SOFTWARE ENGINEERING FUNDAMENTALS

**Layered Technology**

* Tools
* Methods
* Process
* Quality Focus

**Umbrella Activities**

* Software Tracking & Control
* Risk Management
* Software Quality Assurance (SQA)
* Formal Technical Reviews (FTR)
* Measurement
* Software Configuration Management (SCM)
* Reusability Management
* Work Product preparation & production

**Software Development Myths**

* Management Level Myths
* Customer Level Myths
* Practitioner Level Myths

**Management Level Myths**

* No need to change approach to software development
* We can buy software tools & use them
* When needed we can add more programmers for faster software development
* If the developer outsorces project to a 3rd party, he is off the hook
* Book contains standards & procedures for developing software which provides everything
* Organization has state-of-the-art development tools

**Customer level myths**

* Only general statement is sufficient and no need to mention detailed project requirements
* Project requirements continuously change but can be easily accomodated in software

**Practitioner level Myths**

* Once we write program & get it to work, our job is over
* Until I get program running, I have no way of assessing its quality
* When the project is successful, deliverable product is the only working program
* The software engineering process creates larger and unnecessary documentation & ultimately it will slow down the process
* Generic Process Model (CPMCD)
* Communication
* Planning
* Modelling
* Construction
* Deployment
* The Waterfall Model
* Communication : Project Initiation, Requirements
* Planning : Estimation, Scheduling, Tracking
* Modelling : Ananlysis, Design
* Condtruction : Coding, Testing
* Deployment : Delivery, Support, Feedback
* Advantages of Waterfall Model
* Simple, easy to understand
* Systematic, sequential approach
* Each phase at one time, avoids overlapping
* Easy to manage
* Establishes milestones
* Disadvantages of Waterfall Model
* Problems remain uncovered until testing
* Blocking states, dependency on previous phase
* Requirements at the beginning
* Slower due to step-by-step
* Not realistic, useful for small projects
* Does not incorporate risk management