CH.2

- **IEEE**
  - **Program**
  - **Procedures**
    - Order & Schedule
  - **Documentations**
  - **Data**
- **Software faults vs Software failures**
  - **Fault**
    - Mistake
    - Happens in some areas only
    - Doesn’t affect the whole application
    - Some faults could turn into a failure
    - Developers main concern
  - **Failure**
    - State
    - The whole application
    - Users main concern
- **Common software errors**
  - **Faulty definition of requirements**
  - **Client – Developer communication failures**
  - **Deliberate (متعمد) deviations from software requirements**
  - **Logical error**
  - **Coding error**
  - **Non-Compliance with documentation and Coding instructions**
  - **Shortcomings of testing process**
  - **Procedure error**
  - **Documentation error**
- **Crosby**

  - Quality
    - Software Specified Formal requirements
    - Client is responsible for requirements specification errors
- **Joseph Juran**
  - Quality
    - User real needs
    - User satisfaction
    - Spend more time to correct user requirement
    - User has no blame in any professional error in requirements
- **Pressman**
  - Quality
    - Follow stated functional requirements
    - Follow stated contract standards
    - Follow unmentioned / unwritten software standards
- **Software quality assurance**
  - **Objectives**
    - Assuring acceptable level of confidence and conformance to functional technical requirements
    - Assuring acceptable level of confidence and conformance to managerial requirements (Scheduling and budgets)
    - Initiating and Managing activities for the improvement and the efficiency of the software development
- **Software quality assurance vs Software quality control**
  - **Assurance**
    - Activities
    - In development phase
  - **Control**
    - Final product

        - Meet quality or not
        - Go or hold
- **Software quality assurance vs Software engineering**
    - **Assurance**
        - Software engineering is a must in order for it to happen
    - **Engineering**
        - Systematic, Disciplined approach for development

CH.3

- **Quality factors**
    - Team responsibility
    - Quality attributes
- **Software quality models**
    - **McGall 11**
    - **Deutsch and Willis**
    - **Evans and Marciniak**
- **McGall**
    - **Product operation** Factors affecting Daily operation
        - **Correctness :** correctness & completeness & up-to-dates percent of the output
        - **Reliability :** Max failure rate
        - **Efficiency :** Hardware resources and computer processing
        - **Integrity :** Security, Access control and Access audit (Logs)
        - **Usability :** Training

  - **Product revision** Deals with the product maintenance
    - **Maintainability :** Ability to correct failures and maintenance
    - **Flexibility :** Adaptive to any circumstances and easy to use
    - **Testability :** To be able to test itself and then tell the user that there is a problem also the easement of testing to the tester
  - **Product transition** Deals with the product adaptation and interactions with the environments
    - **Portability :** Adaptation to work in any environment (OS, Hardware, etc..)
    - **Reusability :** Ability to be reused again in other projects
    - **Integration (Interoperability):** Can be integrated with other software or hardware
- **Evans and Marciniak**
  - **No testability**
  - **Verifiability**
    - Ability to verify design and programming
    - Modularity and Simplicity
  - **Expandability**
    - Future efforts when population increases and when new applications are added
- **Deutsch and Willis**
  - **No testability**
  - **Safety**
  - **Manageability**
    - Software modification from a dashboard
  - **Survivability**
    - Continuity
    - Time from failure to recovery

CH.4

- ## SQA system
  - **It consists of several components**
    - **Pre-project components**
      - Development plan & Quality plan
      - Contract review
    - **Project life cycle components**

      Has two types:

      - **Development phase**
        - Design and programming errors
        - Reviews, Testing and Experts opinions
      - **Operation maintenance phase**
        - Maintenance
        - SQA of external participants
    - **Infrastructure and improvement components**
      - Reduce rate of errors
      - Based on organization experiences
    - **Software quality management components**
      - Control development and maintenance activities
      - Introduce managerial support actions
    - **Standarization components**
      - Implement international standards
    - **Organizing components**
      - Organization of all SQA human components

- ## Pre-Project components
  - **Contract review**
    - Clarify customer requirements
    - Evaluate man power & hardware resources
    - Evaluate customer capacity

        - Evaluate development risks
        - Evaluate schedule and resources
    - **Development plan & Maintenance plan**
        - **Dev plan**
            - Schedules
            - Man power
            - Methodology
            - Software reuse plans
            - Risks
            - Organization issues
        - **Quality Plan**
            - Quality goals
            - Criteria of start and end
            - Tests, reviews

- **Life cycle**
    - **Review**
        - **Formal**
            - Two
            - High ranks
            - Immediate approval
        - **Peer**
            - Peers
    - **Expert opinion**
        - Insufficient expertise, capabilities
        - Seniors Disagreement
    - **Software testing**
        - Test cases & Scenarios
    - **Software maintenance**
        - Corrective
        - Adaptive
        - Improvement
    - **External participants**
        - In large complex projects
        - Subcontractors ,suppliers and customers

- **Infrastructure**
  - **Lowering software failure rate**
  - **Improvement & Productivity**
  - **Classes**
    - **Procedures and Work instructions**
      - **Procedures**
        - Work for the entire organization for any project to insure quality
      - **Work instructions**
        - Instructions to follow for a certain use or function
    - **Supporting quality devices**
      - **Use templates and checklist**
        - to insure completeness and quality
        - Save time
        - Standarization
    - **Staff training**
    - **Preventive & Corrective actions**
      - Based on organization experience
    - **Config management**
      - Responsible for new versions and their releases
      - Last the entire software service
    - **Document control**

- **Management components**
  - **Managerial control of development & maintenance**
  - **Classes**
    - Project progress control
      - Insure there is no deviation from project plan
      - Maintenance
      - Activities
    - Software quality metrics
    - Software quality costs

- **Standards**
    - **Quality management standards**
        - What is required
    - **Project process standards**
        - Provide methodology