



## Engr. Muhammad Fahad Khan

- ✓ Lecturer Software Engineering Department U.E.T Taxila
- ✓ Bsc Software Engineering (U.E.T TAXILA)
- ✓ Msc Computer Engineering (U.E.T Taxila)  
(Gold Medalist)
- ✓ Ph.D Computer Engineering (In Progress)

Copyright 2011 @ M.Fahad Khan

### List of my research papers published in conferences & journals

- ✓ Research paper titled "**Efficient Car Alarming System for Fatigue Detection during Driving**" accepted and will publish in 2011 3rd International Conference on Information and Multimedia Technology (ICIMT 2011), 28-30 Dec 2011.
- ✓ Research paper titled "**Survey on Patient Information Systems**" accepted and will published in IJCSI Volume 8, Issue 6, November 2011 which is scheduled to be published on Wednesday 30th November 2011.
- ✓ Research paper titled "**Video Analytics for Quantitative Employee Performance Evaluation**" published in Canadian Journal on Image Processing and Computer Vision, Vol. 1, No. 1, February 2010.
- ✓ Research paper titled "**License Plate Recognition Algorithm for Pakistani License Plates**" published in Canadian Journal on Image Processing and Computer Vision Vol. 1, No. 2, April 2010.
- ✓ Research paper titled "**Comprehensive Analysis and Enhancement of Steganographic Strategies for Multimedia Data Hiding and Authentication**" published in International Journal of Computer and Network Security, Vol. 2, No. 3, March 2010.

Copyright 2011 @ M.Fahad Khan

### Grading Criteria

- ✓ 20% Mid Exam
- ✓ 10% Quizzes
- ✓ 10% Assignment
- ✓ 60% Final Exam

Copyright 2011 @ M.Fahad Khan

### Objectives

- ✓ To introduce software process models
- ✓ To describe three generic process models and when they may be used
- ✓ To describe outline process models for requirements engineering, software development, testing and evolution
- ✓ To explain the Rational Unified Process model
- ✓ To introduce CASE technology to support software process activities

### Topics covered

- ✓ Software process models
- ✓ Process iteration
- ✓ Process activities
- ✓ The Rational Unified Process
- ✓ Computer-aided software engineering

## What is software Engineering?

Software engineering is an engineering discipline that is concerned with all aspects of software production.



Copyright 2011 @ M.Fahad Khan

## Software Engineering



Copyright 2011 @ M.Fahad Khan

## What is the difference between software engineering and computer science?

- Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.
- Computer science theories are still insufficient to act as complete underpinning for software engineering (unlike e.g. physics and electrical engineering).

Copyright 2011 @ M.Fahad Khan

## What is the difference between software engineering and computer science?



Computer Science



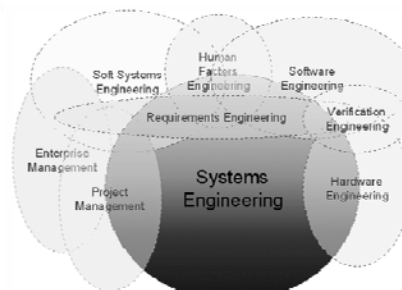
Copyright 2011 @ M.Fahad Khan

## What is the difference between software engineering and system engineering?

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this process concerned with developing the software infrastructure, control, applications and databases in the system.
- System engineers are involved in system specification, architectural design, integration and deployment.

Copyright 2011 @ M.Fahad Khan

## What is the difference between software engineering and system engineering?

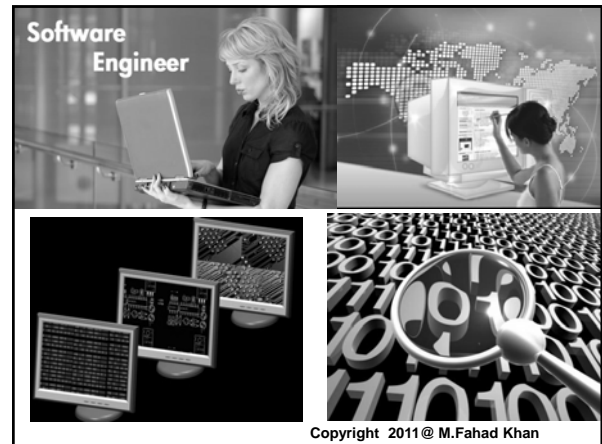


Copyright 2011 @ M.Fahad Khan

## SOFTWARE ENGINEERS

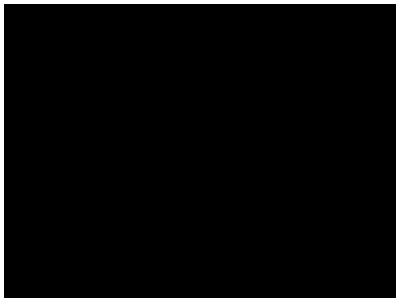
### THE ARCHITECTS OF THE FUTURE !

Copyright 2011 @ M.Fahad Khan



Copyright 2011 @ M.Fahad Khan

## A DAY IN SOFTWARE ENGINEER'S LIFE



Copyright 2011 @ M.Fahad Khan

## SOFTWARE ENGINEERS

- You can change the world as some of the people in this field did.
- Bill Gates
- Steve Jobs

Copyright 2011 @ M.Fahad Khan

## BILL GATES

1. Inspiring the era of the home computer
2. Commercializing the operating system
3. Launching Windows 95
4. Becoming the richest man in the world
  - ✓ Bill Gates became the richest man on the planet at the age of 39. He stayed at number one on the Forbes list of the World's Richest People until 2007.
  - ✓ According to Wikipedia, Bill Gates was worth more than \$101 billion in 1999; while Wired reports that, when Microsoft stock hit a high in 1996, Gates was earning \$30 million a day.
5. Giving his money away

Copyright 2011 @ M.Fahad Khan





**STEVE JOBS**

1. The Invention of Apple
2. Jobs's Marketing and Branding Strategies
3. iCloud
4. The Apple iPhone
5. Involvement in the Disney / Pixar collaboration

Copyright 2011 @ M.Fahad Khan



**Software Processes**

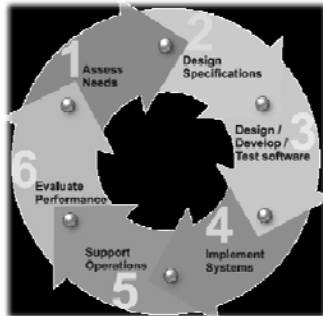
Copyright 2011 @ M.Fahad Khan

**What is a software process?**

- A set of activities whose goal is the development or evolution of software.
- Generic activities in all software processes are:
  - Specification - what the system should do and its development constraints
  - Development - production of the software system
  - Validation - checking that the software is what the customer wants
  - Evolution - changing the software in response to changing demands.

Copyright 2011 @ M.Fahad Khan

## What is a software process?



Copyright 2011@ M.Fahad Khan

## What is a software process model?

- ✓ A simplified representation of a software process, presented from a specific perspective.
- ✓ Examples of process perspectives are
  - Workflow perspective - sequence of activities;
  - Data-flow perspective - information flow;
  - Role/action perspective - who does what.
  - Waterfall;
  - Iterative development;
  - Component-based software engineering.

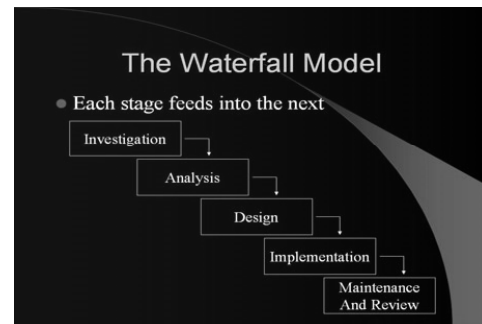
Copyright 2011@ M.Fahad Khan

## Generic software process models

- **The waterfall model**
  - Separate and distinct phases of specification and development.
- **Evolutionary development**
  - Specification, development and validation are interleaved.
- **Component-based software engineering**
  - The system is assembled from existing components.
- **There are many variants of these models** e.g. formal development where a waterfall-like process is used but the specification is a formal specification that is refined through several stages to an implementable design.

Copyright 2011@ M.Fahad Khan

## Waterfall model



Copyright 2011@ M.Fahad Khan

## Waterfall model phases

- Requirements analysis and definition
  - System and software design
  - Implementation and unit testing
  - Integration and system testing
  - Operation and maintenance
- The main drawback of the waterfall model is the difficulty of accommodating change after the process is underway. One phase has to be complete before moving onto the next phase.

Copyright 2011@ M.Fahad Khan

## Waterfall model problems

- Inflexible partitioning of the project into distinct stages makes it difficult to respond to changing customer requirements.
- Therefore, this model is only appropriate when the requirements are well-understood and changes will be fairly limited during the design process.
- Few business systems have stable requirements.
- The waterfall model is mostly used for large systems engineering projects where a system is developed at several sites.

Copyright 2011@ M.Fahad Khan

## Waterfall model

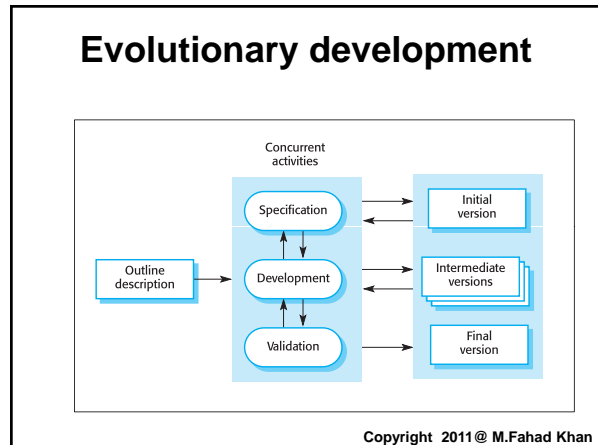
The Rise and Fall  
of  
WATERFALL

Copyright 2011 @ M.Fahad Khan

## Evolutionary Development

- Exploratory development
  - Objective is to work with customers and to evolve a final system from an initial outline specification. Should start with well-understood requirements and add new features as proposed by the customer.
- Throw-away prototyping
  - Objective is to understand the system requirements. Should start with poorly understood requirements to clarify what is really needed.

Copyright 2011 @ M.Fahad Khan



## Evolutionary development

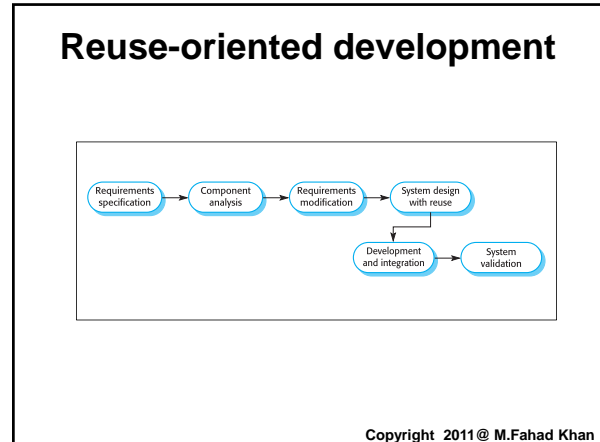
- Problems
  - Lack of process visibility;
  - Systems are often poorly structured;
  - Special skills (e.g. in languages for rapid prototyping) may be required.
- Applicability
  - For small or medium-size interactive systems;
  - For parts of large systems (e.g. the user interface);
  - For short-lifetime systems.

Copyright 2011 @ M.Fahad Khan

## Component-based software engineering

- Based on systematic reuse where systems are integrated from existing components or COTS (Commercial-off-the-shelf) systems.
- **Process stages**
  - Component analysis;
  - Requirements modification;
  - System design with reuse;
  - Development and integration.
- This approach is becoming increasingly used as component standards have emerged.

Copyright 2011 @ M.Fahad Khan



### Process iteration

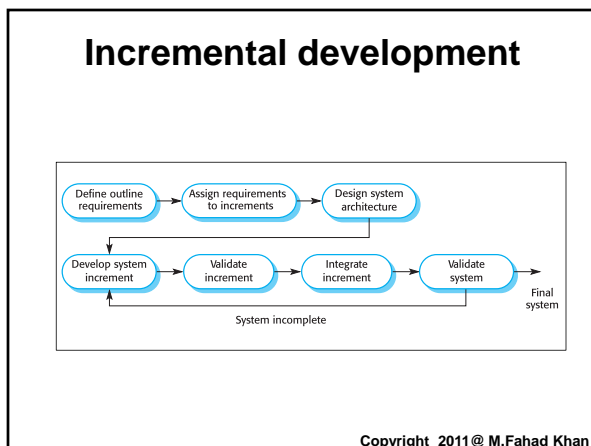
- System requirements ALWAYS evolve in the course of a project so process iteration where earlier stages are reworked is always part of the process for large systems.
- Iteration can be applied to any of the generic process models.
- Two (related) approaches
  - Incremental delivery;
  - Spiral development.

Copyright 2011@ M.Fahad Khan

### Incremental delivery

- Rather than deliver the system as a single delivery, the development and delivery is broken down into increments with each increment delivering part of the required functionality.
- User requirements are prioritised and the highest priority requirements are included in early increments.
- Once the development of an increment is started, the requirements are frozen though requirements for later increments can continue to evolve.

Copyright 2011@ M.Fahad Khan



### Incremental development advantages

- Customer value can be delivered with each increment so system functionality is available earlier.
- Early increments act as a prototype to help elicit requirements for later increments.
- Lower risk of overall project failure.
- The highest priority system services tend to receive the most testing.

Copyright 2011@ M.Fahad Khan

## Incremental development VS Waterfall

a tale of **two teams**



Copyright 2011 @ M.Fahad Khan

## Extreme programming

- An approach to development based on the development and delivery of very small increments of functionality.
- Relies on constant code improvement, user involvement in the development team and pair wise programming.



Test method for projects using agile methodologies such as Extreme Programming

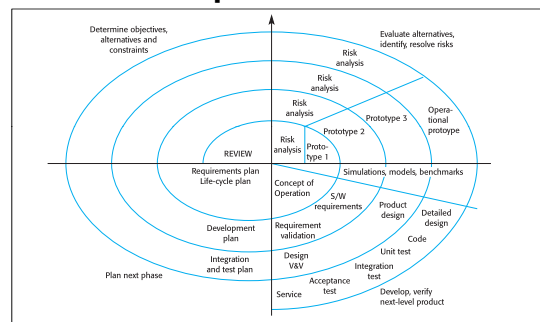
Copyright 2011 @ M.Fahad Khan

## Spiral development

- Process is represented as a spiral rather than a sequence of activities with backtracking.
- Each loop in the spiral represents a phase in the process.
- No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required.
- Risks are explicitly assessed and resolved throughout the process.

Copyright 2011 @ M.Fahad Khan

## Spiral model of the software process



Copyright 2011 @ M.Fahad Khan

## Spiral model sectors

- Objective setting
  - Specific objectives for the phase are identified.
- Risk assessment and reduction
  - Risks are assessed and activities put in place to reduce the key risks.
- Development and validation
  - A development model for the system is chosen which can be any of the generic models.
- Planning
  - The project is reviewed and the next phase of the spiral is planned.

Copyright 2011 @ M.Fahad Khan

## Spiral model of the software process



Copyright 2011 @ M.Fahad Khan

## Structured methods

- Systematic approaches to developing a software design.
- The design is usually documented as a set of graphical models.
- Possible models
  - Object model;
  - Sequence model;
  - State transition model;
  - Structural model;
  - Data-flow model.

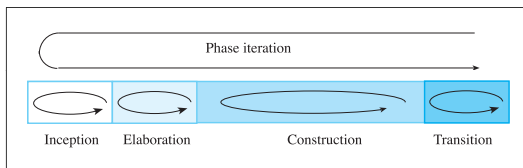
Copyright 2011 @ M.Fahad Khan

## The Rational Unified Process

- A modern process model derived from the work on the UML and associated process.
- Normally described from 3 perspectives
  - A dynamic perspective that shows phases over time;
  - A static perspective that shows process activities;
  - A practice perspective that suggests good practice.

Copyright 2011 @ M.Fahad Khan

## RUP phase model



## RUP phases

- Inception
  - Establish the business case for the system.
- Elaboration
  - Develop an understanding of the problem domain and the system architecture.
- Construction
  - System design, programming and testing.
- Transition
  - Deploy the system in its operating environment.

Copyright 2011 @ M.Fahad Khan

## RUP Good Practice

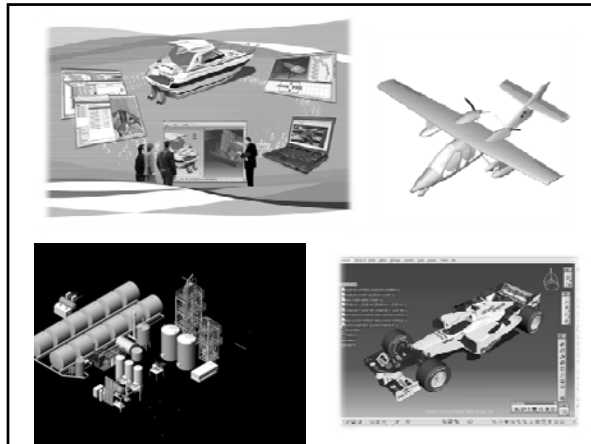
- Develop software iteratively
- Manage requirements
- Use component-based architectures
- Visually model software
- Verify software quality
- Control changes to software

Copyright 2011 @ M.Fahad Khan

## Computer-aided Software Engineering

- Computer-aided software engineering (CASE) is software to support software development and evolution processes.
- Activity automation
  - Graphical editors for system model development;
  - Data dictionary to manage design entities;
  - Graphical UI builder for user interface construction;
  - Debuggers to support program fault finding;
  - Automated translators to generate new versions of a program.

Copyright 2011 @ M.Fahad Khan



## Case Technology

- Case technology has led to significant improvements in the software process. However, these are not the order of magnitude improvements that were once predicted
  - Software engineering requires creative thought - this is not readily automated;
  - Software engineering is a team activity and, for large projects, much time is spent in team interactions. CASE technology does not really support these.

Copyright 2011@ M.Fahad Khan

## CASE Classification

- Classification helps us understand the different types of CASE tools and their support for process activities.
- Functional perspective
  - Tools are classified according to their specific function.
- Process perspective
  - Tools are classified according to process activities that are supported.
- Integration perspective
  - Tools are classified according to their organisation into integrated units.

Copyright 2011@ M.Fahad Khan

## CASE Integration

- Tools
  - Support individual process tasks such as design consistency checking, text editing, etc.
- Workbenches
  - Support a process phase such as specification or design. Normally include a number of integrated tools.
- Environments
  - Support all or a substantial part of an entire software process. Normally include several integrated workbenches.

Copyright 2011@ M.Fahad Khan

## Key points

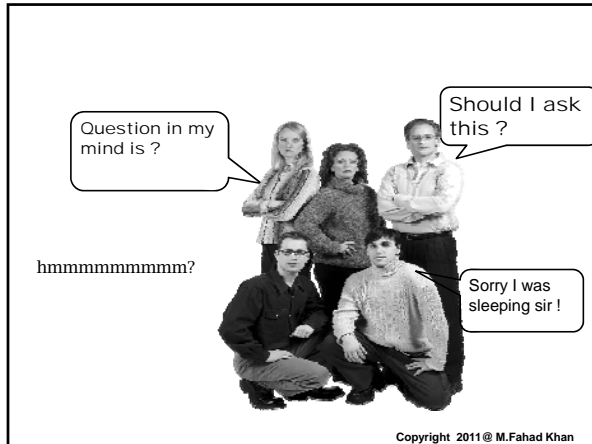
- Software processes are the activities involved in producing and evolving a software system.
- Software process models are abstract representations of these processes.
- General activities are specification, design and implementation, validation and evolution.
- Generic process models describe the organisation of software processes. Examples include the waterfall model, evolutionary development and component-based software engineering.
- Iterative process models describe the software process as a cycle of activities.

Copyright 2011@ M.Fahad Khan

## Key points

- Requirements engineering is the process of developing a software specification.
- Design and implementation processes transform the specification to an executable program.
- Validation involves checking that the system meets to its specification and user needs.
- Evolution is concerned with modifying the system after it is in use.
- The Rational Unified Process is a generic process model that separates activities from phases.
- CASE technology supports software process activities.

Copyright 2011@ M.Fahad Khan



If you have any query please feel free to ask

**Phone:** +92-51-9047-592

**Fax:** +92-51-9047-420

**Email:** fahad.khan@uettaxila.edu.pk

University Of Engineering & Technology Taxila Pakistan

Copyright 2011 @ M.Fahad Khan

A rectangular box containing contact information. The text is centered and includes a general invitation to ask questions, phone and fax numbers, an email address, and the name of the institution: University Of Engineering & Technology Taxila Pakistan. A copyright notice "Copyright 2011 @ M.Fahad Khan" is at the bottom right.