AGILE-PARADIGM SHIFT IN SDLC

Session - 5

Specialized Process Models

Objectives

- Explain RAD model
- Explain the phases in the RAD model
- Explain component-based development
- List the advantages and disadvantages of component-based development
- Explain Aspect-Oriented Software Development (AOSD)
- List the advantages and disadvantages of AOSD
- Explain the selection of process model
- Explain Web engineering

Introduction

- Software engineering provides parallel process model that allows building different components in an incremental model.

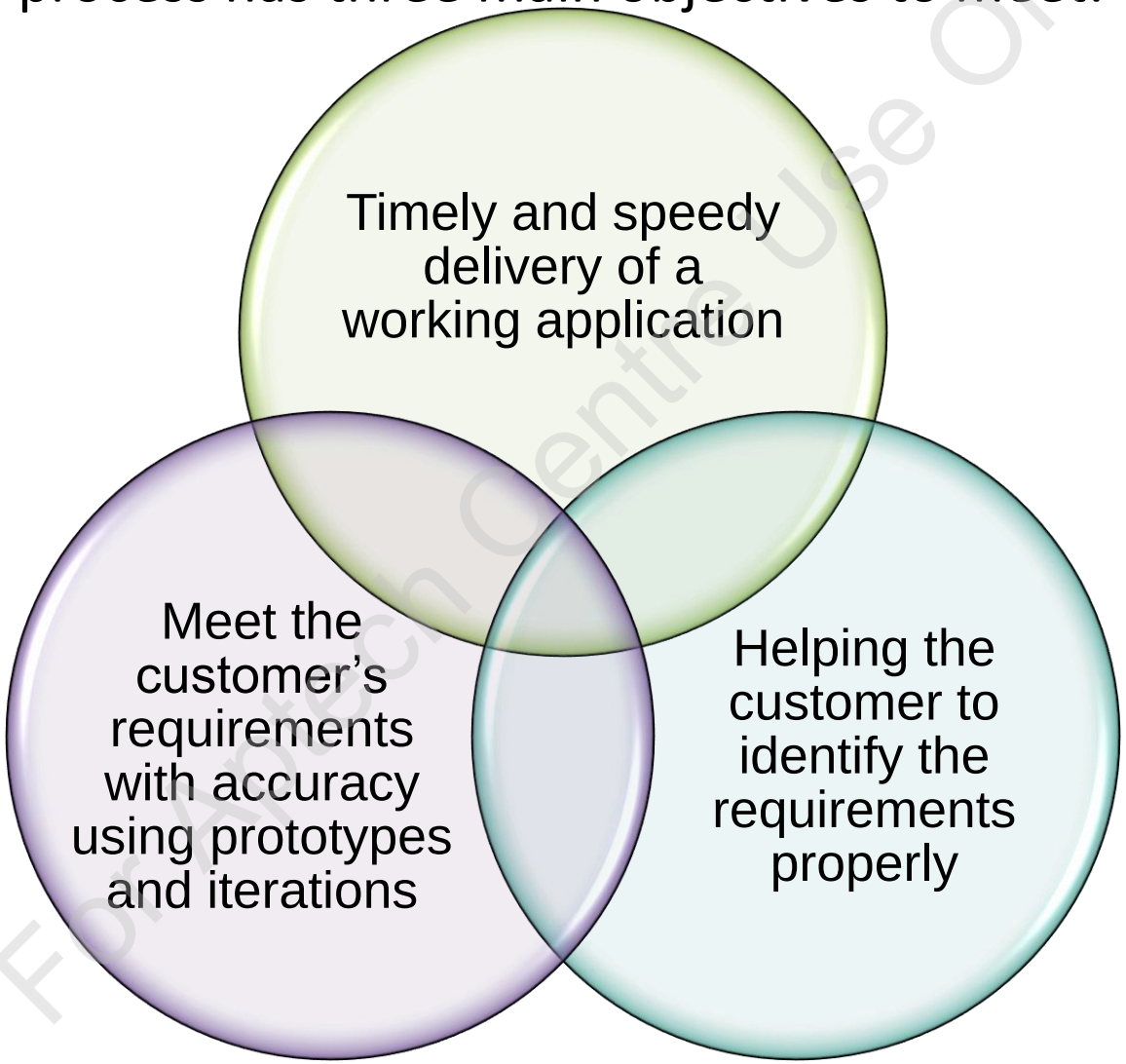
One such type of process model is **Rapid Application Development (RAD)**.

Rapid Application Development [1-2]

- Is a software development methodology that uses very little planning.
- Uses the concepts of parallelism where several prototypes representing functional modules are developed in parallel and are later combined to get the final product.
- Employs minimal planning and hence is able to accommodate changes.
- Uses iterative and incremental development strategies.
- Involves user participation thereby increasing chances of early user community acceptance and an overall reduction in project risk.

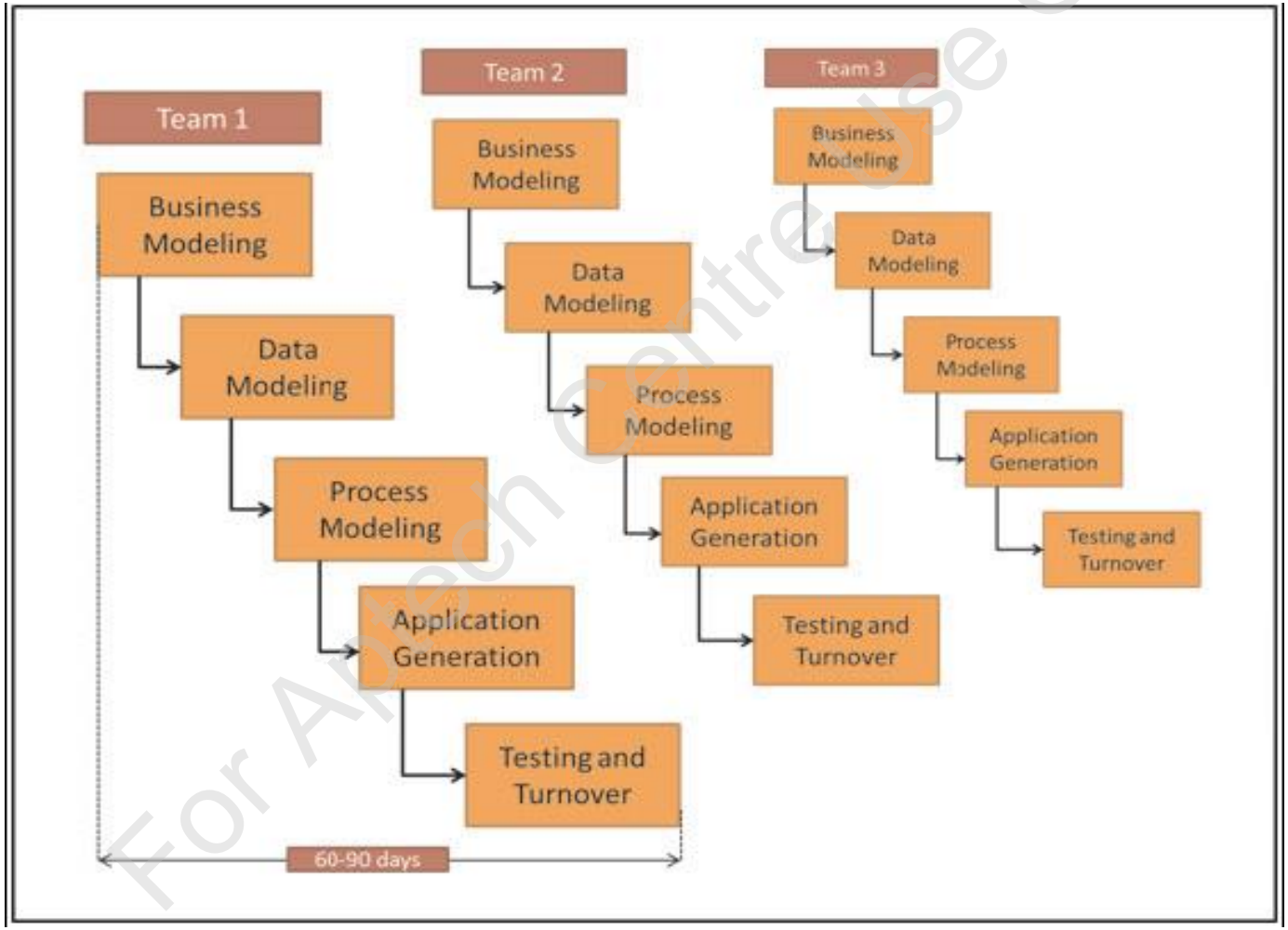
Rapid Application Development [2-2]

- The RAD process has three main objectives to meet:



RAD Phases [1-4]

- RAD lifecycle includes analysis, design, construct, and test in the series of cycles that promote short and iterative development.



RAD Phases [2-4]

- The phases in RAD Model are:

Business Modeling

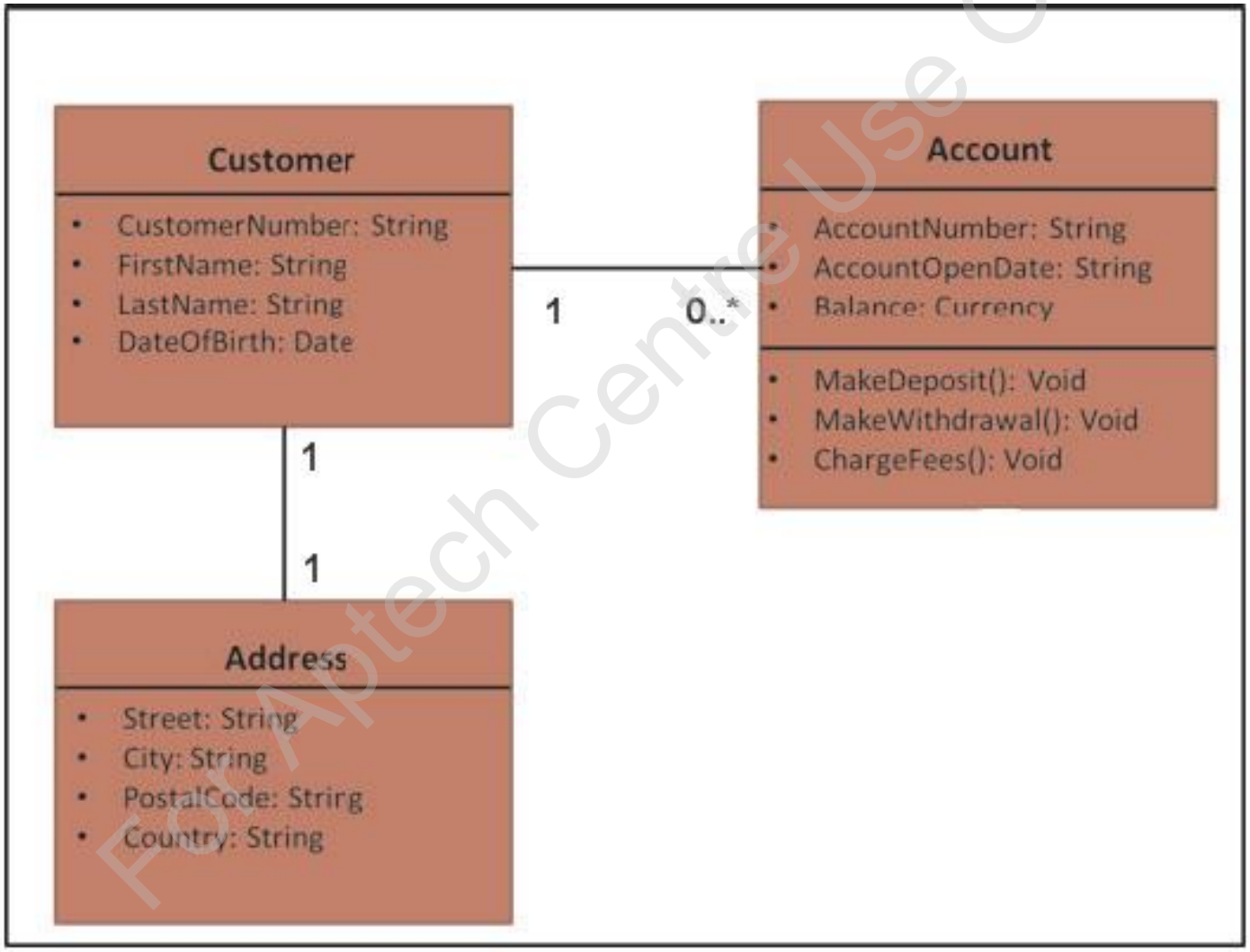
- Business model of the developed product is designed.
- Design model identifies flow of information between various functions of the business.

Data Modeling

- Information gathered during the business. analysis is reviewed in terms of the data objects involved in the business.

RAD Phases [3-4]

- Figure shows the data modeling for banking.



RAD Phases [4-4]

Process Modeling	• Identified data objects are implemented to achieve identified information flow in the business.
Application Generation	• Actual prototypes of actual systems are built by providing implementation for identified process and data models.
Testing and Turnover	• Each prototype is independent so the total time taken to test overall system is reduced. • Also, reduces the major problems that may be identified when the final project is tested.

Applications of RAD Model

- When there is a need to develop system as a set of working modules that are delivered in increments.
- Resources such as modeling designers, high budget, and automated code generating tools are available.
- Skilled manpower with strong business knowledge is available.

Advantages and Disadvantages of RAD

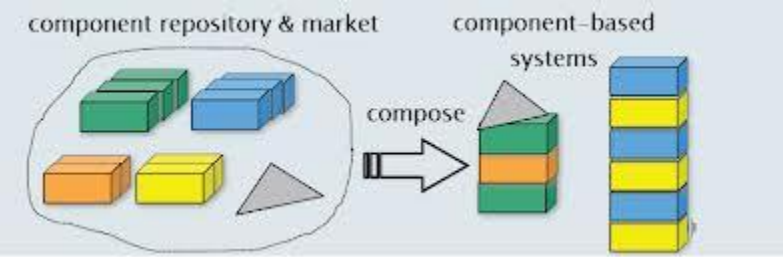
- Table lists advantages and disadvantages of RAD model.

Advantage	Disadvantage
Time taken to build the project is reduced as the project is delivered in small increments.	Higher dependency on technical skills to identify business requirements.
The project manager can estimate costs accurately.	Not suitable for projects with low cost, as generation of automated code is high.
Components ready for the repository are already tested and need not be tested again.	Highly skilled manpower is required for such projects.
Requirements are gathered dynamically and a suitable prototype is developed. Any additional requirement is addressed in the next version of the prototype.	Rapid development sometimes compromises quality parameters such as standardization, reliability, and consistency leading to issues later on.
There is a strong participation by the customer who gives feedback on a continuous basis.	Promotes component-based development which can be scaled as per the business needs.

Component-based Development [1-6]

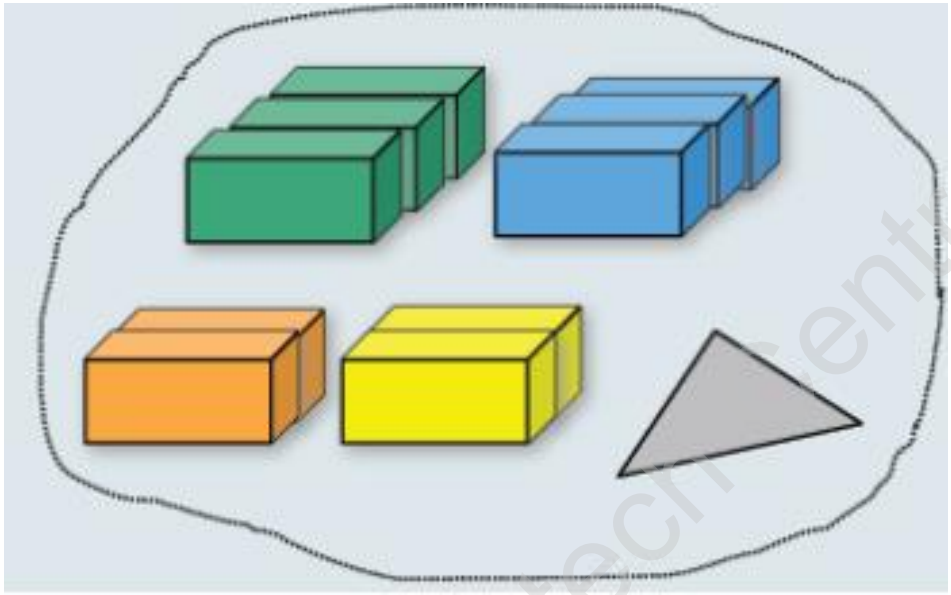
- Component-based development is a variation on general application development, in which:
  - The application is built from discrete executable components, which are developed and deployed relatively independently of one another, potentially by different teams.
  - The application may be upgraded in smaller increments by upgrading only some of the components that comprise the application.
  - Components may be shared between applications, creating opportunities for reuse, but also creating inter-project dependencies.

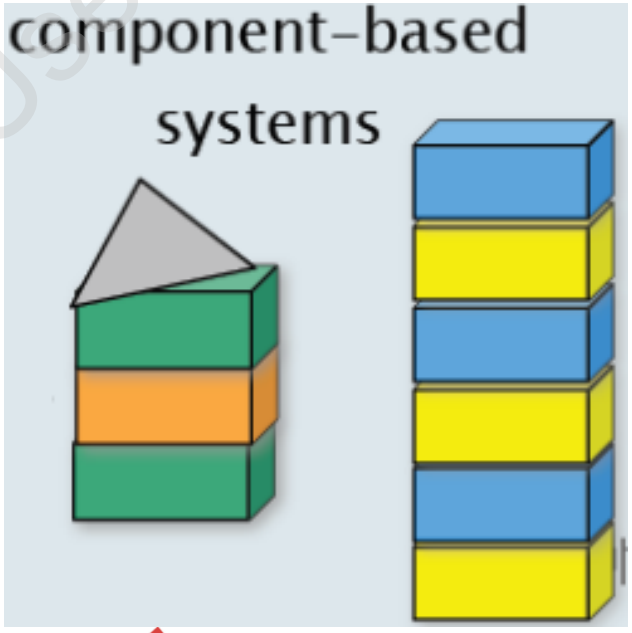
Object technologies provide the technical framework for a component-based process model for software engineering.



Component-based Development [2-6]

- Object technologies provide technical framework for a component-based process model of software engineering.





Component-based Development [3-6]

- Advantages of Component-based development are:

General	
Flexible	• A runtime component can work independently and are less dependent on the software environment.
Reusable	• A developed component can be reused across programming languages and operating systems.
Easy to maintain	• Components are easy to test and maintain as their functionality is implemented only once.
Cost and time-efficient	• Lower development cost and lesser time to develop.

Component-based Development [4-6]

Improving Business Processes

Simplification

- Involves removing unnecessary and redundant steps and reducing variety in processes.

Integration

- Requires combining unconnected processes into a larger coordinated process.

Transformation

- Components are disassembled and put together in a different way to transform the business process.

Component-based Development [5-6]

Improving Software

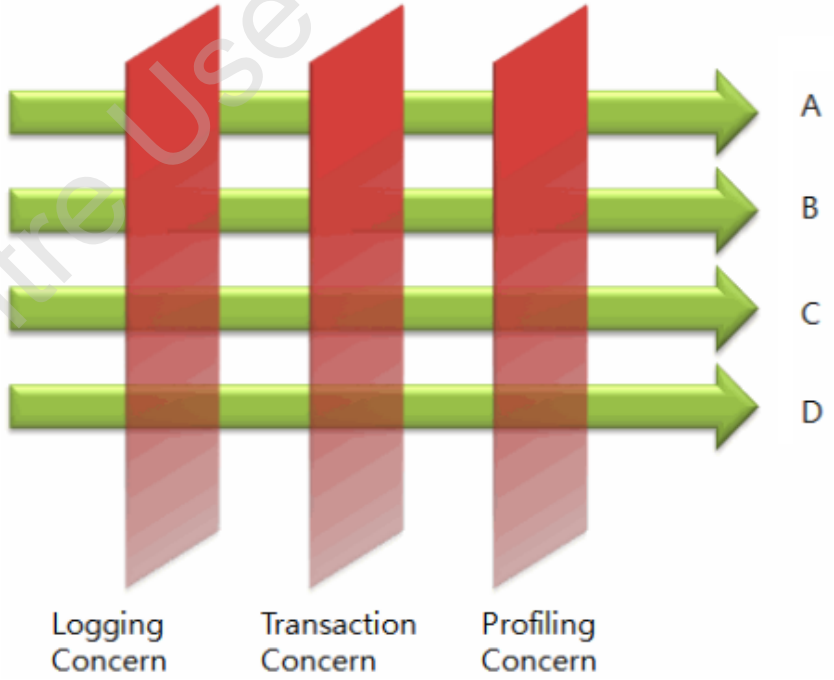
- Greater functionality can be delivered faster by using already existing components.
- Replacing individual components can be replaced easily making maintenance easier.
- A single graphical user interface can be used for standard components.
- Performance can be increased on a component-to-component basis, by optimizing the performance within each component. This improves the overall efficiency.
- If an application can be broken down into components such that each component can completely implement the requirement for which it is designated, then the overall software's reliability is considered to be excellent.
- Components are portable and can be quickly adjusted to work in different platforms without affecting other components.

Component-based Development [6-6]

- Disadvantages of component-based development are:
  - Building the environment that fits the components is challenging.
  - There are not sufficient standards governing middleware in which components work.
  - Comparison between the available standards is not possible, as only CORBA is language-independent.
  - It may not always be possible to find the best fit components for the requirements specification.

Aspect-Oriented Software Development

- It is a new method of developing software applications.
- It involves breaking down a software system into components known as modules.
- It focuses on identification, specification, and representation of cross-cutting concerns and their modularization into separate functional units.



AOSD Concepts and Terminology

Concern

- Any demand, requirement, or expectation on a software system by any stakeholder of that system.

Cross-cutting Concern

- A concern which deals with different areas of the software system.

Aspect

- A module that contains a concern.

Join Point

- A well-defined point in the program's execution where an aspect is invoked.

Pointcut

- A specifically defined subset of join points.

Advice Body

- Code that is executed when a join point is reached.

Weavers

- Specific type of compilers that compose the aspect's implementation with other modules.

Advantages and Disadvantages of AOSD

- Table lists the advantages and disadvantages of AOSD.

Advantage	Disadvantage
Increased modularity.	Reduced efficiency due to increased overhead aspects.
Increased maintainability.	Security risks involving aspects.
Increased reusability.	Lack of a formalized process.
Reduction in the size of the code.	Lack of UML support. Lack of systematic methods of testing.

Choosing a Process Model

- New projects learn from the failures of old projects.

- Previous experience with similar projects is used to organize tasks in differing sequences.

- Organizations define a standard process that describes a sequence of activities or a task, but allows for tailoring to suit a particular project.

Process Tailoring

- Tailoring is defined as the process of adjusting the standard process of an organization to obtain a process that is suitable for the particular business or technical needs of a project.

- The process may be tailored with respect to its scope, formality, frequency, and granularity.

- Some of the factors that may need to be considered for tailoring the process include the skill level of the team, the peak team size, and the criticality of the application.

Web Engineering

- Web Engineering is an **adaptable** and **incremental** (evolutionary) process.
- It is populated by a set of framework activities that occur for all business-critical **WebApp** projects, regardless of the size or complexity.
- The following framework activities are considered for Web engineering:
  - Formulation
  - Planning
  - Analysis
  - Modeling
  - Page generation and testing
  - Customer evaluation

Summary [1-2]

- RAD is a software development methodology that uses very little planning. The concepts of parallelism is used in this model, several prototypes representing functional modules are developed in parallel for quicker product delivery.
- The phases in RAD are namely, Business modeling, data modeling, process modeling, application generation, and testing and turnover.
- Object technologies provide the technical framework for a component-based process model for software engineering.
- The component assembly model leads to software reuse and reusability provides the software engineers with a number of measurable benefits.

Summary [2-2]

- AOSD is a new method of developing software applications which involves breaking down a software systems into components known as modules, so as to separate secondary functions from the main program's business logic.
- The focus of AOSD is on identification, specification, and representation of cross-cutting concerns and their modularization into separate functional units and combining them back automatically into a working system.
- Web Engineering is an adaptable and incremental (evolutionary) process. It is populated by a set of framework activities that occur for all business-critical WebApp projects, regardless of the size or complexity.