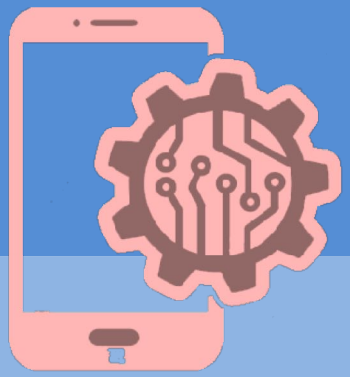
Lecture 1.3.1
Spiral & RAD

Introduction to Software Engineering



Different Process Models

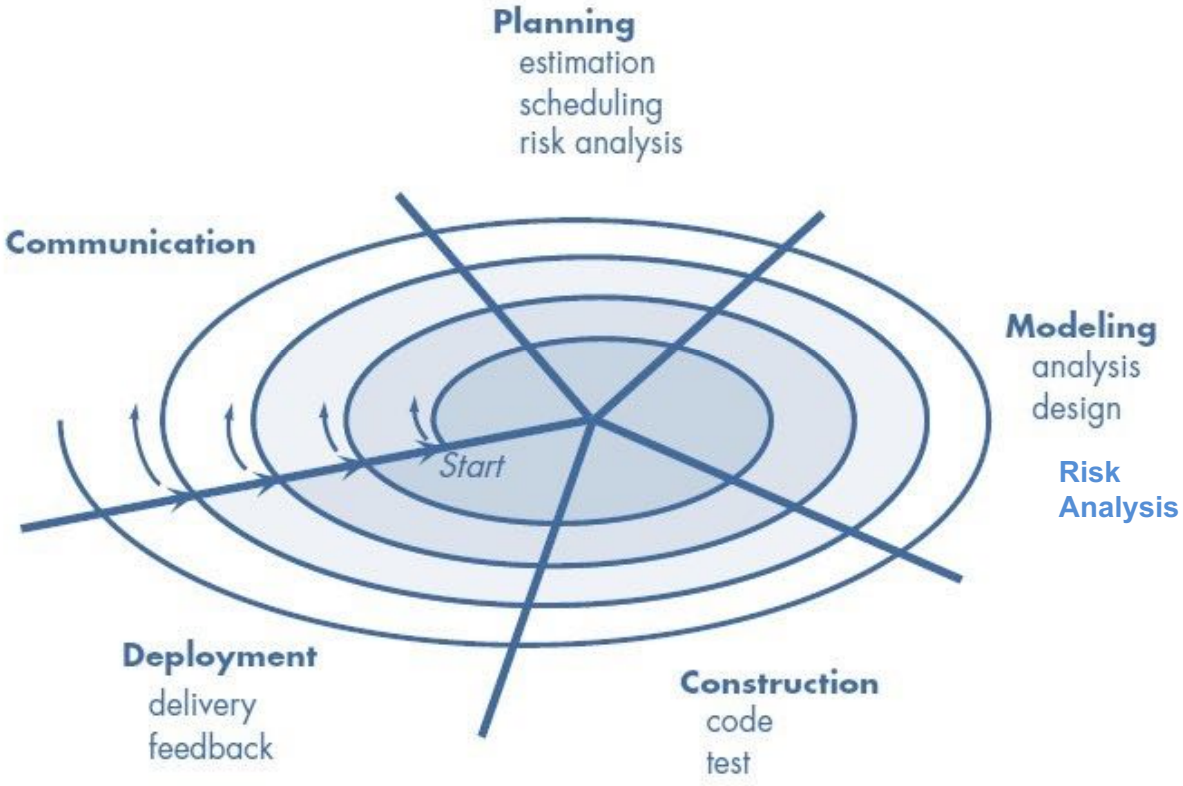
- Waterfall Model (Linear Sequential Model)
- Incremental Process Model
- Prototyping Model
- The Spiral Model
- Rapid Application Development Model
- Agile Model

The Spiral Model

- Spiral model is one of the most important Software Development Life Cycle model, which provides support for Risk Handling.
- In its diagrammatic representation, it looks like a spiral with many loops.
- Each loop of the spiral is called a Phase of the software development process.
- This model is much more flexible compared to other SDLC models.

*In this the features of the product analyzed and the risks at that point of time are identified and are resolved through prototyping.

The Spiral Model



The Spiral Model cont.

- The Spiral model is an evolutionary process model that couples iterative nature of prototyping with the controlled and systematic aspects of waterfall model
- It provides the potential for rapid development.
- Software is developed in a series of evolutionary releases.
- Early iteration release might be prototype but later iterations provides more complete version of software.
- It is divided into framework activities (C,P,M,C,D). Each activity represent one segment of the spiral
- Each pass through the planning region results in adjustments to
  - the project plan
  - Cost & schedule based on feedback

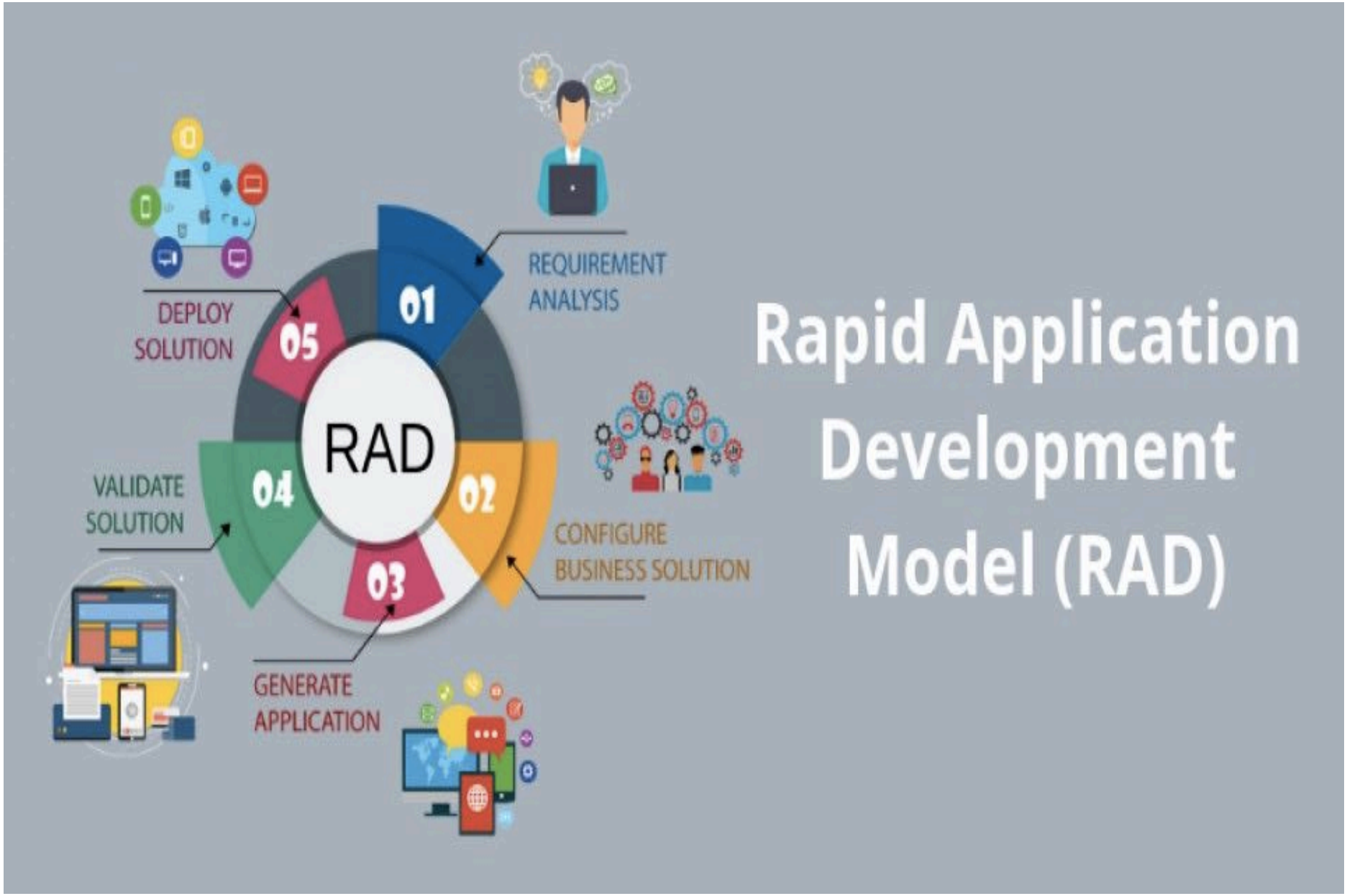
The Spiral Model cont.

- Advantages
  - High amount of risk analysis hence avoidanc of Risk is enhanced. , e
  - Strong approval and documentation control. Additional
  - functionality can be added at a later date. Software is
  - produced early in the Software Life Cycle.
- Disadvantages
  - Can be a costly model to use.
  - Risk analysis requires highly specific expertise.
  - Project's success is highly dependent on the risk analysis phase
  - Doesn't work well for smaller projects.

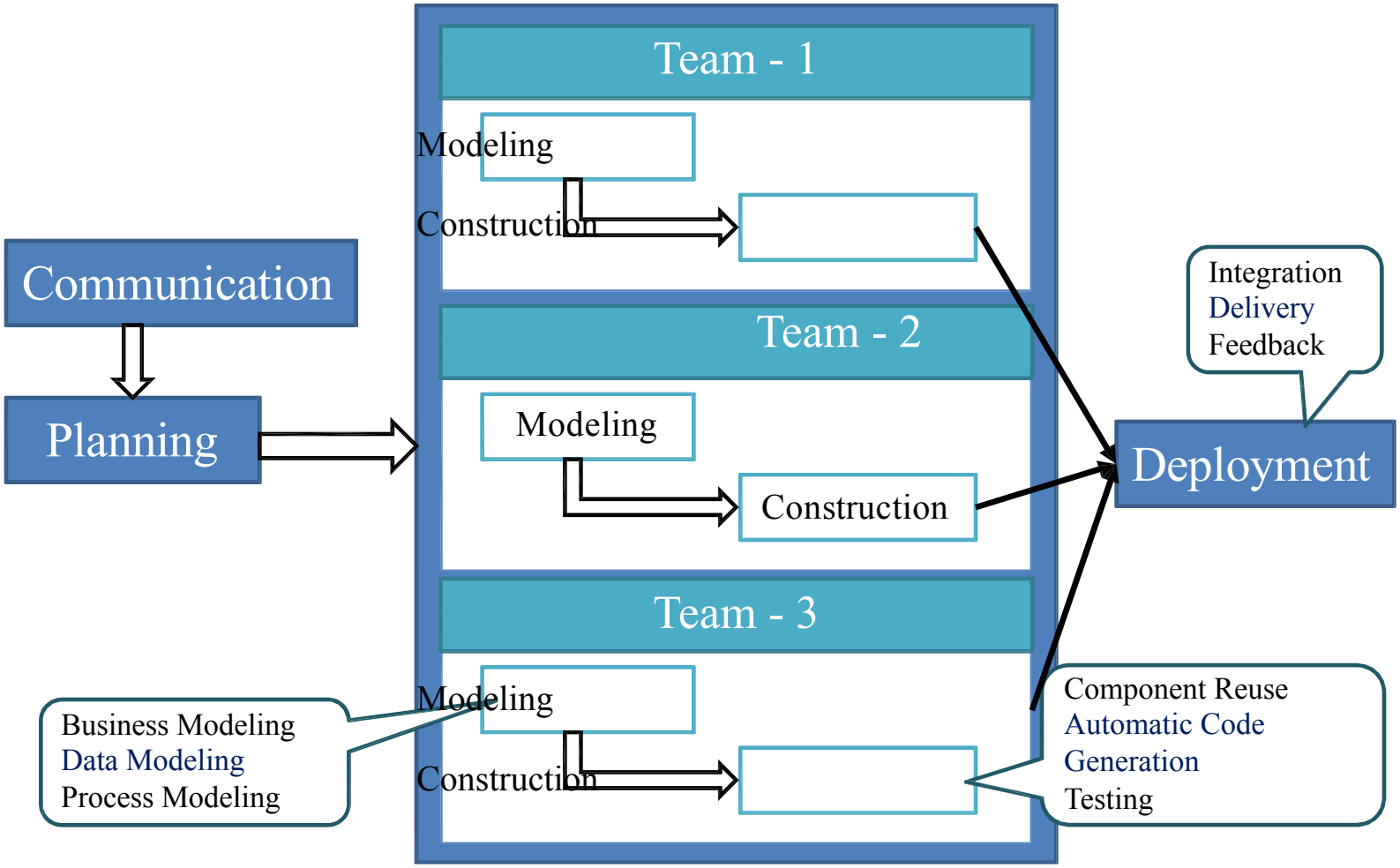
The Spiral Model cont.

When to use Spiral Model

- For development of large scale / high-risk projects.
- When costs and risk evaluation is important.
- Users are unsure of their needs.
- Requirements are complex.
- New product line.
- Significant (considerable) changes are expected.


Rapid Application Development Model (RAD)
RAD
01
REQUIREMENT ANALYSIS
02
CONFIGURE BUSINESS SOLUTION
03
GENERATE APPLICATION
04
VALIDATE SOLUTION
05
DEPLOY SOLUTION

Rapid Application Development Model



RAD Model Cont.

- Modeling
  - Business Modeling: Information flow among the business.
    - Ex. What kind of information drives (moves)?
    - Who is going to generate information? From where
    - information comes and goes?
  - Data Modeling: Information refine into set of data objects that are needed to support business.
  - Process Modeling: Data object transforms to information flow necessary to implement business.
- Construction
  - It highlighting the use of pre-existing software component.
- Deployment
  - Deliver to customer basis on subsequent iteration.

RAD Model Cont.

- When to Use ?
  - There is a need to create a system that can be modularized in 2-3 months of time.
  - High availability of designers and budget for modeling along with the cost of automated code generating tools.
  - Resources with high business knowledge are available.
- Advantages
  - Reduced development time.
  - Increases reusability of components.
  - Quick initial reviews occur.
  - Encourages customer feedback.
  - Integration from very beginning solves a lot of integration issues.

RAD Model Cont.

- Drawback
  - For large but scalable projects, RAD requires sufficient human resources.
  - Projects fail if developers and customers are not committed in a much shortened time-frame.
  - Problematic if system can not be modularized.
  - Not appropriate when technical risks are high (heavy use of new technology).

Based OnCharacteristicsOf Requirements

Requirements	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
Are requirements easily understandable and defined?	Yes	No	No	No	No	Yes
Do we change requirements quite often?	No	Yes	No	No	Yes	No
Can we define requirements early in the cycle?	Yes	No	Yes	Yes	No	Yes
Requirements are indicating a complex system to be built	No	Yes	Yes	Yes	Yes	No

Based On Status Of Development Team

Development team	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
Less experience on similar projects?	No	Yes	No	No	Yes	No
Less domain knowledge (new to the technology)	Yes	No	Yes	Yes	Yes	No
Less experience on tools to be used	Yes	No	No	No	Yes	No
Availability of training if required	No	No	Yes	Yes	No	Yes

Based on User's Participation

Involvement of Users	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
User involvement in all phases	No	Yes	No	No	No	Yes
Limited user participation	Yes	No	Yes	Yes	Yes	No
User have no previous experience of participation in similar projects	No	Yes	Yes	Yes	Yes	No
Users are experts of problem domain	No	Yes	Yes	Yes	No	Yes

Based On Type Of Project With Associated Risk

Project type and risk	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
Project is the enhancement of the existing system	No	No	Yes	Yes	No	Yes
Funding is stable for the project	Yes	Yes	No	No	No	Yes
High reliability requirements	No	No	Yes	Yes	Yes	No
Tight project schedule	No	Yes	Yes	Yes	Yes	Yes
Use of reusable components	No	Yes	No	No	Yes	Yes
Are resources (time, money, people etc)	No	Yes	No	No	Yes	No