

Unit-1

Software Process

Chamundeswari Arumugam
Professor
SSN College of Engineering, Chennai

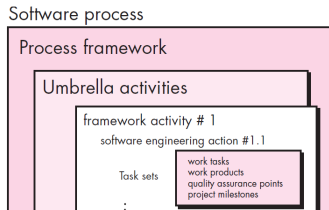
December 2017

- Software process
- Perspective process
- Specialized process

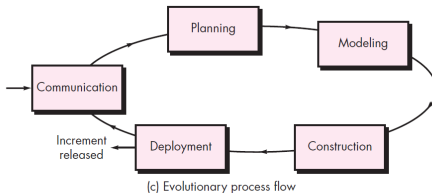
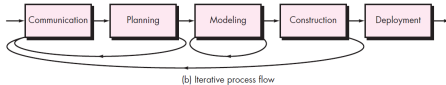
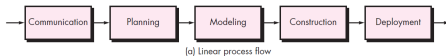
Software process

- Software process - Collection of work activities, actions, and tasks to create work products.
- Framework - Activities, actions, and tasks reside with relationship.
- Framework activities - communication, planning, modeling, construction, and deployment
- Umbrella activities - project tracking and control, risk management, quality assurance, etc.

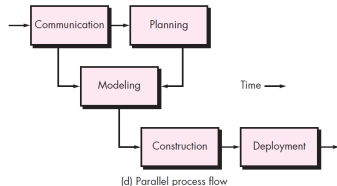
Figure: Framework



Software process(contd..)



- 1 Process flow activities are organized with respect to sequence and time.
- 2 Process flow execution can be organized as linear / iterative / evolutionary / parallel.



Software process(contd..)

Software process task

- task sets - a collection of software engineering work tasks, related work products, quality assurance points, and project milestones. Eg. Elicitation

Process assessment

- CMMI Assessment Method
- SPICE (ISO/IEC15504)
- ISO 9001:2000 for Software

Software process pattern

- Process pattern - process-related problem, identifies the environment, proven solutions to the problem.

Pattern example

- **Pattern Name** - given name of the pattern, eg. technical review
- **Forces** - identifies the environment, eg. Effort to complete the technical review.
- **Types**
 - 1 Stage pattern (defines a problem associated with a framework activity) eg. Review Co-ordination,
 - 2 Task pattern (defines a problem associated with a software engineering) eg. Review on Tasks
 - 3 Phase pattern (define the sequence of framework activities) eg.iterative review

Software process(contd..)

Pattern example(contd..)

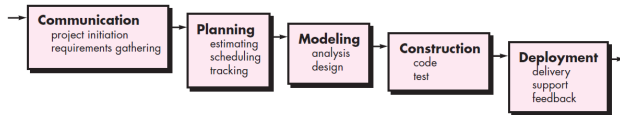
- **Initial context** - Describes the conditions under which the pattern applies. eg. Communication executed between technical review team and project team.
- **Problem** - specific problem eg. To complete technical review satisfying the norms of the organization.
- **Solution** - Describes how to implement the pattern
- **Resulting context** - Describes the conditions that will result once the pattern has been successfully implemented. Eg. Exit criteria for review team.
- **Related pattern** - Provide a list of all process patterns that are directly related to this one. Eg. ProjectDeliverables, ProjectAssessment, ReviewMeasures, etc.
- **Known uses and examples** - Indicate the specific instances in which the pattern is applicable. Eg. Communication

Perspective process model

- Prescriptive process models were originally proposed to bring order to the chaos of software development.
- Each software process models has generic framework activities but a order of activities, actions and task differ.
 - ① Waterfall Model
 - ② Incremental Process Models
 - ③ Evolutionary Process Models prototyping, spiral.
 - ④ Concurrent model

Perspective process model- Waterfall process model

- The waterfall model, the classic life cycle- a systematic, sequential approach to software development.
- Progresses through planning, modeling, construction, and deployment.
- Workflow from **communication** through **deployment**.



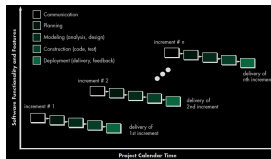
- When to use this model ?
 - 1 software requirements are reasonably well defined,
 - 2 work is to proceed to completion in a linear manner.

Problem

- Sequential flow, Natural uncertainty, Late working version, Delay in Dependency task, Exceed production time.

Perspective process model - Incremental process model

- Combines activity of linear and parallel process flows.
- The first increment is often a core product.
- Supplementary features are added up.
- Focuses on the delivery of an operational product with each increment.



- When to use this model ?
 - 1 Limited set of software functionality.
 - 2 Refine and expand on that functionality in later software releases.

Advantage

- When staffing is unavailable

Perspective process model- Evolutionary process model

- Process model that has been explicitly designed to accommodate a product that evolves over time.
- Evolutionary models are iterative
- Two evolutionary model
 - 1 Prototyping
 - 2 Spiral

Perspective process model- Evolutionary prototyping process model

- The prototype can serve as the first system and mechanism for identifying software requirements.
- Quick design focuses on aspects visible to end users. Quick design leads to the construction of a prototype.
- Working programs generated quickly
- The prototype is deployed and evaluated by stakeholders,



- **Best approach. when?**
 - 1 Developer may be unsure of the efficiency of an algorithm
 - 2 Adaptability of an operating system
 - 3 Requirements are fuzzy.

Problem

- Developed haphazardly. No actual thought on implementation

Perspective process model- Evolutionary spiral process model

- Adapted to apply throughout the life of the software.
- Combines linear and iterative process model
- Software is developed in a series of evolutionary releases and will evolve through a number of iterations around the spiral.
- Risk is considered as each revolution is made.



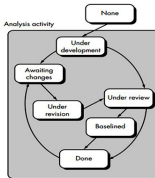
- **Best approach. when?**
 - 1 When a change is initiated, the process starts at the appropriate entry point.
 - 2 Realistic approach to the development of large-scale systems and software


Problem

- Difficulty in risk assessment

Perspective process model- Concurrent process model

- Applicable to all types of software development and provides an accurate picture of the current state of a project.
- Defines a series of events that will trigger transitions from state to state for each of the software engineering activities, actions, or tasks.
- Independently built independent pieces, improved resource utilization, early finish



 Represents a state of a software engineered activity

• Example states

- 1 Communication activity - done state into the awaiting changes state,
- 2 Modeling - inactive state to under development state

Problem

- Complex resource allocation, few risk emerge after integration

Specialized process model

- For narrowly defined software engineering approach a specialized process model can be applied.
 - ① Component-Based Development
 - ② Formal Methods Model
 - ③ Aspect-Oriented Software Development

Perspective process model- Component-Based Development process model

- Commercial off-the-shelf (COTS) software components
- Works towards targeted functionality
- Supports well defined interfaces
- Many components can be integrated into the software
- Modeling and construction activities begin with the identification of candidate components.
- **Best approach. when?**
 - 1 Reusability is high instead of developing from beginning
 - 2 Reduction in development cycle time and cost

Problem

- Modification is problematic

Perspective process model- Formal methods process model

- Formal mathematical specification
- Specify, develop, and verify using mathematical notation
- Cleanroom software engineering
- Application of mathematical analysis
- Program verification - discover and correct errors
- Best approach. when?
 - 1 To develop defect free software
 - 2 Best to apply in safety critical software

Problem

- time consuming, expensive, involve extensive training

- Software process and models used in SE
- Identify the activities in software process model / framework.
- List the different execution sequence of the activities in a process.
- Identify a few goals in assessment models.

- List out an activity, actions and tasks in SE.
- List the perspective software process model and its applicability in projects.
- List the specialized software process model and its applicability in projects.