



UGANDA CHRISTIAN UNIVERSITY

A Centre of Excellence in the Heart of Africa

NAME: GALIWANGO CHARLES JOVANS

REGISTRATION NUMBER: S19B23/119

COURSE: BSCS THREE

COURSE UNIT: SOFTWARE CONSTRUCTION

LECTURER: Mr. SIMON LUBAMBO

CHAPTER THREE: SOFTWARE PROCESS STRUCTURE

A **software process** as a framework for the activities, actions, and tasks that are required to build high-quality software.

Howard Baetjer Jr. [Bae98] comments on the software process saying software development is a social learning process

The process is a dialogue in which the knowledge that must become the software is brought together and embodied in the software.

The process provides interaction between users and designers, between users and evolving tools, and between designers and evolving technology.

Difference between Software process and software Engineering

Software process defines the approach that is taken as software is engineered whereas.

Software engineering also encompasses technologies that populate the process technical methods and automated tools.

Generic Process Model

The software process is represented schematically in the **Figure 1** below. Referring to the figure, each framework activity is populated by a set of software engineering actions.

Each software engineering action is defined by a task set that identifies the work tasks that are to be completed, the work products that will be produced, the quality assurance points that will be required, and the milestones that will be used to indicate progress.

A generic process framework for software engineering defines five framework activities *communication, planning, modelling, construction, and deployment*. In addition, a set of umbrella activities project tracking and control, risk management, quality assurance, configuration management, technical reviews, and others are applied throughout the process.

Process flow describes how the framework activities and the actions and tasks that occur within each framework activity are organized with respect to sequence and time.

Below are some of the types of Process flows.

- A linear process flow executes each of the five framework activities in sequence, beginning with communication and culminating with deployment.
- An iterative process flow repeats one or more of the activities before proceeding to the next
- An evolutionary process flow executes the activities in a “circular” manner. Each circuit through the five activities leads to a more complete version of the software.
- A parallel process flow executes one or more activities in parallel with other activities.

The figures to the process flows are seen below in *figures 3.2*.

Figure 1

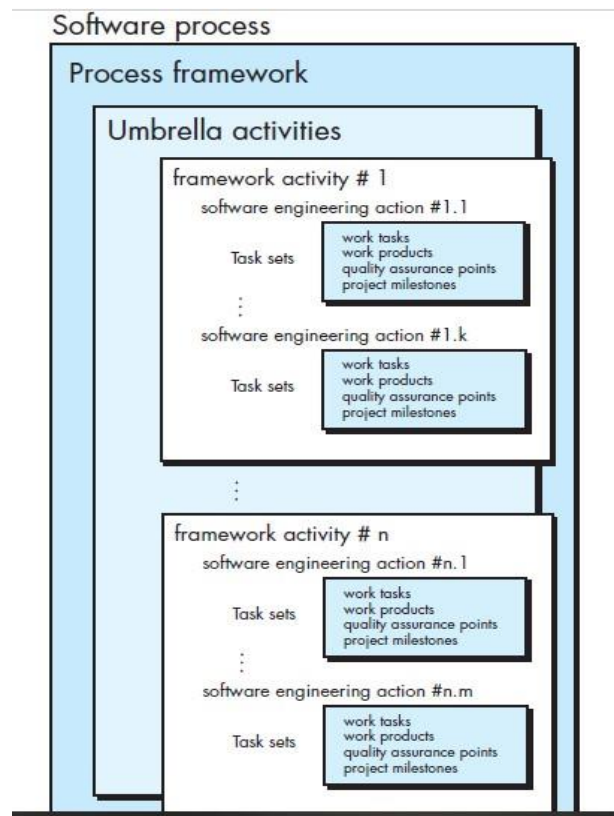
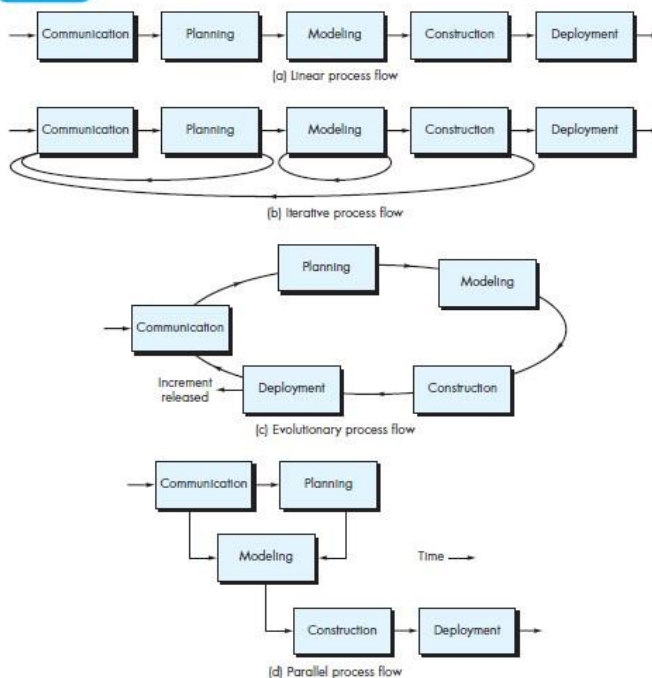


Figure 3.2 Process flow



Task Set: A task set defines the actual work to be done to accomplish the objectives of a software engineering action. The goal of requirements gathering is to understand what various stakeholders want from the software that is to be built.

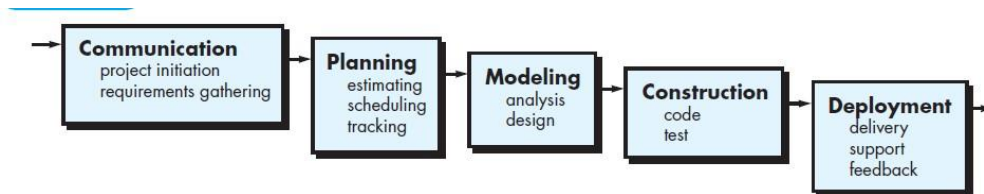
A few different approaches to software process assessment and improvement have been proposed over the past few decades:

- Standard CMMI Assessment Method for Process Improvement (SCAMPI)
- CMM-Based Appraisal for Internal Process Improvement (CBA IPI)
- SPICE (ISO/IEC15504 ISO 9001:2000 for Software)

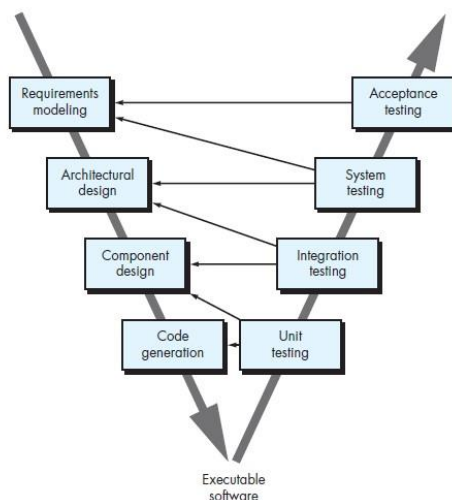
CHAPTER FOUR: PROCESS MODELS

process models were originally proposed to bring order to the chaos of software development.

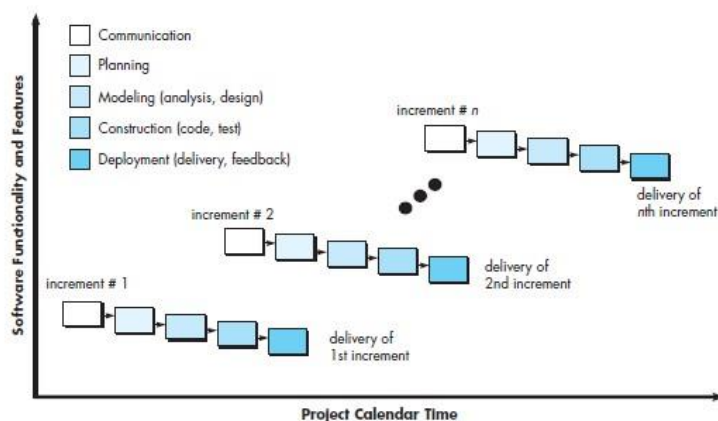
The Waterfall Model



V process model



Incremental Process



Evolutionary Process Models

Prototyping. Often, a customer defines a set of general objectives for software, but does not identify detailed requirements for functions and features

The Spiral Model. is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model.

Specialized process Models

Specialized process models take on many of the characteristics of one or more of the traditional models presented in the preceding sections. However, these models tend to be applied when a specialized or narrowly defined software engineering approach is chosen.

Chapter Five: AGILE DEVELOPMENT

Agility

Agility means effective (rapid and adaptive) response to change, effective communication among all stockholder.

Agility principles

- Early and continuous delivery of valuable software
- Embrace change.
- Frequent delivery
- Cooperation
- Autonomy and motivation
- Better communication
- Working software
- Stable work environments
- Quality assurance
- Simplicity

Extreme programming

Extreme programming (XP) is one of the most important software development frameworks of Agile models. It is used to improve software quality and responsiveness to customer requirements.

The extreme programming model recommends taking the best practices that have worked well in the past in program development projects to extreme levels for example.

- Code Review
- Testing
- Incremental development
- Simplicity
- Design
- Integration testing
- Basic principles of Extreme programming

Extreme programming (XP) process

XP describes four basic activities that are performed within the software development process. Listed below are the processes it goes through.

- Coding
- Testing
- Listening
- Designing

Other agile process models

- Adaptive Software Development (ASD)
- Dynamic Systems Development Method (DSDM)
- Scrum
- Crystal
- Feature Driven Development (FDD)

Chapter Six: Human Aspects of software engineering

Characteristics of a software Engineer

- Individual responsibility
- Acute awareness
- Brutally honest
- Resilience under pressure
- Heightened sense of fairness
- attention to detail
- pragmatic

Layered behavioural of SD

- at an individual level
- At team and project levels
- at outer layers
- at team levels

Effective software team must establish.

- sense of purpose
- sense of involvement
- sense of trust
- sense of improvement

Factors that foster a potentially toxic team environment

- A chaotic work atmosphere
- high frustration that causes friction among team members
- A fragmented or poorly coordinated software process
- An unclear definition of roles on the software team
- Continuous and repeated exposures to failure

Teams In Software Development

- Agile Teams
Agile teams are self organizing uses elements of Constantine's random open and synchronous paradigms
- XP teams
- Global Teams

Impact of social media

Most social media enable the formation of "communities" of users with similar interests. For example, a community of software engineers who specialize in real-time embedded systems might provide a useful way for an individual or

- Microblogs (e.g., Twitter) allow a member of a software engineering network to post short messages to followers who subscribe to them.
- Targeted on-line forums allow participants to post questions, opinions, case studies or any other relevant information.
- Social networking sites (e.g., Facebook, LinkedIn) allow degrees-of-separation connections among software developers and related technologists.
- social bookmarking sites (e.g., Delicious, Stumble, CiteULike) allow a software engineer or team to recommend Web-based resources that may be of interest to a social media community of like-minded individuals.

Software engineering using the cloud.

Cloud computing provides a mechanism for access to all software engineering work products, artifacts, and project-related information.

It runs everywhere and removes the device dependency that was once a constraint for many software projects.

It allows members of a software team to conduct platform-independent, low-risk trials of new software tools and to provide feedback on those tools.