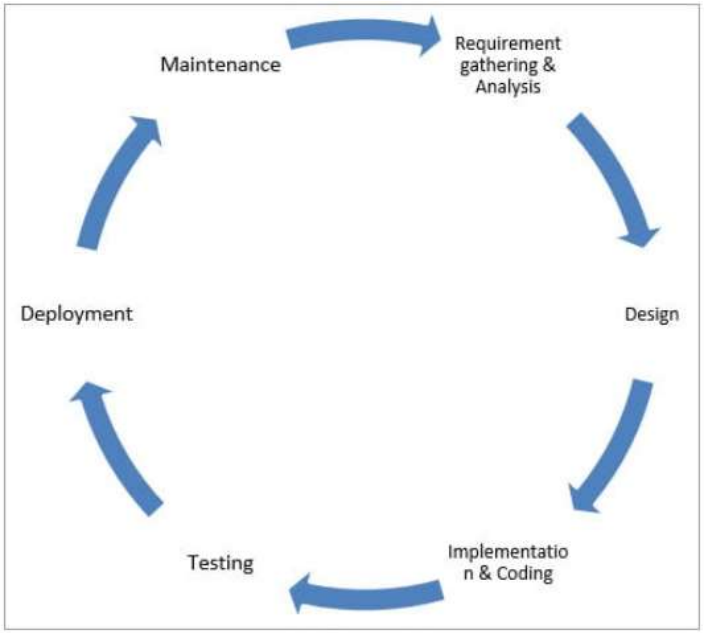
Software Modelling

SDLC (Software Development Life Cycle)

- Requirements : What to do?
- Design: How to build the code/system
- Implementation: the actual doing of the code with unit testing
- Testing: Did we meet the requirements, and make sure no problems in the code
  - System Testing is done by testers (not developers)
  - Unit testing is done by developers during coding
- Deployment: making the application work on a target device
- Maintenance: Fixing issues and upgrading with new features

- Software Modelling is the representation of the software system before coding
  - **Mainly in requirement & design phases**
- **We develop software based on object-oriented concepts**
  - **UML is an object-oriented software modelling language**



What is UML?

- The **U**nified **M**odeling **L**anguage (UML) is a general-purpose *visual modeling* ***language*** for systems

UML does *not* give any kind of modeling method. It just provides a *visual syntax* that we can use to construct models

- UML is *not* tied to any specific process model

Why “Unified”?

- Unification is not just historical in scope

UML

- Is used from requirements to implementation
- Is used to model from hard real-time embedded systems to management decision support systems
- Is language neutral and platform neutral
- Supports many software engineering processes

What is UML?

- Basic premise of UML: software and systems are modeled as a ***collection of interacting objects***
- Two aspects to UML model:
  - **Static structure** : describes what types of objects are important for modeling the system and how they are related
  - **Dynamic behavior**: describes the behavior of these objects and how they interact with each other to deliver the system functionality

Unified Modeling Language (UML)

Just a *diagramming notation* standard.

Trivial and relatively unimportant.

Not a method, process, or design guide.