Software Engineering

200511357 조재연

1.Software Engineering

- The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.
- A Layered technology
- Process layer
- “how to’s” for building software
- Automated or semiautomated support

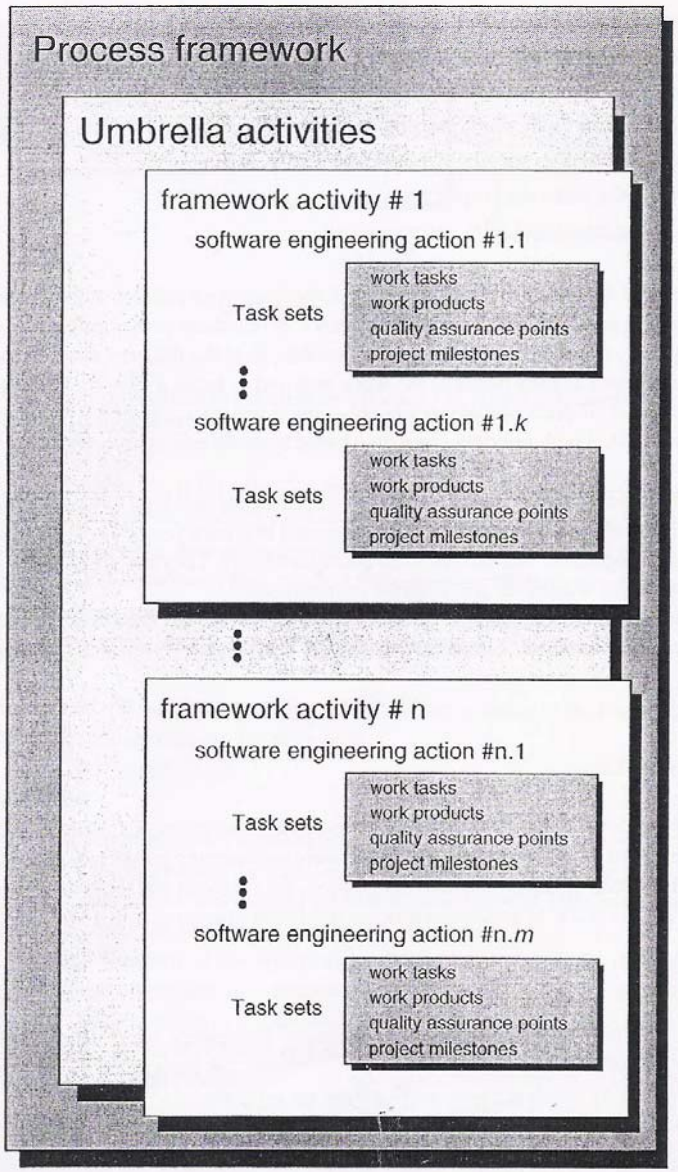


Figure 2 A software process framework.

2. A Process Framework

- Generic process Framework
  - Communication
  - Planning
  - Modeling
  - Construction
  - Deployment
- That can be used during the development of small, simple program.

2. A Process Framework

- Typical activities
  - Software project tracking and control
  - Risk management
  - Software quality assurance
  - Formal technical reviews
  - Measurement
  - Software Configuration management
  - Reusability management
  - Work product preparation and production

3. Software Process Model

- Incorporates a development strategy that encompassed the process, methods, and tools layers described earlier.
- Prescriptive Models
  - The Waterfall Model
  - Incremental Process Models
  - Evolutionary Process Models
  - Specialized Process Models
  - The Unified Process

3.1.1 The Waterfall Model

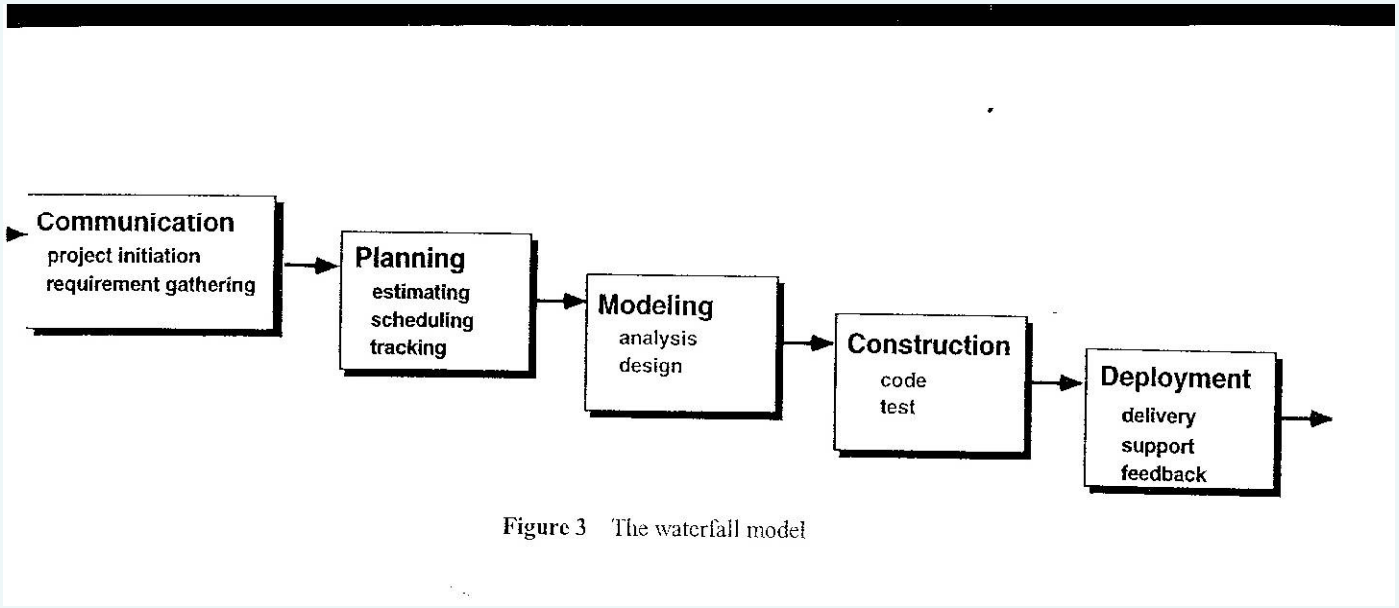
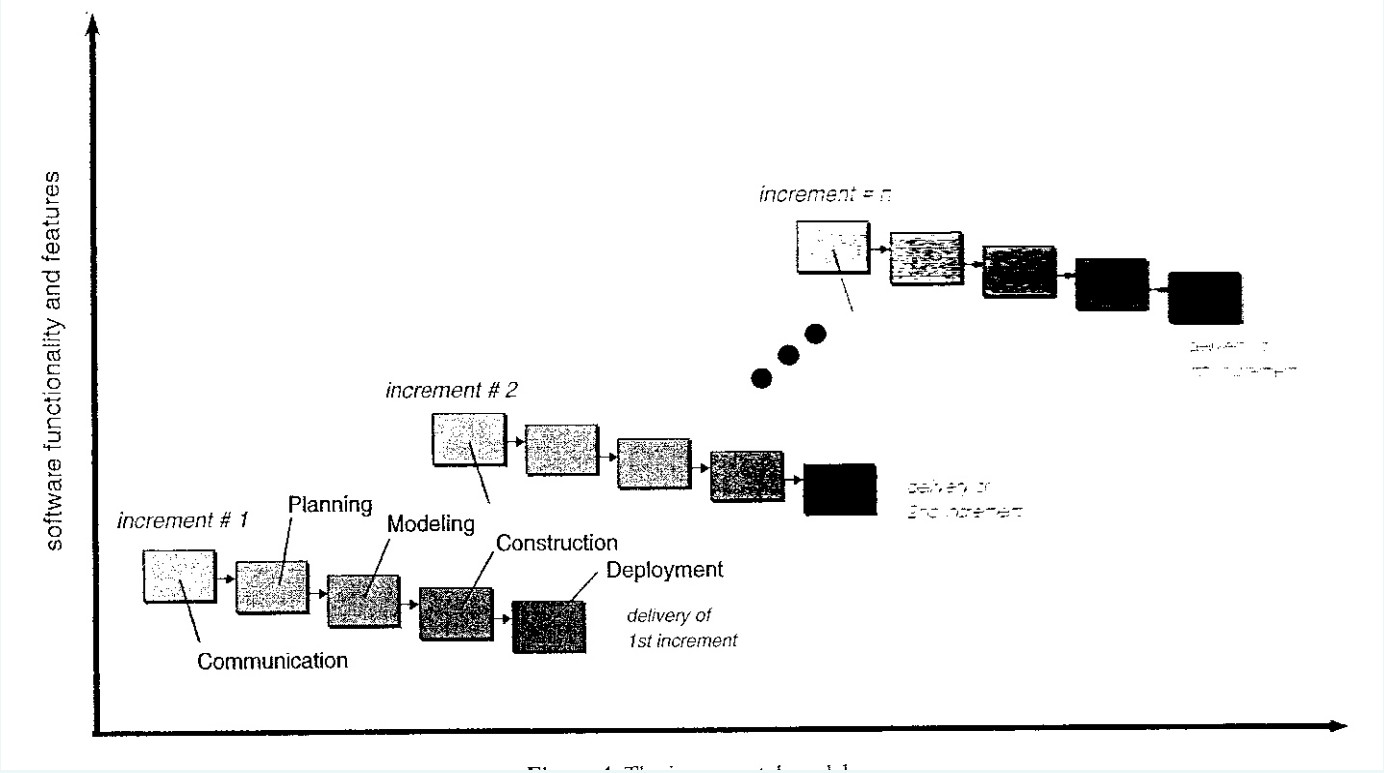


Figure 3 The waterfall model

3.1.2 Incremental Process Models



The Prototyping paradigm

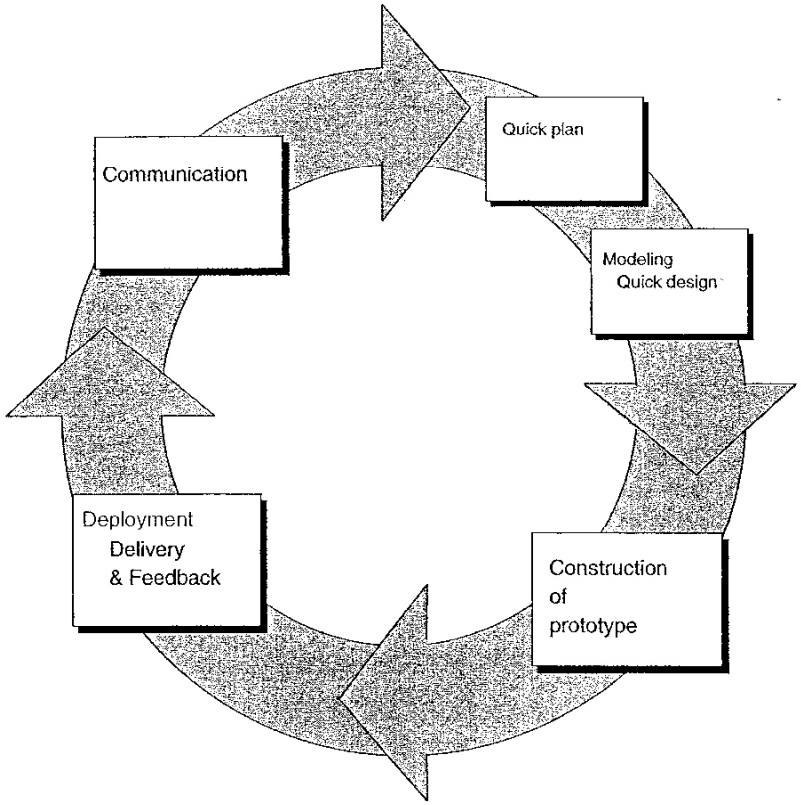


Figure 5 The prototyping paradigm.

Sprial model

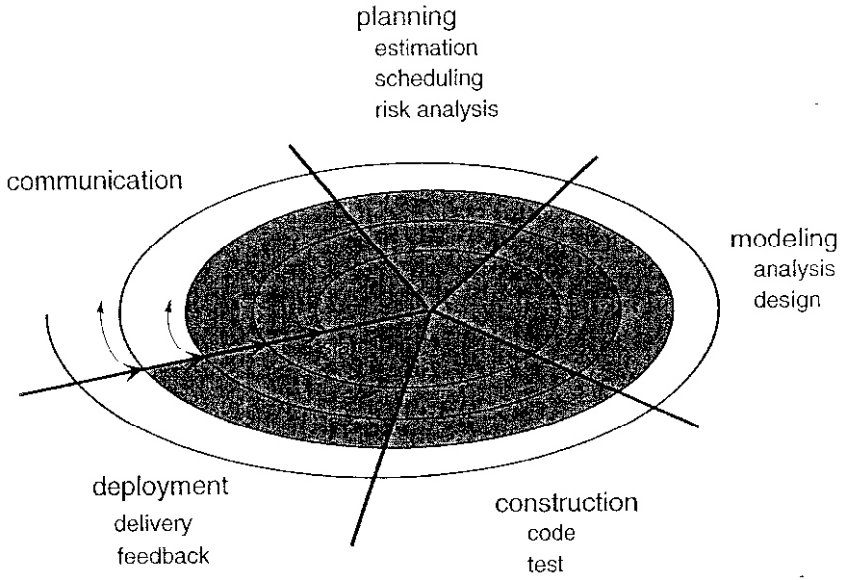


Figure 6 A typical spiral model.

Specialized Process Models

- This model tend to be applied when defined software engineering approach is chosen.
- Component-Based Development
  - Component provide targeted functionality
- The Formal Method Model
  - A set of activities that lead to formal mathematical specification of computer Software
- Aspect Oriented Software Development

The Unified Process

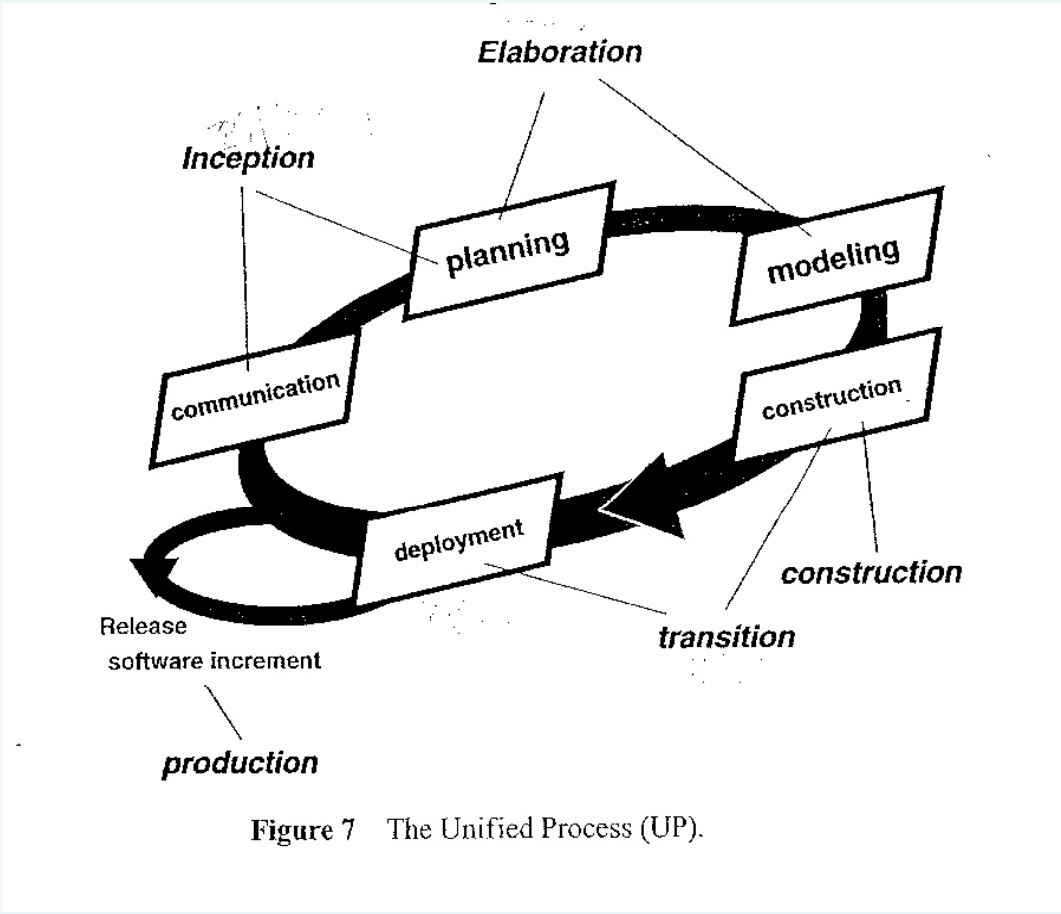


Figure 7 The Unified Process (UP).

3.2 Agile Software Development

- Through this work we have come to value.
  - Individuals and interactions over process and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over Following plan

3.2 Agile Software Development

- Key assumptions about the majority of software projects:
  - It is difficult to predict in advance which software requirements will persist and which will change.
  - For many types of software, design and construction are interleaved.
  - Analysis, design, construction, and testing are not as predicable as we might like.

The Extreme Programming process

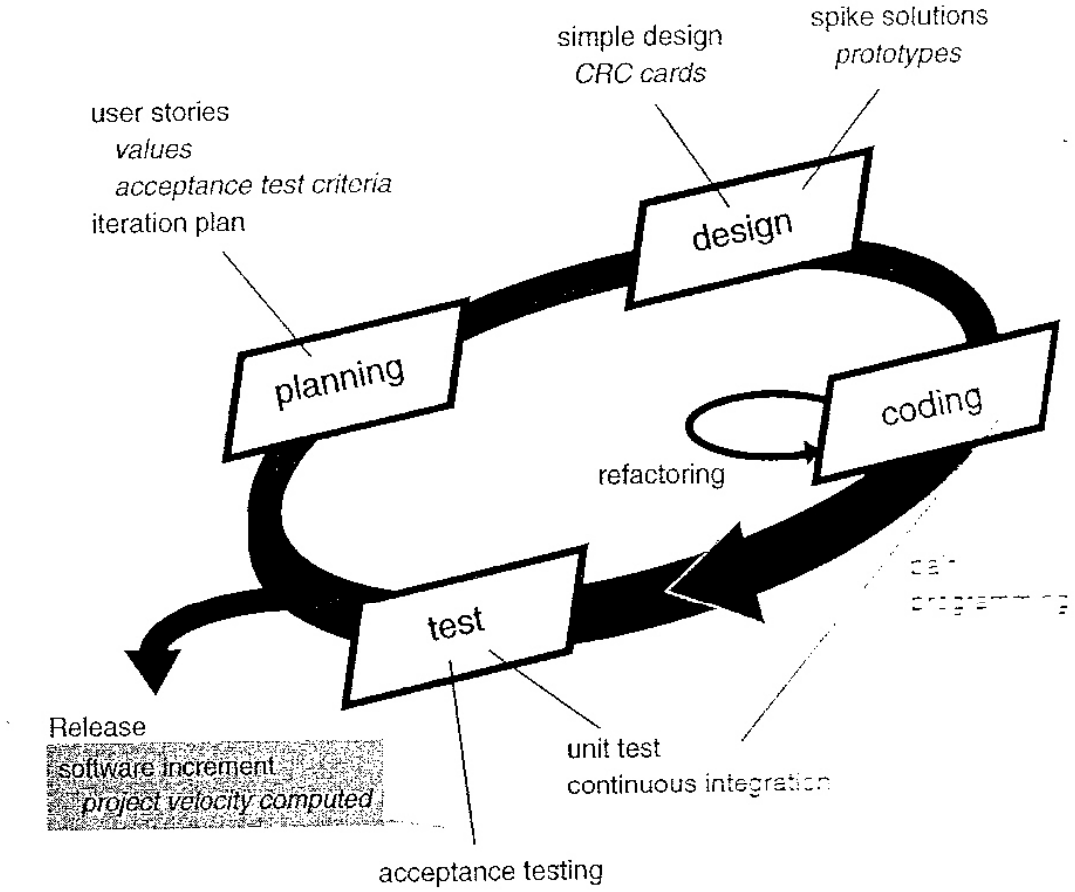


Figure 8 The Extreme Programming process.

4.The Management Spectrum

- People>Problem>Process
  - People
    - Recruiting, selection, performance management, training, compensation, career development, organization, and team and culture development
  - The Problem
    - The software developer and customer must to define project objectives and scope.
  - The Process
    - The Capability Maturity Model Integration (CMMI)

5. Software Project Management

- Measurement and Metrics
- Project Estimating
  - Effort estimation techniques
  - Size-Oriented Estimation
  - Empirical Models
- Risk Analysis
- Scheduling
- Tracking and Control

6.Software Quality Assurance

- Correctness
- Reliability
- Integrity
- Efficiency
- Usability
- Maintainability
- Flexibility
- Testability
- Reusability
- Interoperability

7. Software Configuration Management

- Identify changes
- Control changes
- Ensure that changes are being properly implemented
- Report changes to others who may have an interest

8. The Technical Spectrum

- Software Engineering Methods
  - Conventional software engineering methods
  - Object-oriented approaches
  - Formal methods

8.2 Problem Definition

- Analysis Principles
  - The data domain of the problem must be modeled.
  - The functional domain of the problem must be modeled
  - The behavior of the problem must be modeled.
  - Model of data, function, and behavior must be partitioned
  - The overriding trend in analysis is from essence toward implementation

8.2 Problem Definition

- Analysis Methods
  - Scenario-based elements
  - Class-based elements
  - Behavioral elements
  - Flow-oriented elements

The Design pyramid

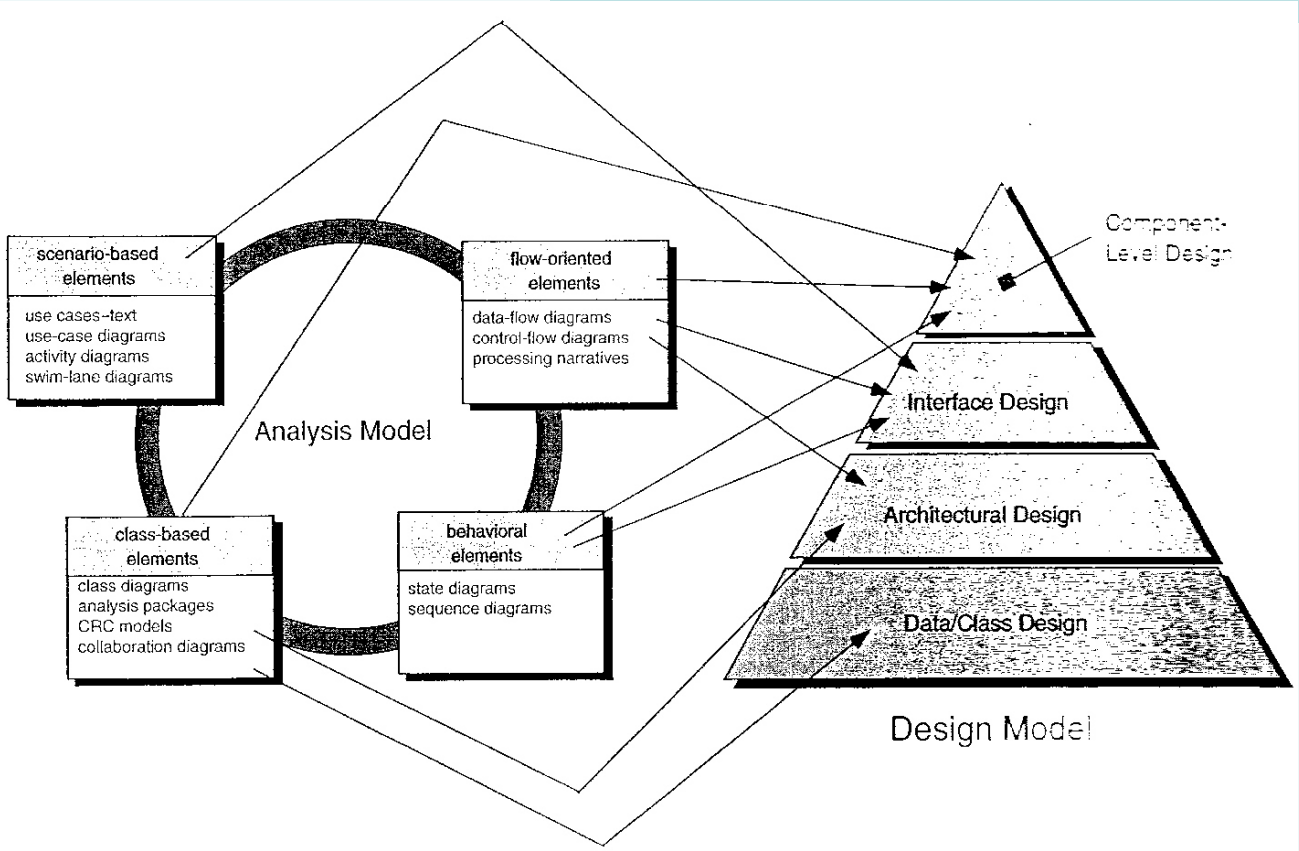


Figure 9 The design pyramid.

8.5 Software Testing

- Objectives
  - Testing is a process of executing a program with the intent of finding an error
  - A good test case is one that has a high probability of finding an as-yet-undiscovered error
  - A successful test is one that uncovers as-yet-undiscovered error
- Strategy
- Tactics
  - Black box testing
  - White box testing

9.Software Engineering Patterns

- Process Patterns
- Analysis Patterns
- Design Patterns
- Testing Patterns

10. The Road Ahead and the Three rs

- Reuse
- Reengineering
- Retooling

11. Summary

- Various questions are asked and reasked about software engineering.