

Process Models: Perspective Process Models (Continue)

Lecture # 5

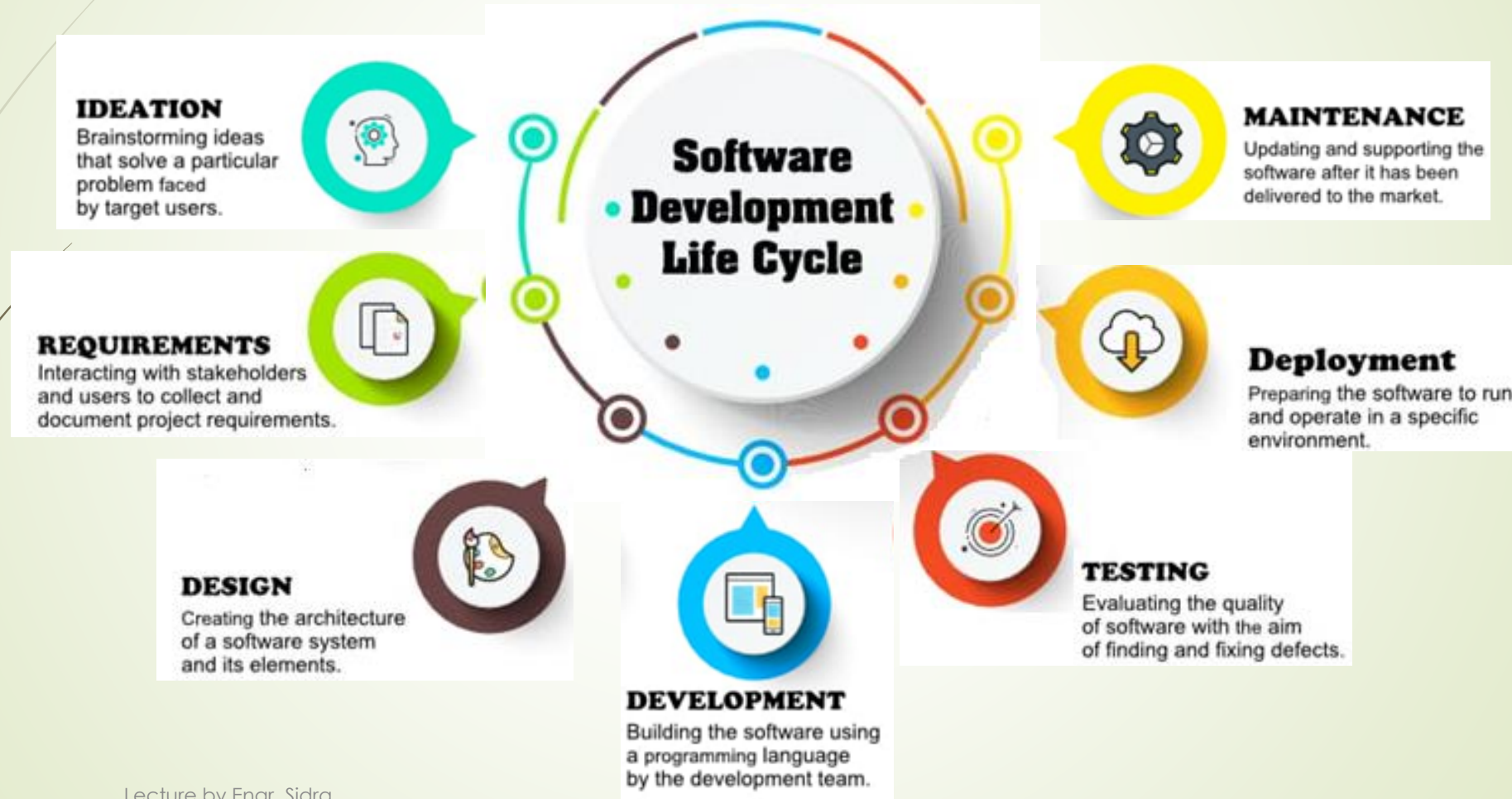


SDLC

- Software Development Life Cycle (SDLC) is a series of steps that must be followed by a development team to develop and maintain software
- **Why the software life cycle is important:**
 - It provides visibility for the engaged parties
 - It allows to control the project
 - Predictable deliveries throughout an entire development process
 - Eliminating risks like going over budget or deadline breach
 - The process goes on until all the requirements are met



SDLC Phases



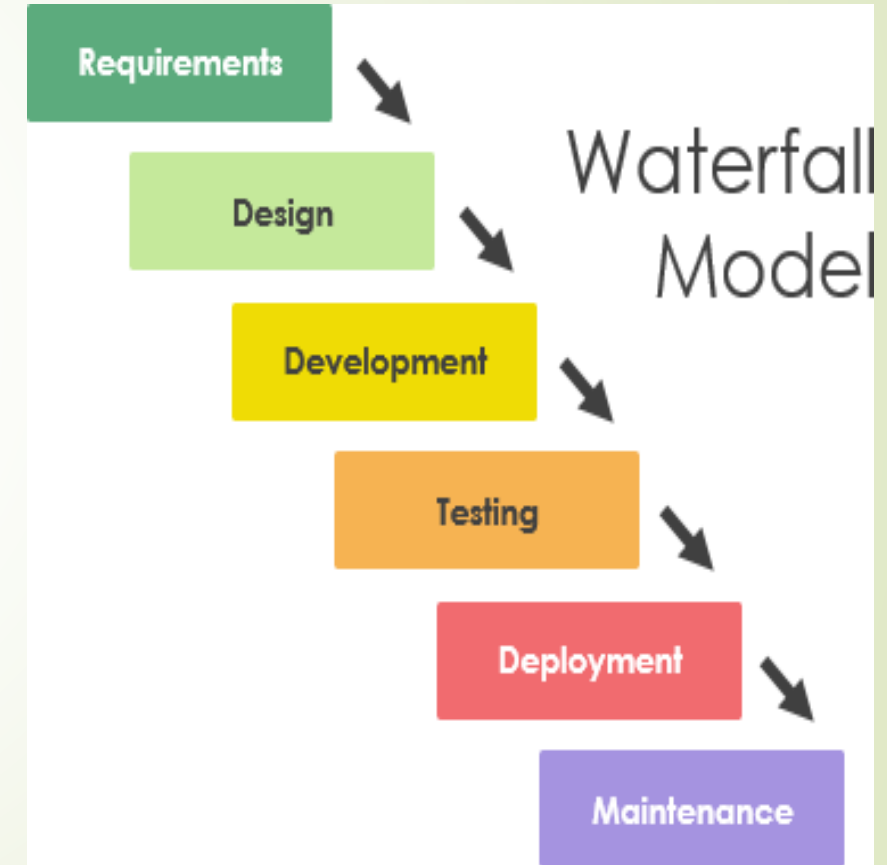
Perspective Process Model

- Perspective process models define a prescribed set of process elements and a predictable process work flow.
- They can be adapted and extended to create more specific processes.
- The most used, popular and important SDLC models are given below:
 - Waterfall model
 - V model
 - Incremental model
 - RAD model
 - Iterative model
 - Spiral model
 - Prototype model



Waterfall Model

- The waterfall model is also called as '**Linear sequential model**' or '**Classic life cycle model**'.
- In this model, each phase is fully completed before the beginning of the next phase.
- Each phase depends on the deliverables of the previous one and corresponds to a specialisation of tasks.
- Feedback is taken after each phase to ensure that the project is on the right path.
- Testing part starts only after the development is complete.
- It is used for the small projects.



Waterfall Model - Application

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- Ample resources with required expertise are available to support the product.
- The project is short.



Advantages of waterfall model

- Simple and easy to understand and use
- All the requirements are known at the beginning of the project, hence it is easy to manage.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.



Advantages of waterfall model

- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.
- This model is preferred for those projects where the quality is more important as compared to the cost of the project.



Disadvantages of Waterfall model

- This model is not good for complex and object oriented projects.
- Poor model for long and ongoing projects.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.



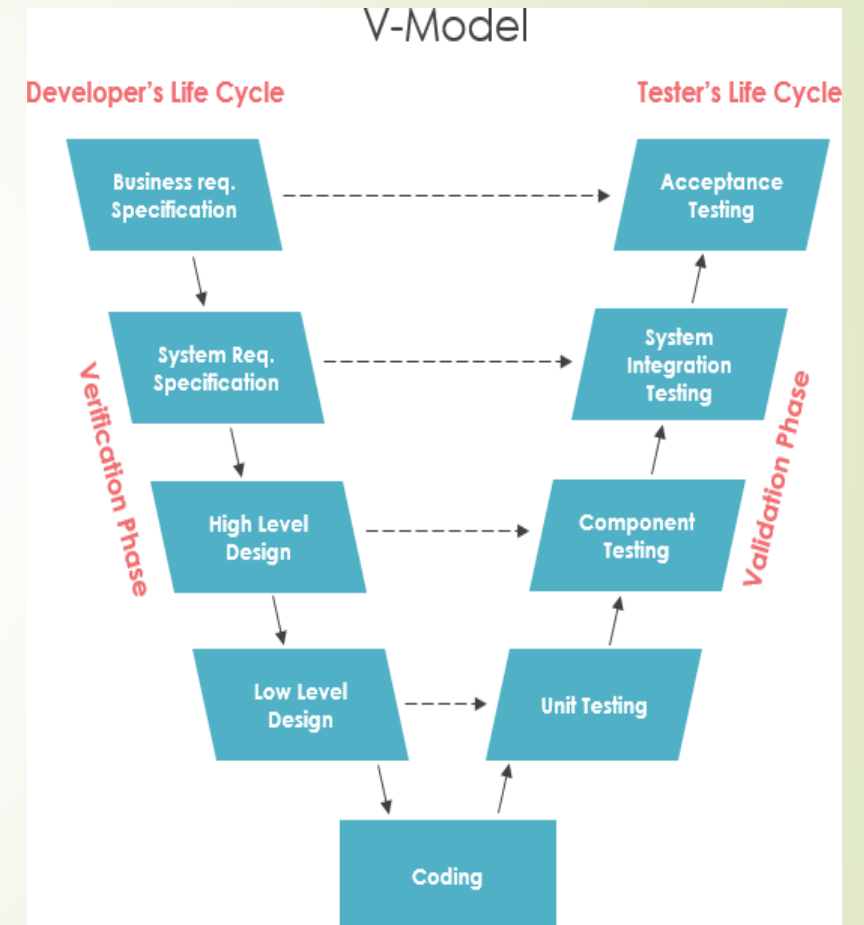
Disadvantages of Waterfall model

- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang" at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.
- The problems with this model are uncovered, until the software testing.



V-Model

- The V-model represents a development process that may be considered an extension of the waterfall model.
- It is also known as **Verification and Validation model**.
- The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing.
- Under the V-Model, the corresponding testing phase of the development phase is planned in parallel.
- So, there are Verification phases on one side of the 'V' and Validation phases on the other side. The Coding Phase joins the two sides of the V-Model
- This is a highly-disciplined model and the next phase starts only after completion of the previous phase



V-Model - Applications

- Requirements are well defined, clearly documented and fixed.
- Product definition is stable.
- Technology is not dynamic and is well understood by the project team.
- There are no ambiguous or undefined requirements.
- The project is short.



Advantages of V- Model

- This is a highly-disciplined model and Phases are completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
- Proactive defect tracking – that is defects are found at early stage.
- Avoids the downward flow of the defects.



Disadvantages of V-Model

- High risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.
- Once an application is in the testing stage, it is difficult to go back and change a functionality.
- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated.

