

## **BBM 382 Reading Assignment 3**

**Prepared by Okan ALAN – 21526638**

### **UML SOFTWARE ARCHITECTURE AND DESIGN DESCRIPTION**

UML, short for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems.

UML models are the first created thing to show software architecture.

#### **Practitioner Reflections on UML use**

Most of the practitioner use of UML in use case, logical views, component views, deployment, and scenarios.

There are a few important criteria that practitioners use to determine stopping model activities. One of them is completeness. Second is passing a review. The last one is the deadline.

There are four major problems with UML descriptions. These are respectively scattered information, incompleteness, disproportion, and inconsistency. Other problems are diagram quality, informal use, lack of modeling conventions.

There is no formal semantics for UML. The generality and freedom of UML, which is of great importance in serving the industry, is its greatest weakness at the same time. This causes communication problems among people.

#### **Defects in Industrial UML Models**

There are several case studies that display defects of UML modeling rules.

Methods that are not called in sequence diagrams: To show class interactions. The public methods must be in a sequence diagram.

Classes not occurring in sequence diagrams: A class doesn't show as a sequence diagram object.

Objects without names: We must give a name every object in the sequence diagram.

Messages not corresponding to methods: Every message corresponds to a method to explain it.

Classes without methods: Each class should contain a method for the object-oriented concept.

These defects may procreate problems in the interpretation stage.

#### **Opportunities for Improving UML Usage**

UML look like a map. UML show specific architectural styles and pattern like a path in the map. We can see which type of block in the model and what types of relations can exist between the block.

## **UML Metrics**

Software metrics are a very strong technique to manage software design and architecture. There are some architecture metrics such as class dynamicity, number of classes per use case and number of use cases per class.

Class dynamicity: We can say a class is important for the system if the class appears different sequence diagram and has many connections.

Existing tools shows metrics and defects visualized UML diagrams.

UML diagrams help us to on increase capabilities in following areas: detection of design defects and mismatch, more uniformly in modeling, Domain- or project-specific reference architectures and patterns, more consistency between UML models and system requirements as well as implementation, defined quality goals for UML models.