UNIT-1
THE SOFTWARE PRODUCT AND SOFTWARE PROCESS

1. Software Process -
   It is defined as the structured set of activities that are required to develop the software system. The fundamental activities are:
   
   (1) Specification
   (2) Design and Implementation
   (3) Validation
   (4) Evaluation Evolution

   A software process specifies a method of developing software. A software project, on the other hand, is a development project in which a software process is used and where the outcome is a software product.

2. Software Characteristics -
   (1) Software is engineered, not manufactured.
   (2) Software does not wear out.
   (3) Most software is custom built rather than being assembled from components.

3. Common process framework -

   COMMON PROCESS FRAMEWORK
   
   FRAMEWORK ACTIVITIES
   
   TASK SETS
   Tasks
   Milestones, deliverables
   SQA points

   UMBRELLA ACTIVITIES

   THE SOFTWARE PROCESS

   SQA = Software Quality Assurance
4. **Software Process Models**

   It is a simplified representation of software process. Each process model represents the a process from a particular perspective, and thus provides only partial information about that process.

   **Types of Software Process Models**
   1. **Iterative Model**
   2. **Prototyping Model**
   3. **Rapid Application Development (RAD) Model**
   4. **Evolutionary Process Models**
      - **Incremental Model** characterized in a manner that
      - **Spiral Model** enables software engineers to develop
   5. **Component Assembly Model** increasingly more complete version of the software

5. **Software Process Characteristics**

   - Predictability
   - Testability and Maintainability
   - Early defect removal and defect prevention
   - Process improvement

6. **Iterative Model (Waterfall Model)**

   ![Waterfall Model Diagram]

   **Advantages**
   - Easy to understand
   - For implementation of small systems
   - Scientific deliverables and milestones
   - Document driven
   - Work well on mature products and weak teams
   - Widely used and known
Disadvantages:
- Ideological, doesn't match reality well
- Unrealistic to complete accurate requirements so early in project
- Software is delivered late in project, delay discovery of serious errors

7. Prototyping Model:

   LISTEN TO CUSTOMER
   →
   CUSTOMER TEST DRIVES MOCK UP
   →
   BUILD/REVISE MOCK UP

Advantages:
- Better communication between customers and developers
- Satisfied user requirements
- High Reliability

Disadvantages:
- Time consuming
- Increased complexity of the system

8. Waterfall Model:

   REQUIREMENTS → DESIGN → IMPLEMENTATION
   ↓ → VERIFICATION → MAINTENANCE

Advantages and Disadvantages is similar to linear sequential model.

9. Rapid Application Development (RAD) model:

   TEAM #1 BUSINESS MODELING → DATA MODELING → PROCESS MODELING → APPLICATION GENERATION → TESTING AND Turnover
   TEAM #2 BUSINESS MODELING → DATA MODELING → PROCESS MODELING → APPLICATION GENERATION → TESTING AND Turnover
Advantages:
- Real-time
- Reliable
- Low Maintenance
- Fast Processing

Disadvantages:
- For large projects, requires a large number of RAD teams
- Not all applications are compatible for RAD
- RAD is not appropriate when technical risks are high

Incremental Model:

Increment 1: **ANALYSIS** → **DESIGN** → **CODE** → **TEST** Delivery of 1st increment
Increment 2: **ANALYSIS** → **DESIGN** → **CODE** → **TEST** Delivery of 2nd increment

Increment n: **ANALYSIS** → **DESIGN** → **CODE** → **TEST** Delivery of nth increment

Advantages:
- Generates working software quickly
- Easier to test and debug
- Lower initial delivery cost
- Easier to manage risk

Disadvantages:
- Need good planning and design
- Total cost is higher than waterfall

Special Model:
Advantages:
- High amount of risk analysis
- Good for large and mission-critical projects
- Software is produced early in the software life cycle
Disadvantages -
- Risk analysis requires highly specific expertise.
- Doesn't work well for small projects.

Component Assembly Model
- Planning
  - Risk Analysis
  - Identify Candidate Components
  - Construct
    - Design & derivation of system
  - Put new components in library
    - Reuse components if available
    - Build components if unavailable
  - Engineering
    - Construction and release

Concept of Object-oriented programming is used.

Rational Unified Process (RUP) - (Framework for Object oriented models)
- There are five phases for Rational unified process model:
  1. Inception
  2. Elaboration
  3. Construction
  4. Transition
  5. Production
Agile Unified Process (AUP) - (simplified version of RUP)
There are seven phases of AUP -
(1) Model
(2) Implementation
(3) Test
(4) Deployment
(5) Configuration Management
(6) Project Management
(7) Environment

Software process customization and improvement -
Improving the quality and reducing the cost of products are fundamental goals of software process.
Quality Improvement Paradigm (QIP) consists of six basic steps -
(1) Characterize
(2) Set Goals
(3) Choose Process
(4) Execute
(5) Analyze
(6) Package

Compatbility Maturity Model (CMM) -
It is used in assessing how well an organization's process allows to complete and manage new software projects.
Various process maturity levels are -
(1) Initial
(2) Repeatable
(3) Defined
(4) Managed
(5) Optimizing
Software metrics -

There are three types of software metrics -

1. **Product metrics** (describe characteristics of the product e.g. design)
2. **Process metrics** (used to improve software development and maintenance)
3. **Project metrics** (describe project characteristics and execution)