence en communication diagram.
* **Process**: Activity diagram.
* **Physical**: Deployment diagram.
* **Development**: Component en package diagram.
* **Use Case**: Use Case diagram.

## Vragen bij Statische vs Dynamische benadering van systeemontwikkeling

**1. Wat laat een statisch model zien?**

* **Uitleg.** Toont de structurele eigenschappen van het systeem.
* **Wat levert het model op?** Het geeft een gedetailleerd overzicht van de onderdelen waar het systeem uit bestaat.
* **Welke diagrammen gebruik je hiervoor?** Class, Object, Use Case diagram.

**2. Wat laat een dynamisch model zien?**

* **Uitleg.** Toont de gedrageigenschappen van het systeem.
* **Wat levert het model op?** Geeft een overzicht van benodigde objecten en hoe deze met elkaar samenwerken met gebruik van methods en messages.
* **Welke diagrammen gebruik je hiervoor?** Sequence, Communication, State en Activity diagram.

**3. Wat laat een implementatie model zien?**

* **Uitleg.** Toont de benodigde elementen voor de uitrol van het systeem.
* **Wat levert het model op?** Toont de fysieke hardware zien die nodig is voor het systeem.
* **Welke diagrammen gebruik je hiervoor?** Component diagram en deployment diagram.