## Chapter 1.

# SOFTWARE PROJECT MANAGEMENT

# 1. OBJECTIVES, DEFINITIONS

## 1.1 SOFTWARE PRODUCTION ACTIVITY

- Innovative character
- Requires a large amount of creativity
- The tendency was to develop SW in an artisan manner. Negative results:
    - Difficult integration of programmers in teams and organizations which develop SW
    - The high complexity of SW development, in absence of a real organizational template, can determine a waste of resources
- SW Production Particularities:
    - (1) Requires Efficiency and Flexibility related to
        - Standard Application Domain
        - Standard Development Platform
        - Standard Development Process
        - Standard Development Environment
    - (2) Induces Unforeseen Impact
        - Change of Requirements by Customer
        - Change of Development Platform by Management
        - Fluctuation of Staff (e.g. Shift of Employees)
        - Change of Delivery Dates by Management
        - Delays caused by internal & external Suppliers
    - (3) Immaterial Nature of a SW Product (Active noncorporale)
        - Features hardly experienced by Customer (Cost Relation)
        - Intermediate Results hard to measure (acc. final Result)
        - Hard to evaluate
        - Difficult to introduce in accountancy
        - Risks: Feature Set, Costs, Duration
    - (4) Level of innovation
        - Makes Use of newest Technologies
        - Realizes new Functions from Scratch
        - Risks: low Quality and Runtime Behavior
    - (5) Missing Process Maturity
        - Missing standardized and established Development Proc.
        - Missing Understanding of engineering type of SW Eng.
        - Risks: low Quality and Runtime Behavior

- Capability Maturity Model Integration (CMMI)

- CMMI is a process maturity model aiming to help organizations improve their performance.
- CMMI describe an improvement path for immature processes to a mature, better organized process.
- Most large software organizations and corporations adopted CMMI.
- CMMI levels refer to a set of steps which each organization can obtain in order to increase its value on the market.
- CMMI contains five maturity levels illustrated in figure 1.

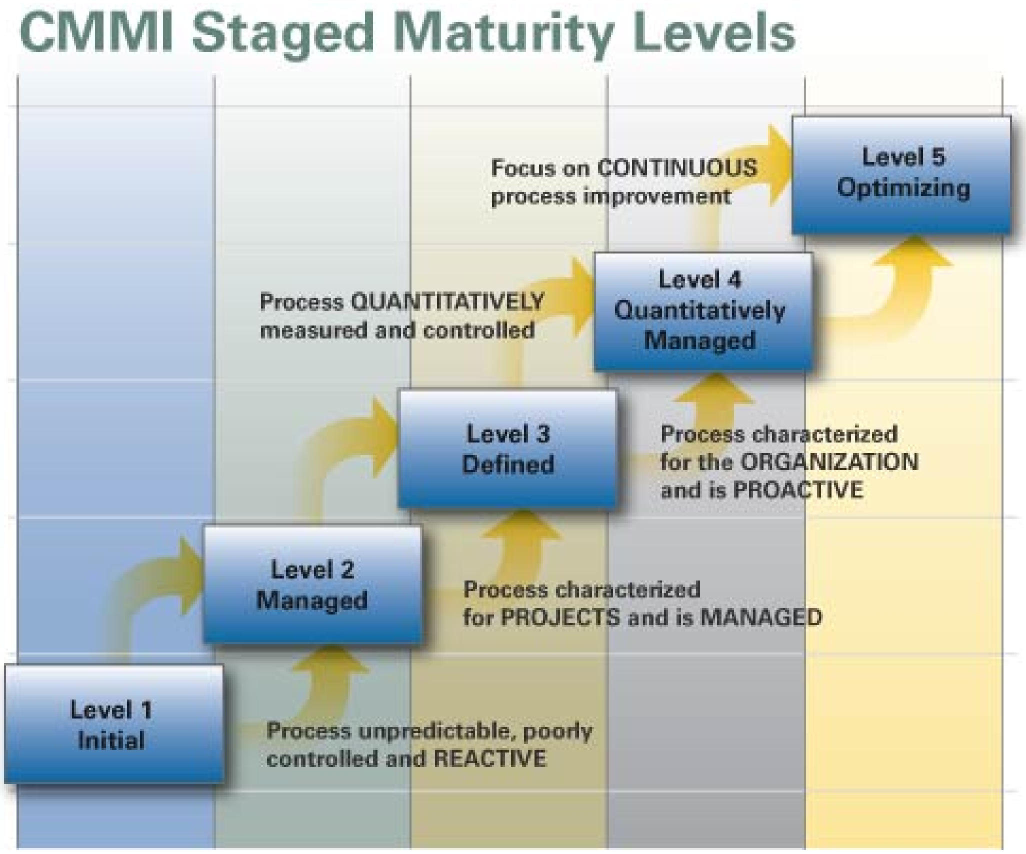
CMMI Staged Maturity Levels
Level 5
Optimizing
Focus on CONTINUOUS process improvement
Level 4
Quantitatively Managed
Process QUANTITATIVELY measured and controlled
Level 3
Defined
Process characterized for the ORGANIZATION and is PROACTIVE
Level 2
Managed
Process characterized for PROJECTS and is MANAGED
Level 1
Initial
Process unpredictable, poorly controlled and REACTIVE

Fig. 1. Characteristics of the maturity levels

- ISO 9000 Standard

  - ISO 9000 is a family of standards for quality management systems.
  - It describes fundamentals of quality management systems and defines related terms.
  - It promotes  top management commitment to quality, customer focus, adequacy of resources, employee competence, process management , quality planning,  product design, review of incoming orders, purchasing, monitoring and measurement of the processes and products.

### 1.2 ESSENTIAL REQUIREMENTS FOR A SUCCESSFULLY SW PROJECT

- (1) Covers exactly the customer needs
- (2) Is finalized in time
- (3) The development costs do not exceed the allocated budget
  - To fulfill these requirements a rigorous organizational process is necessary.

### 1.3 DEFINITIONS

- Classical definition of Production Process:
  - **Definition:** Production Process – *A term which describes the production process of material goods. Usually it presumes a sustained effort to replicate a prototype developed in a previous phase*.
- For the **SW production** the usual term is SW PROJECT
  - **Definition1:** SW Project – An assembly of organized activities, related to the development of programs, parts of programs or systems of programs, with the main purpose of achieving well defined characteristics for the developed products.
  - **Definition2:** SW Project - A Project is a Proposal, which is characterized by the Uniqueness of its Conditions, for instance:
    - Target/Goal,
    - Limits in terms of Time/Financial Resources/Staffing/others,
    - Differentiation with other Proposals,
    - Project specific Organization.

      (German Industrial Norm, #69 901)

## 2. THE NEED FOR SW PROJECT MANAGEMENT

### 2.1 REASONS TO ORGANIZE THE SW PROJECT DEVELOPMENT

- Initial estimation (when is ready, how much does it cost)
- Activities factorization (determined by project complexity which on its turn is determined by the complexity of the functional requirements)
- Project execution (preparation, planning, supervision, control)
- Configuration Management (project integrity)
- Control of the planed schedule (time, budget)
- Control of product quality (*) (Quality Management)

(*) **Definition:** quality is the measure of the manner in which the SW product satisfies the requirements imposed by users, clients, buyers, or customers

## 2.2 PROJECT DECOMPOSITION IN CONSTITUENT ACTIVITIES

- **Project management** it's in fact a *disintegrate process* which presumes in the same time an ***organizational process*** which manages the interdependencies between the resulting parts or activities.
  - The interdependencies can be related to time or to the human and material resources necessary to the activities accomplishing.
- Usually the development of a SW product is achieved through a ***development process*** which consists in several phases.
  - These phases are presented in an intuitive manner in **Deming's quality wheel.**

## 2.3 QUALITY WHEEL (DEMING) (THE PHASES OF THE DEVELOPMENT CIRCLE)

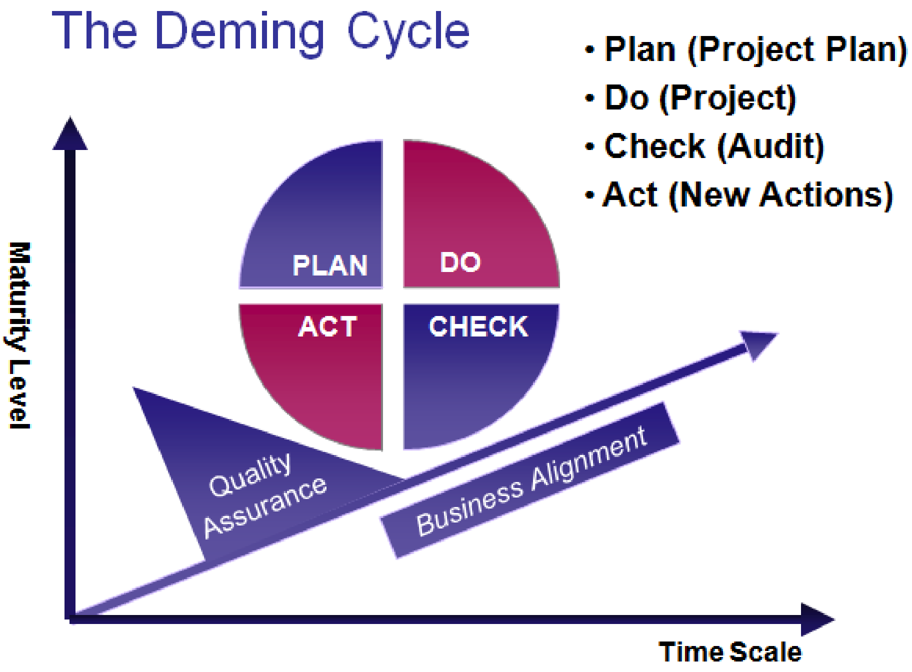

**Fig.2.3.a.** Deming's Wheel

- (1) **Plan** presumes:
  - Product definition
  - Rigorous resources estimation
  - Establishing the interdependencies between activities and/or costs
  - Establishing the ending phases conditions (time estimation and phases exits criteria)
  - Establishing the operations inside actions and the implied actors (costs and human resources estimation)

- (2) **Do** – the effective product development. Usually is the only phase accepted by programmers
- (3) **Check**
  - Quality estimation
  - Measurements of the developed product
  - Costs determination
  - Establishing if the scheduled terms were respected
- (4) **Act**
  - Analyze of information obtained in the previous phase
  - Correct the definition deficiencies or developing process problems

- **Obs**:
  - **Three** of the four phases are in fact related to management
  - The management phases:
    - Grow the development effort (costs, time, resources)
    - Induce favorable effects by increasing the reliability of the development process

## 3. PROCESSES, ACTIVITIES AND TASKS IN A SW PROJECT

### 3.1 ISO/IEC 12207: 1995 STANDARD

- The multitude and the complexity of the problems related to the development of a SW product implied the necessity of a systematical approach and standardization.
- The result was **ISO/IEC 12207:1995 Standard** having as main purpose to establish for **SW industry:**
  - A **common framework**
  - A **well defined terminology**

### 3.2 DEFINITIONS

- **Definitions:** In accordance with this standard a **SW PROJECT** consists in:
  - (1) **Processes** – an assembly of *resources* and *interdependent activities* oriented to a well defined purpose.
  - (2) **Activities** – are *parts of a process* consisting in *types of actions* through which, process resources are used for project purpose.
  - (3) **Tasks** – are *components of activities* consisting in one or an assembly of *actions.*
    - A **task** can be related with a person or a group of persons having the responsibility of their accomplishment
    - For any **task** must be established or estimate
      - A **resources allocation**
      - A **time horizon**
      - A **cost**

### 3.3 TYPES OF PROCESSES

- (1) **PRIMARY PROCESSSES (P)**
- (2) **SUPPORT PROCESSSES (S)**
- (3) **ORGANIZATIONAL PROCESSSES (O)**

### 3.4 PRIMARY PROCESSSES

- **PRIMARY PROCESSES** are the processes deserving the main parts (**actors**) of a SW project: *acquisition, supplier, developer, operator (user) and maintainer of the product*
- **ISO/CEI 12207:1995 STANDARD** defines **5 Primary Processes**:
    - (1) **Acquisition Process** – defines the activities through which an organization **acquires** a system, a product or a SW service
    - (2) **Supplying Process** – defines the activities through which an organization **supplies** a system, a product or a SW service
    - (3) **Development Process** – consists in activities through which an organization **defines and elaborates** a system, a product or a SW service
    - (4) **Utilization Process** – defines the activities through which an organization **utilizes** a system, a product or a SW service
    - (5) **Maintenance Process** – defines the activities through which an organization **supplies maintenance** service for a system, a product or a SW service

### 3.5 SUPPORT PROCESSES

- **SUPPORT PROCESSES** are processes which support other processes. They contribute to the success and the quality of a SW project.
- **ISO/CEI 12207:1995 STANDARD** defines **8 Support Processes**:
    - (1) **Documentation Process** – includes the activities concerning the **definition** and **recording** of all information resulted from the SW developing process.
        - That presumes *user documentation* as well as *documents related to developing process*: plans, reports, specifications, internal standards, associated documents, internal procedures.
    - (2) **SW Configuration Management Process (SCM)** – consists in **administrative** and **technical procedures** which
        - Identify, define and establish the *SW configuration elements* (components, modules, units, files, data structures)
        - Control the storage, the handling and the delivery of the *SW components*
        - Establish *product versions*
        - Establish *state of the components* (functionalities, disfunctionalities, errors)

    - Control the *modifications* on passing from a version to another (Control Versions Management)

  - (3) **Quality Assurance Process (QA)** – defines the assembly of activities which assure in an objective manner that
    - The realized SW product fulfill the *specified requirements*
    - The implied processes comply with a set of established *plans* and *procedures*
  - (4) **Testing Process (*)** – defines the assembly of activities having as purpose the verification of the products resulted from developing activities, which *satisfy imposed requirements and conditions*.
    - The verification has different degrees of depth depending on the activity whose product is tested
  - (5) **Validation Process (*)** – defines the assembly of activities which verifies if a SW product which is in a **final phase**, satisfies the *planned utilization requirements* (covers the user's needs resulted from the analyze process)
  - (6) **Common Analyze Process (*)** – is the process of **analyze/evaluation** of the state of a process or product.
    - It's a *periodical process* which involve the parts implied in project (usually the developer, the beneficiary and the purchaser or supplier)
    - It focuses on either the analyze *of SW product requirements* or the measurement of the "pulse" of the project
  - (7) **Audit Process (*)** – contains the activities oriented to **certify** the conformity with norms, requirements, schedules, and statements of the **contract** for a product or a SW process.
    - In principle, these activities are **similar** with those realized by test, **validation** or **analyze processes**, with the following differences:
    - (1) They are accomplished **during** the development of the activity or task, and **not** at the end, as in the case of test or validation process.
    - (2) The auditing part has **no** direct responsibilities in the implied products and processes, element that differentiates the auditing process from the common analysis one.
  - (8) **Problems Solving Process (*)** – includes activities concerning *analyze and solving of the problems* (non-conformities, functional errors, unexpected situations)

- **Obs.** The processes marked with (*) (**Testing, Validation, Common Analyze, Audit, Problem Solving**) can be utilized as **techniques** for the Quality Assurance Process

### 3.6 ORGANIZATIONAL PROCESSES

- **ORGANIZATIONAL PROCESSES** are processes related to the *management, infrastructure, training, and improving*
- **ISO/CEI 12207:1995 STANDARD** defines **4 Organizational Processes:**

- (1) **Management Process** – defines the basic activities related to the management of any process
- (2) **Infrastructure Process** – consists in all the activities concerning establishing, achieving and maintaining the infrastructure of any process.
    - By infrastructure we mean hardware, software, tools, techniques, standards and facilities for development, exploitation and maintenance
- (3) **Training Process** – specifies the set of activities for training and maintaining the professional level of the personal.
    - The main effort is directed to improve the knowledge and to increase the qualification of the personal.
- (4) **Improving Process** – consists in the set of activities oriented to definition, evaluation, measurement, control and improvement of any process.

# 4. DEVELOPMENT PROCESS

- It belongs to the **Primary Processes.**
- Is the main part of the entire SW project implying the highest support from the other processes.
- Consists in a number of specific activities.
- It is directed and controlled by the Management Process (O).

## 4.1 DEVELOPMENT PROCESS ACTIVITIES

- (1) **Process Initiation** – presuming:
    - (a) Selection and utilization of a **life cycle model** in accordance with the dimension, the complexity and the application domain of the SW product to be developed
    - (b) Elaboration of the **Project Development Plan** based on Documentation Process (S) specifications, consisting in:
        - **Standards, methods and specific tools** used in development. Usually these are outputs of the Infrastructure Process (P)
        - **Factorization of the process actions in tasks**
            - Identification of the knowledge and the aptitudes necessary for tasks achievement
            - Establishing the tasks scheduling
            - Identifying the persons responsible with the carry out of each task, based on estimation of the necessary skills.
        - If the **team have to be trained**, this is performed as part of the Training Process (O)
        - **Identification** of the **Development Process outputs**, their scheduling, and specification, or referring the Configuration Management Process (S) if necessary

        - **Identification** of the **deliverable outputs** of the Development Process (P) and specification of their characteristics

- (2) **SW and System Requirements Analyze**
    - The output of the activity is the document named **Specification (Problem Specification, SPEC).**
    - This document is conform with Documentation Process (S) and includes:
        - System and SW features and capabilities
        - Security, ergonomic and business requirements
        - Organizational requirements
        - Interface requirements with user, other components, other existing SW systems
        - Exploitation and maintenance requirements
        - User documentation requirements
    - This activity is part of the Common Analyze Process (S), because usually is developed not a single solution but a class of solutions solving the multitude of the problem requirements, from which the optimum solution had to be selected
    - Defining **Validation Tests Plan** which elaboration is considered part of this activity
- (3) **System Architecture Design** – consists in elaboration of a set of documents referring to:
    - The **HW components** of the system and their interconnection modalities
    - **SW configuration elements** and their assignation to the HW components
    - **Manual operations** allowed by the system
    - User and SW configuration elements **Interfaces**
    - **High level architecture** of the SW configuration elements (their components), interfaces between components and the general structure of their data base (if necessary)
    - Preliminary version of the **user and administration manuals**
    - Preliminary version of the **Integration Test Plan*.***
- (4) **Detailed SW Design** – consists in elaboration of a set of documents which details the basic design. It consists in:
    - **Detailed project** of each SW component identified in the design phase. That presumes:
        - Component decomposition in SW units (the level of detail reaching classes in OO approach),
        - The specification of the role, of the interface, as well as the specific life cycle for each unit.
        - The detailed design must allow the direct codification of the components without other supplementary information.
    - Detailed project of the structure of the **data basis**.
    - The **SW Units' Test Plan** designated to test SW units
    - Up-dating of the **Integration Test Plan**

- (5) **Codification** – refers to the codification of the SW components
- (6) **Test of the written code** – it's named also *SW Qualification Test.* It is accomplished on the base of the **SW Units' Test Plan**
  - The results of the tests are documented as **Test Reports**
  - Encountered problems (bugs) are solved based on Problems Solving Process (S)
- (7) **System integration** – presumes activities related to integration of the SW elements with HW elements and with the other existing systems
- (8) **Integration test** – known also under the name of *System Qualification Test*
  - Presumes verification of the correctness of the system functionality as a hole.
  - Based on the **Integration Test Plan*.***
  - The results of the tests are documented as **Test Reports**
  - Encountered problems (bugs) are solved based on Problems Solving Process (S)
- (9) **SW Installing** – presume installation and configuration of the SW product on the target environment. The typical tasks of this activity are:
  - Elaboration of the **Installing Plan** which refers to:
    - Specification of the necessary actions and resources
    - Sharing responsibilities between developer and purchaser (user)
    - Establishing the conditions for data migration if an old existing system is replaced
  - Effective installing of the system in accordance with the **Installing Plan**
  - The events and results of the installing process are registered in specific documents in accordance with Documentation Process
- (10) **Validation System Support** – is given by the developer to the system user and consists in:
  - Assistance in validation tests execution
  - Validation of the system conformity with specified requirements
  - The test results are registered in specific documents in accordance with Documentation Process (S)
  - Any encountered problem is fixed in accordance with Problems Solving Process (S)
  - The successfully end of this activity, usually presume the end of the Development Process (P)

## 5. LIFE CYCLES IN A SW PROJECT DEVELOPMENT PROCESS

- Regardless of how software development is achieved, it must proceed through certain **steps** or **development phases**
- After software is developed it must be **supported** (i.e., maintained).

- The combination of software development phases and support activities phases is referred to as the **Software Life Cycle(*) SWLC** or **SWPLC (Software Project Life Cycle)**

- **(*)Definition:** ***SW Life Cycle*** *is the abstract description of the structured, methodological development and modification process typically showing the main stages in producing and maintaining executable software* – (John McDermid, *"The Software Engineer's Reference Book"*)
  - SW Life Cycle is part of SW Development Process (P), in fact is an activity of the mentioned process
- There are a number of software development techniques organizations can use, each having a different impact on software costs
- In practice, the logic of temporal organization of the activities during the SW Project Development Process (P) imposed the development of some templates (models) for project life cycles (PLC's) applicable to different types of projects.

## 5.1 THE WATERFALL LIFE CYCLE

- WATERFALL LIFE CYCLE – is the classical linear model, the oldest life cycle. Applicable to:
  - Low complexity projects
  - Very well initial defined requirements

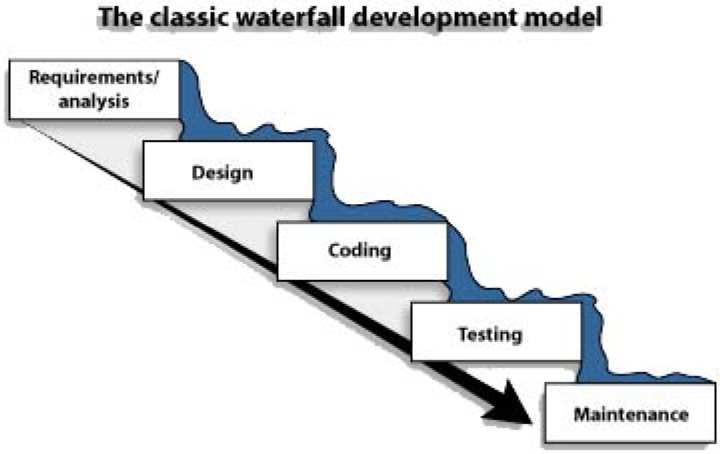

**Fig. 5.1.a.** The Waterfall Life Cycle

## 5.2 THE "V' LIFE CYCLE

- "V" LIFE CYCLE – also a linear cycle. Requires:
  - A set of very well initial specified requirements

- An user with disposability to collaborate and to participate effectively to the project specification process

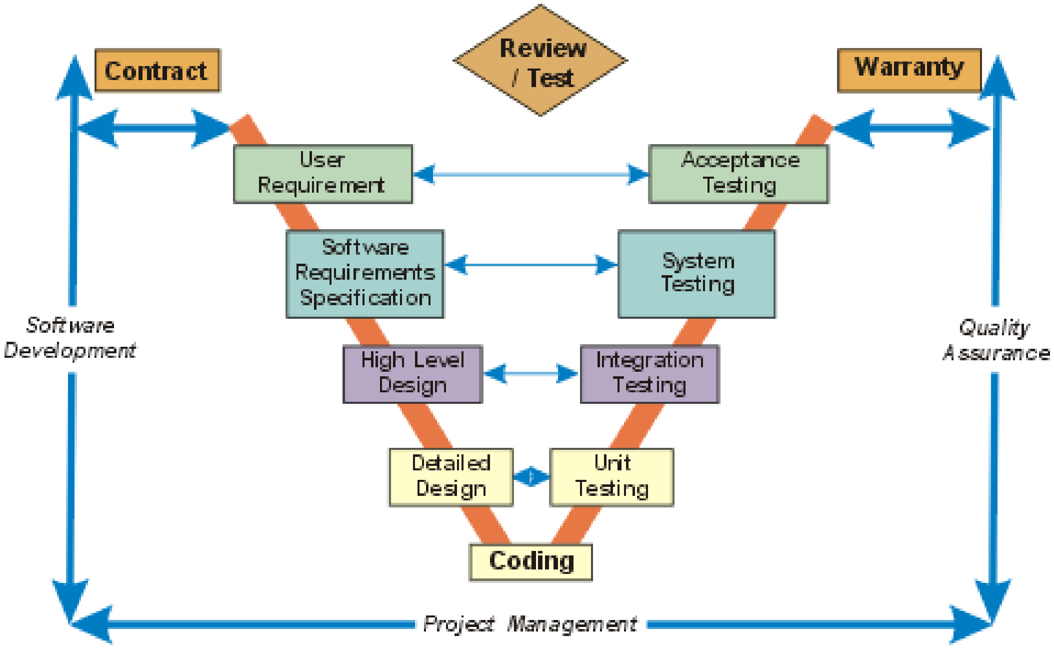

**Fig.5.2.a.** “V” Life Cycle

### 5.3 THE PROTOTYPING

- PROTOTYPING– recommended in the case of projects in which the client can’t participate, or is not interested in producing a list of well defined requirements
    - Analyze and even design activities are **iterative**
    - The result is an “**obtaining-validation-correction**” cycle applicable to product prototype
    - Requires specific rapid and efficient prototypes developing tools
    - It can be used as a **preliminary** development cycle, followed by another type of cycle for the finally product development

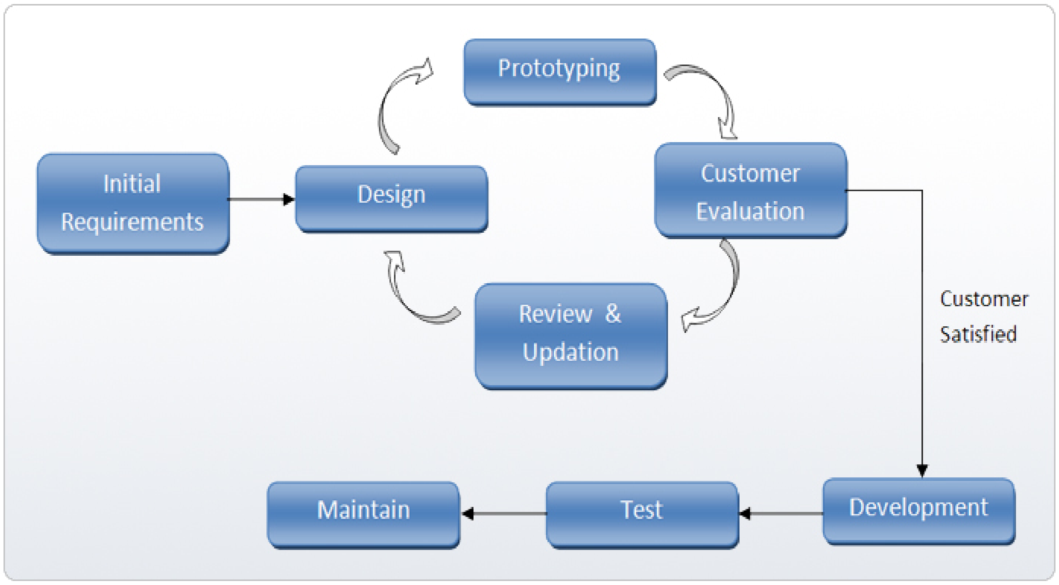

**Fig.5.3.a.** Prototyping Life Cycle

## 5.4. EVOLUTIONARY MODEL

- Based on versions
- Each version is derived quickly
- This process is named iteration
- Each version is send to the customer which analyze it an supplies feed back
- Each iteration is based on a Waterfall Life Cycle
- RUP (Rational Development Process) is based on this model

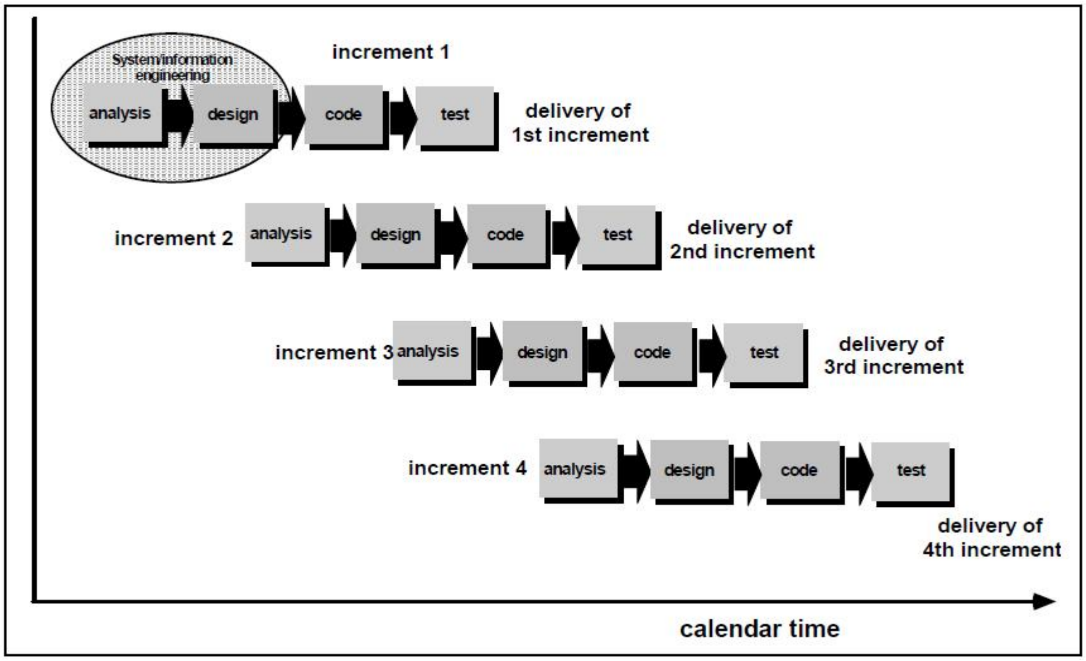

**Fig.5.4.a.** Evolutionary Life Cycle

## 5.5 THE SPIRAL LIFE CYCLE (BOEHM)

- SPIRAL LIFE CYCLE (Boehm) – recommended for projects:
  - With high development **risks**
  - With high **complexity**
  - Which requires very **special technologies**
  - For which is **not** precisely known the modality of solving customer requirements
  - Are very **expensive**
  - Spiral Cycle can be considered as a **repetition** of linear cycles, each new cycle added to the spiral, adding in the same time new facilities to the product

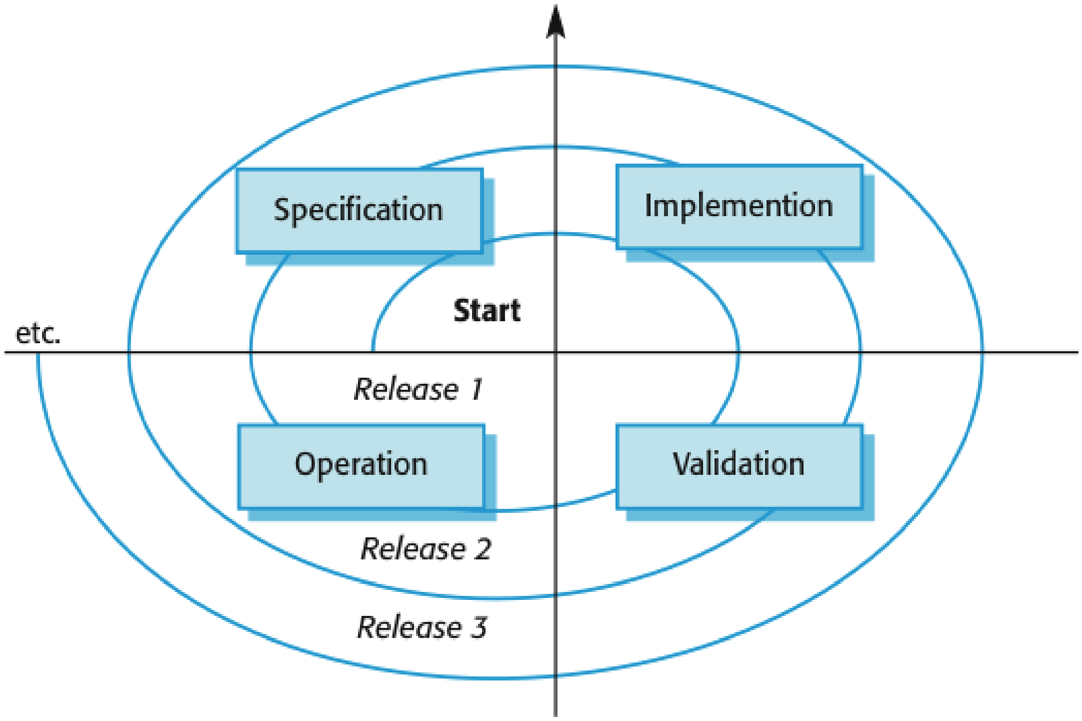

**Fig. 5.3.a.** Spiral Life Cycle

## 5.6 FORMAL SYSTEM DEVELOPMENT

- Uses mathematical methods
- Formal specification based on Stepwise Refinement
- Formal approach, no tests are necessary
- Useful for non-interactive systems, but require experts

## 5.7 IEEE/EIA 12207 “STANDARD FOR INFORMATION TECHNOLOGY - SOFTWARE LIFE CYCLE PROCESSES

- **IEEE/EIA 12207** “STANDARD FOR INFORMATION TECHNOLOGY - SOFTWARE LIFE CYCLE PROCESSES” is a **generic software process** used as a framework for many systems currently being developed or supported
  - IEEE/EIA 12207 defines a set of recommended development activities and documentation alternatives for software intensive systems.
  - This standard is compatible with a number of different software development methods, including the waterfall model.
  - IEEE/EIA 12207 defines a **standard software hierarchy** and an **associated terminology** (commonly used for complex **DOD** software and other complex **MIS** systems)

### 5.7.1 The Standard Software Hierarchy

- Generally, **a System** (e.g., F-22 fighter aircraft) is first partitioned into **various Subsystems** (e.g., avionics), and at times, **prime items** and **critical items** (e.g., attack radar)

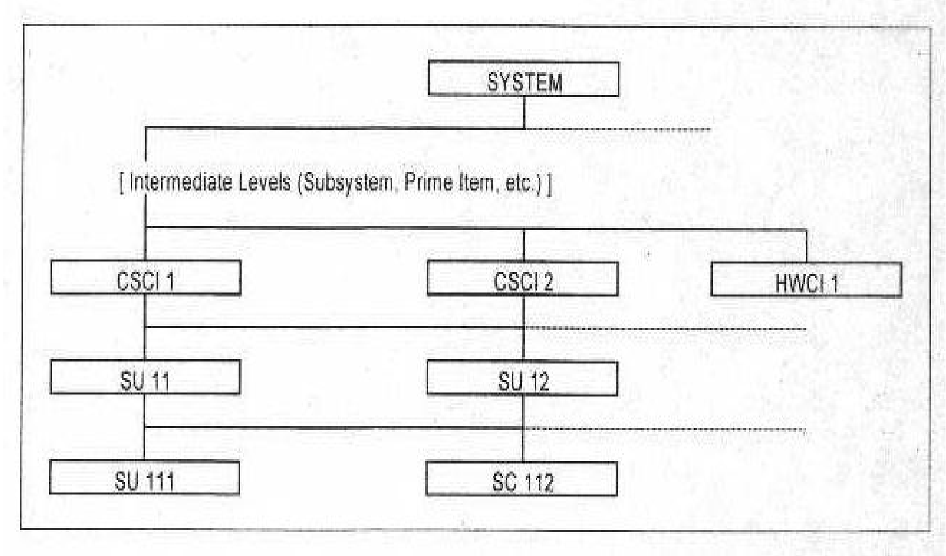

**Fig.5.7.1.a.** IEEE/EIA 12207 Standard: Software Hierarchy

- These **Subsystems** or **items** are further partitioned into:
  - (1) **Hardware Configuration Items** (**HWCI**s)
  - (2) **Computer Software Configuration Items** (**CSCI**s)
- A **CSCI** **(Computer SW Configuration Item)** is defined as an aggregation of software that satisfies a common end-use function.
  - When **CSCI**s are large (e.g., exceed 100,000 LOCs), they are again partitioned into more manageable tiers, called **Software Units** (**SU**s).

  - The lowest-level **SUs** generally contain between 100 and 200 lines of code (LOCs).
  - The structure and number of SU tiers depends on the nature and complexity of the particular CSCI.
- The **Standard Software Hierarchy** structure is also an example of a **product-oriented WBS (Work Breakdown Structure)**.
  - A **WBS** is a management technique used to subdivide a total system into its component elements.
  - **WBS**s are generally product-oriented family trees composed of hardware, software, services, and other work tasks.
  - A **WBS** defines the product(s) to be developed, and relates the work elements to each other and the end product.

### 5.7.2 The IEEE/EIA 12207 Software Development Process

- The **software development process** defined by IEEE/EIA 12207 standard is based on **Phase Model**
- The figure 5.7.1.a shows the software life cycle phases associated with the Waterfall Model

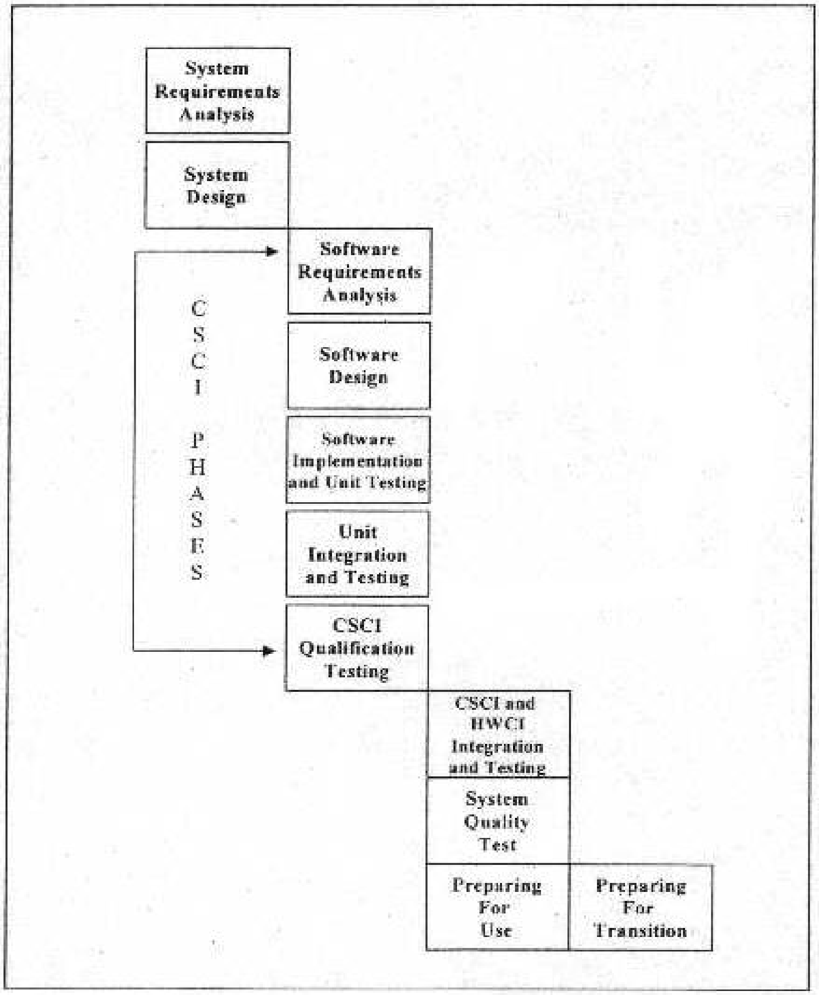
System Requirements Analysis
System Design
Software Requirements Analysis
Software Design
Software Implementation and Unit Testing
Unit Integration and Testing
CSCI Qualification Testing
CSCI PHASES
CSCI and HWCI Integration and Testing
System Quality Test
Preparing For Use
Preparing For Transition

**Fig.5.7.1.a.** IEEE/EIA 12207 Standard: Software Waterfall Model Development Phases

- (1) During the *first two phases* (**System Requirements Analysis Phase** and **System Design Phase**), *the system level requirements* are **partitioned** into **CSCI** and **HWCI** *level requirements*
- (2) **Each CSCI** is then developed using a **CSCI life cycle process** similar to that shown in slide (*CSCI phases*):
  - (2.1) **Software Requirements Analysis Phase**, the specific CSCI *requirements* are defined in detail.
  - (2.2) **Software Design Phase**, *software requirements* are further *refined* to the **SU *level*** and *partitioned into* modules where functions, inputs, outputs, and constraints are defined.
  - (2.3) Once software is completely designed, it can be **coded**
    - According to *Cheadle*, CSCI requirements and design phases may account for 60% of the entire CSCI development effort, whereas coding may account for only 20%
  - (2.4) **Software Implementation and Unit Testing Phase:** *writing source code* (e.g., C++ language statements) for each SU and *testing* each SU;
  - (2.5) **Unit Integration and Testing:** aggregates of SUs;
  - (2.6) **Qualification Testing Phase:** performing of specific tests on the overall CSCI to insure all requirements are successfully met.
- (3) **CSCI and HWCI Integration and Testing Phase:** after individual CSCIs are tested, aggregates of HWCIs and CSCIs are *integrated and tested*.
- (4) **System Quality Test Phase:** *qualification testing* is performed on the entire system to ensure the system-level requirements are met.
- (5) **Preparing for Use** and **Preparing for Transition Phases:** after all testing are complete, the software is *transferred* to the using and supporting agencies.
- During each software development phase, a number of other **key activities** may occur such as:
  - (1) *Software project management,*
  - (2) *Software configuration management,*
  - (3) *Software quality assurance*.
- As discussed by *Boehm* and *Reifer*, each of the activities performed for each discipline, during each phase, can be organized into an activity **WBS** for each **CSCI**.
  - This WBS can be used with the product WBS as a basis for *management reporting* and *tracking* for the CSCI.
- The software life cycle (i.e., CSCI) phases shown do not have to occur sequentially, as the illustration may imply.
- Many modern development practices can result in a different order of activities or even in a combination (or overlapping) of activities.
  - That explains the occurrence of a multitude of alternative software development methodologies.

- The type of software development methodology used generally has a significant impact on the software development and maintenance total life cycle costs.

## 6. THE MANAGEMENT PROCESS

- It belongs to the **Organizational Processes**
- It's simple in its essence: plan and control
- Actually it can become of great complexity due to the multiple activity directions which must be handled.
- The **Management Process** activities are:

### 6.1 ACTIVITIES

- (1) **Initiation and establishing of the application domain** – this activity presume:
  - (1.1) Identification of the process to be handled
  - (1.2) Establishing the process requirements;
  - (1.3) Establishing the process feasibility (analyzing the existence and the adequacy of the human, material, technological, environmental and time resources in accordance with the project scope)
  - (1.4) Depending on necessities, is allowed to adapt the process requirements to the available resources, with previous accord of the all implied parts
  - (1.5) Establish and analyze the project risks
- (2) **Planning**– presume the carry out of the execution plans of the target process tasks.
  - In principle these **plans** refer to:
    - (2.1) The description of the activities
    - (2.2) The description of the associated tasks
    - (2.3) The identification of the products to be delivered
  - A **plan** includes:
    - (a) Temporal restrictions imposed by the process environment (execution deadlines);
    - (b) Effort estimation;
    - (c) Identification of the properly resources;
    - (d) Tasks assignation and establishing responsibilities;
    - (e) Establishing schedules and deadlines for all activities;
    - (f) Identification and evaluation of the risks associated with each task and setting down the adequate counteract measures;
    - (g) The preparation of the environment and of the infrastructure;

- (h) Setting down the quality control measures to be used inside the process;
- (i) Establishing the costs associated to the process execution.

- (3) **Execution and control** – presumes the following tasks:
  - (3.1) Initiation of the plan implementation
  - (3.2) Monitoring the process execution and progress reporting to the customer as well as to the upper management
  - (3.3) Investigation, analysis and identification of the solution for any discovered problem.
  - (3.4) Supervising the application of the solution and register the problem fixing
  - (3.5) If the problem solving implies the modification of the plan, the manager is responsible with the assurance that the impact is determined, controlled and monitored
  - (3.6) Monitoring the risk levels associated to each task and appliance in real-time of the required measures
- (4) **Analysis and evaluation** – this activity presumes:
  - (4.1) Evaluation (testing) of the SW products and finished tasks for conformity with requirements
  - (4.2) Analysis of the results of the evaluation
- (5) **Finalization** – presumes that the project manager:
  - (5.1) Makes sure that the process is finished and all its objectives have been reached (all the planed criteria were accomplished)
  - (5.2) Verifies the documents completeness, mainly the records referring to the developed SW products and to the executed activities and tasks
  - (5.3) Archives the results and the documents on a adequate and stabile support

## 7. LEADERSHIP AND MANAGEMENT

### 7.1 GENERAL TERMS DEFINITION:

- The **Wordweb Thesaurus Dictionary/ Merriam-Webster Dictionary**:
  - **Leadership** means**:**
    - The activity of leading
    - A position as leader of a group, organization, etc,
    - The body of people who lead a group
    - The status of a leader
    - The power or ability to lead other people
  - **Management** means**:**
    - The act or art of managing something: the conducting or supervising of something (as a business)

- The act or skill of controlling and making decisions about a business, department, sports team, etc.
- Judicious use of means to accomplish an end
- The collective body of those who manage or direct an enterprise
- Those in charge of running a business

- **Peter Druker and Warren Bennis**:
  - Management means doing the things right
  - Leadership means doing the right things
- **Stephen R.Covey**:
  - *Leadership deals with the top line: What are the things to accomplish?*
  - *Management is bottom up line focus: How can I best accomplish certain things?*
  - *Management is efficiency in climbing the ladder of success;*
  - *Leadership determines whether the ladder is leaning against the right wall.*
  - *Proactive powerful leadership must constantly monitor environmental change, particularly customer buying habits and motives, and provide the force necessary to organize resources in the right direction.*
  - *The managers identify the tools, write policy and procedure manuals, hold training programs, bring improved technologies, set up working schedules and compensation programs.*
  - *No management success can compensate for failure in leadership.*

## 7.2 SPECIFIC TERMS DEFINITION:

- **Management of a SW Project:** *comprises all Activities and Tasks to plan and control the Activities of Staff Members, thus a Target can be reached which that can not be reached by the Staff on its own.*

  (Balzert, H. *Lehrbuch der Software-Technik*, Spektrum, 1998)

- **Software Engineering:** *Target oriented Deployment (e.g. in terms of Costs, Time, Quality) and systematic Use of Methods, Concepts, Notations and Tools for engineer type Development and Application of complex Software Systems.*

  (Balzert, H. *Lehrbuch der Software-Technik*, Spektrum, 1998)

- **Software Engineering:** Technical Subject, Engineering Discipline

- **Project Management:** Non-technical Organization of the Development Process

  - Presumes strong **Interfaces** to
    - Project Execution Model/Process Models
    - Quality Management
    - Configuration Management

- **Project Management** comprises all **Tasks** during the **Execution of a Project,** such as
    - Preparation (Structure and Staff)
    - Planning
    - Supervision
    - Control

- Beyond the **Scope** of an individual **Project,** the following **Tasks** exist**:**
    - Finalization of Overall Project
    - Improvement of Processes
    - People Management

- The **Task of a Project Manager** is to make sure that the Software Project is carried out according Budget and Time and to deliver Software that contributes to reach given economical Targets.

(Sommerville, I. *Software Engineering*, Pearson, 2001)

## Exercise #1

1) Describe the main characteristics of the software  production activity
2) Which are the reasons to organize the SW Project development?
3) Define the following terms: *Production Process, SW Project*
4) What is ISO/CEI 12207:1995 Standard? Define the following terms: *Processes, Activities, Tasks*
5) Describe the  primary, support and organizational processes
6) Which are the activities of the *Development Process*?
7) What kind of *Life cycle in SW Project development* do you know?
8) Which are the activities of the *Management Process*?
9) Reveal the differences between the terms *leadership* and *management*
10) Define the following concepts: *Software Engineering* and *Software Project Management*? What is the differentiation between them?