IF2261 Software Engineering

Software Process

Program Studi Teknik Informatika
STEI ITB

Software Process

- The roadmap to building high quality software products
- Adapted to meet the needs of software engineers and managers
- Provides a framework for managing activities
- Different types of projects require different software processes
- Work products are produced by the software process
- The best indicators of how well a software process has worked are the quality, timeliness, and long-term viability of the resulting software product

** SEPA 6th ed, Roger S. Pressman*

Common Process Framework

- Communication
  - customer collaboration and requirement gathering
- Planning
  - establishes engineering work plan, describes technical risks, lists resource requirements, work products produced, and defines work schedule
- Modeling
  - creation of models to help developers and customers understand the requires and software design
- Construction
  - code generation and testing
- Deployment
  - software delivered for customer evaluation and feedback

** SEPA 6th ed, Roger S. Pressman*

SE Umbrella 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

** SEPA 6th ed, Roger S. Pressman*

Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI)

** SEPA 6th ed, Roger S. Pressman*

SEI – CMMI

www.sei.cmu.edu/cmmi

- Level 0: Incomplete
  - process is not performed or does not achieve all goals defined for level 1
- Level 1: Performed
  - work tasks required to produce required work products are being conducted
- Level 2: Managed
  - people doing work have access to adequate resources to get job done, stakeholders are actively involved, work tasks and products are monitored, reviewed, and evaluated for conformance to process description
- Level 3: Defined
  - management and engineering processes documented, standardized, and integrated into organization-wide software process
- Level 4: Quantitatively Managed
  - software process and products are quantitatively understood and controlled using detailed measures
- Level 5: Optimizing
  - continuous process improvement is enabled by quantitative feedback from the process and testing innovative ideas

** SEPA 6th ed, Roger S. Pressman*

SEI - CMMI

Level	Focus
Optimizing	Continous process improvement
Quantitatively Managed	Quantitative management
Defined	Process standardization
Managed	Basic project management
Performed	

** SEPA 6th ed, Roger S. Pressman*

CMM Level 3

- Software **CMM Level 3**

  Foster-Miller achieved SW-**CMM Level 3** certification in December of 2005 to processes as defined by the Software Engineering Institute at Carnegie Mellon **...**
  www.foster-miller.com/software_**cmm_level3**.htm
- Weserv Systems International, Inc. (WeServ), a wholly owned subsidiary of Fujitsu Philippines, Inc., recently passed the Capability Maturity Model for **...**
  www.fujitsu.com/ph/news/pr/20041215.html

CMM Level 4

- **CMM Level 4** Certified Company | Software Application Development **...**

  Trigent is an SEI **CMM Level 4** certified company with development centers in the US and India. Provides information about Trigent's software application **...**
  www.trigent.com/company/**cmm**-certified-company.htm
- On April 16th, Kingdee passed **CMM Level 4** evaluation with the United States' **...** At present, less than 100 software companies pass **CMM Level 4** worldwide and **...**
  global.kingdee.com/en/news/dongtai/76/200**4**-0**4**/5**4**2.htm

CMM Level 5

- Managing IT: Life After **CMM Level 5**

  More than half the world's **CMM Level 5** companies are based in India. Software firms also used **CMM** to establish credentials as developers of quality software **...**
  www.india-today.com/ctoday/20020401/mit2.html
- SEI **CMM Level 5** Wipro is the first software services company in the world **...** We achieved **CMM level 5** certification in June, 1999. As part of the **CMM level ...**
  www.wipro.com/aboutus/quality/sei**cmm**.htm

CMM Level 5

- http://dqindia.ciol.com/content/advantage/103102703.asp
- **Why "India Inside" Spells Quality**
  Did you know that 75% of the world's CMM Level 5 software centers were in India? Here's how the quality movement transformed the Indian IT services industry

  Monday, October 27, 2003

  Europe, and the need for ISO certification, provided the trigger to the quality movement in India. But the real impetus came after Motorola's software center at Bangalore became the world's second CMM Level 5 unit in 1994 (the first was at NASA)
- Even for those familiar with India's software industry, this is a startling number.
- There are 80 software centers on the planet that are assessed at CMM Level 5.Of all those centers, 60 are in India.

Process Assessment

- Assessment is attempts to understand the current state of the software process with the intend of improving it
- SPICE (ISO/IE15504) defines a set of requirements for process assessment
  - Examines the processes used by organizations to determine whether they are effective in achieving their goals
  - Provides a reference model that examines the purpose and measurable objectives of the process (process dimension) and the set of process attributes that should be present (capability dimension)
  - SPICE model assessment is a structured evaluation of a process model (activities, tasks, work products, etc.)
- ISO 9001:2000 for Software defines requirements for a quality management system that will produce higher quality products and improve customer satisfaction
  - Stresses the importance for an organization to identify, implement, manage, and continually improve the effectiveness processes needed for a quality management system and to manage process interactions to achieve organization objectives
  - Process effectiveness and efficiency are assessed through internal or external review of processes using maturity scales
  - Results can be documented and monitored over time to reach improvement goals using "plan-do-check-act" cycle

** SEPA 6th ed, Roger S. Pressman*

Process Model

- Prescriptive Models
  - The Waterfall Model
  - Incremental Models
  - Evolutionary Process Models
- Specialized Process Models
- The Unified Process

** SEPA 6th ed, Roger S. Pressman*

Prescriptive Models

- Originally proposed to bring order to the chaos of software development
- Called "prescriptive" because:
  - prescribe a set of process elements (activities, actions, tasks, work products, quality assurance etc for each project)
  - each process model also prescribes a workflow
- They brought order to software engineering work and provide reasonable guidance to software teams
- Yet, they have not provided a definitive answer to the problems of software development in an ever changing computing environment

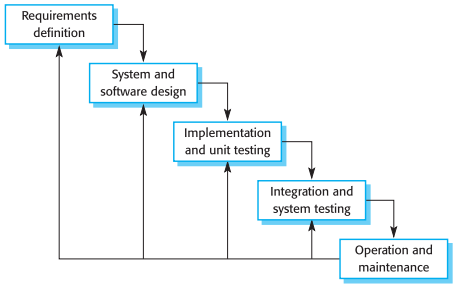
** SEPA 6th ed, Roger S. Pressman*

The Waterfall Model

- Waterfall Model
  - old fashioned but reasonable approach when requirements are well understood

** SEPA 6th ed, Roger S. Pressman*

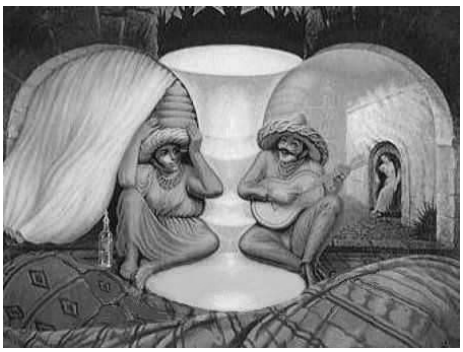
The Waterfall Model



The Waterfall Model Problem

- 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.

** Software Engineering 7th ed, Ian Sommerville*

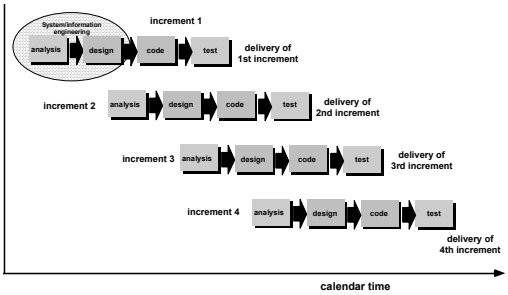


Incremental Models

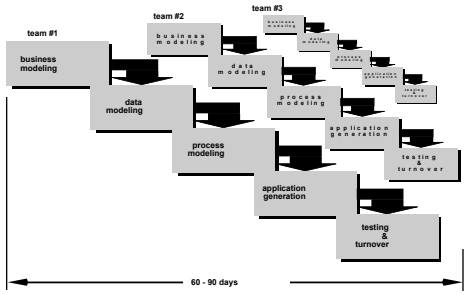
- Incremental Model
  - delivers software in small but usable pieces, each piece builds on pieces already delivered
- Rapid Application and Development (RAD) Model
  - makes heavy use of reusable software components with an extremely short development cycle

** SEPA 6th ed, Roger S. Pressman*

Incremental Model



RAD Model

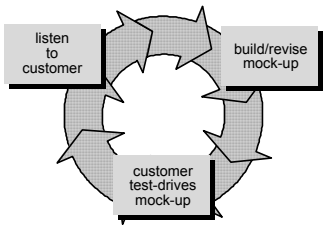


Evolutionary Process Models

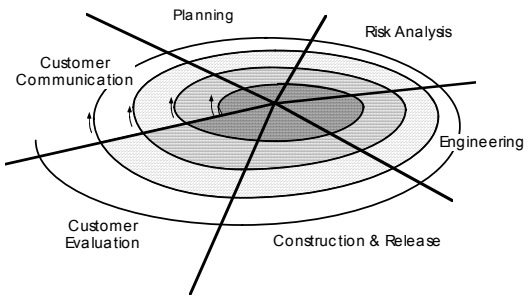
- Prototyping Model
  - good first step when customer has a legitimate need, but is clueless about the details, developer needs to resist pressure to extend a rough prototype into a production product
- Spiral Model
  - couples iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model
- Concurrent Development Model
  - similar to spiral model often used in development of client/server applications

** SEPA 6th ed, Roger S. Pressman*

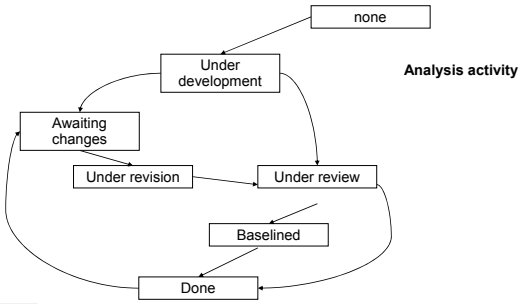
Prototyping Model



Spiral Model



Concurrent Development Model



Discussion

- Incremental Model Problem ?
- RAD Model Problem ?
- Prototyping Model Problem ?
- Spiral Model Problem ?
- Concurrent Development Problem ?

Discussion

- Incremental Model Applicability ?
- RAD Model Applicability ?
- Prototyping Model Applicability ?
- Spiral Model Applicability ?
- Concurrent Development Applicability ?

Specialized Process Models

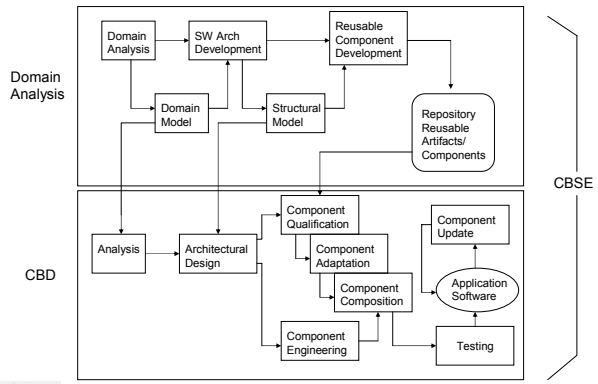
- Component-Based Development
  - spiral model variation in which applications are built from prepackaged software components called classes
- Formal Methods Model
  - rigorous mathematical notation used to specify, design, and verify computer-based systems
- Aspect-Oriented Programming
  - provides a process for defining, specifying, designing, and constructing software aspects like user interfaces, security, and memory management that impact many parts of the system being developed

** SEPA 6th ed, Roger S. Pressman*

CBSE

- 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.

** SEPA 6th ed, Roger S. Pressman*

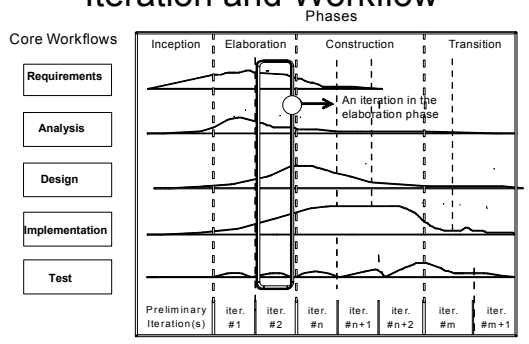


The Unified Process

- Use-case driven, architecture centric, iterative, and incremental software process
- Attempts to draw on best features of traditional software process models and implements many features of agile software development
- Phases
  - Inception phase (customer communication and planning)
  - Elaboration phase (communication and modeling)
  - Construction phase
  - Transition phase (customer delivery and feedback)
  - Production phase (software monitoring and support)

** SEPA 6th ed, Roger S. Pressman*

Iteration and Workflow



Unified Process Work Products

- Inception phase
  - Vision document
  - Initial use-case model
  - Initial project glossary
  - Initial business case
  - Initial risk assessment
  - Project plan (phases and iterations)
  - Business model
  - Prototypes
- Elaboration phase
  - Use-case model
  - Functional and non-functional requirements
  - Analysis model
  - Software architecture description
  - Executable architectural prototype
  - Preliminary design model
  - Revise risk list
  - Project plan (iteration plan, workflow, milestones)
  - Preliminary user manual
- Construction phase
  - Design model
  - Software components
  - Integrated software increment
  - Test plan
  - Test cases
  - Support documentation (user, installation, increment)
- Transition phase
  - Delivered software increment
  - Beta test reports
  - User feedback

** SEPA 6th ed, Roger S. Pressman*

An Agile View of Process

- Represents a reasonable compromise between conventional software engineering for certain classes of software and certain types of software projects
- Can deliver successful systems quickly
- Stresses continuous communication and collaboration among developers and customers
- Embraces a philosophy that encourages:
  - customer satisfaction,
  - incremental software delivery,
  - small project teams (composed of software engineers and stakeholders),
  - informal methods, and
  - minimal software engineering work products
- Stress on-time delivery of an operational software increment over analysis and design

** SEPA 6th ed, Roger S. Pressman*

Manifesto for Agile Software Development

- Proposes that it may be better to value:
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
- While the items on the right are still important the items on the left are more valuable under this philosophy
- Note: although most practitioners agree with this philosophy in theory, many pragmatic issues surface in the real world that may cause items on the right to be as important as items on the left

** SEPA 6th ed, Roger S. Pressman*

Agile Process Models

- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Dynamic Systems Development Method (DSDM)
- Scrum
- Crystal
- Feature Driven Development (FDD)
- Agile Modeling (AM)

** SEPA 6th ed, Roger S. Pressman*