**Chapter 1 and 2 (First 10 pages) Notes**

**Chapter 1**

**Software**: Programs and all associated documents

**Good software**: delivers required functionality and performance to the user and is maintainable, dependable, and usable

**SW Engineering**: Engineering discipline that is concerned with all aspects of software production

**Fundamental SW Engineering Activities**: Specification, Development, Validation, Evolution

**Difference between SWE and CS**: CS focuses on theory and fundamentals, while SWE is concerned with the practicalities of developing and delivering useful software

**Difference between SWE and System Engineering**: SE is concerned with all aspects of computer-based systems including hardware, software, and process engineering. SWE is part of SE.

**Key Challenges for SWE**: Coping with increasing diversity, demands for reduced delivery times, and developing trustworthy software.

**Costs of SWE**: 60% Development, 40% Testing

**SW Process**: The systematic approach that is used in SWE. It is a sequence of activities that leads to the production of a software product.

1. Specification: Defining the software that is to be made and the constraints on its operation
2. Development: SW is designed and programmed
3. Validation: SW is tested and checked to ensure that it is what the customer requires
4. Evolution: Software is modified to reflect changing customer and market needs

**2 types of software**:

* Generic Products
* Standalone-systems and sold on the open market to any customer
* Customized (bespoke) products
* Commissioned by a customer. Developed especially for this customer

**3 General Issues of SWE**:

1. Heterogeneity: Systems are required to operate as distributed systems across network that include different types of computers and devices (mobile, etc.).
2. Business and Social Change
3. Security and Trust

**Different types of applications (this decides which SWE method and technique is used)**:

1. Stand-alone applications
2. Interactive transaction-based applications
3. Embedded Control Systems
4. Batch Processing Systems
5. Entertainment Systems
6. Systems for Modeling and Simulation
7. Data Collection Systems
8. Systems of systems

**SWE Principles that apply to all types of SW Systems**:

1. Developing using a managed and understood developments processing
2. Dependability and Performance
3. Understanding and managing the software specification and requirements
4. Make as effective use as possible of existing resources

**Ways Web-based systems are engineered**:

1. Reusing existing software components and systems to facilitate creating the Web system
2. It is impractical to specify all the requirements in advance so develop and deliver Web systems incrementally.
3. GUIs on Web systems are often poorer than PC components

**Expected Ethics in SWE**:

1. Confidentiality:
2. Competence:
   1. Do not misrepresent your level of competence
3. Intellectual Property Rights:
   1. Protect the intellectual rights of your employers
4. Computer misuse:

**Key Points**:

1. SWE is an engineering discipline concerned with all aspects of SW production
2. SW includes programs and documentation
3. SW Process includes all of the activities involved in SW development (Specification, Development, Validation, and Evolution are all part of all SW processes)
4. SWE Fundamentals that apply to all types of system development are SW Processes, Dependability, Security, Requirements, and Reuse.

**Chapter 2**

**SW Process**: Set of related activities that leads to the production of a software product.

1. Specification
2. Development
3. Validation
4. Evolution

**2 Different Categories for SW Processes**:

1. Plan-driven
   1. All of the process activities are planned in advance and progress is measured against this plan
2. Agile Processes
   1. Planning is incremental and it is easier to change the process to reflect changing customer requirements.

**Process Models**

1. Waterfall Model
   1. Takes the fundamental process activites of specification, development, validation, and evolution and represents them as process pahses such as:
      1. Requirements Definition
      2. System and Software Design (design the entire architecture)
      3. Implementation and Unit Testing
      4. Integration and System Testing (all the programs are integrated and tested)
      5. Operation and Maintainance (put into use in market and any bugs are corrected)
   2. It is Plan-driven
   3. Each phase depends on the previous phase
   4. Weakness: its inflexible to partition the project into distinct phases, commitments must be made at an early stage in the process.
   5. This should only be used when requirements are well understood and unlikely to change radically during development.
2. Incremental Developments
   1. Interleaves the activities of specification, development, and validation. The systems are developed as a series of version (increments), with each version adding functionality to the previous version
   2. Is Agile-Driven
   3. Developing an initial implementation, exposing this to user comment and evolving it through several versions until an adequate system has been develop.
   4. Specification, Development, and Validation activities are interleaved rather than separate.
   5. Advantages: Cheaper and easier to make changes, don't have to define all requirements in the beginning, rapid delivery, and easier to get customer feedback
   6. Disadvantages: It is not feasible to create documents for each release and System structure tends to degrade as new incremented are added
3. Reuse-oriented Software Engineering
   1. Based on the existence of a significant number of reusable components. The SW development process focuses on integrating these components into a system rather than developing them from scratch.
   2. Phases
      1. Requirements Specification
      2. Component Analysis
         1. Search for component to implement the specification
      3. Requirements Modification
         1. Requirements modified to suit the component analyzed
      4. System Design with Reuse
         1. System is designed or an existing framework is reused
      5. Development and Integration
      6. System Validation
4. Examples of Reusable Components
   1. Web services
   2. Packages in .NET or j2EE
   3. Stand-alone systems
5. Advantages
   1. Reduces the amount of software to be developed and reducing costs and risks
6. Disadvantages
   1. Control is lost over components that are reused

**Process Activities**

1. SW Specification
   1. The process of understanding and defining what services are required from the system and identifying the constraint on the systems operation and development
   2. Presented at 2 levels
      1. End-users and Customer
      2. Developers
   3. 4 Main Activities
      1. Feasibility study
         1. Decides whether it is feasible to go ahead with a more detailed analysis
      2. Requirements elicitation and analysis
         1. Other existing systems are inspected and discussions with potential users occur here
      3. Requirements specification
         1. Translating the information gathered in previous stage into a document (1 is User Requirements for the customer and 1 is System Requirements for the developers)
      4. Requirements Validation
         1. Requirements are checked for realism, consistency, and completeness.