UGANDA CHRISTIAN UNIVERSITY

A Centre of Excellence in the Heart of Africa

Name	:	KAUTA MARVIN
Registration Number	:	S20B23/204
Faculty/School	:	SCIENCE AND TECHNOLOGY
Course	:	BACHOLERS OF SCIENCE IN COMPUTER SCIENCE
Lecturer	:	***MR LUBAMBO SIMON***

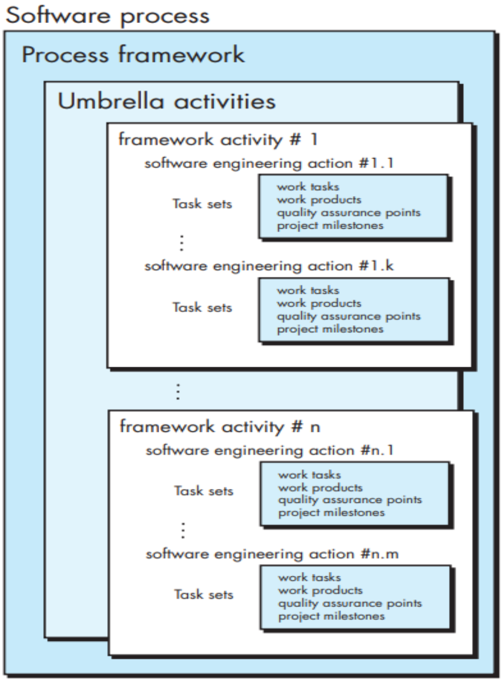
SOFTWARE PROCESS STRUCTURE

A process was defi ned as a collection of work activities, actions, and tasks that are performed when some work product is to be created

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 tools [technology]

The hierarchy of technical work within the software process is activities, encompassing actions, populated by tasks.

A GENERIC PROCESS MODEL



Each framework activity is populated by a set of software engineering actions.

Each software engineering action is defi ned 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
- deployment

In addition, a set of umbrella activities

- project tracking and control
- risk management
- quality assurance
- configuration management

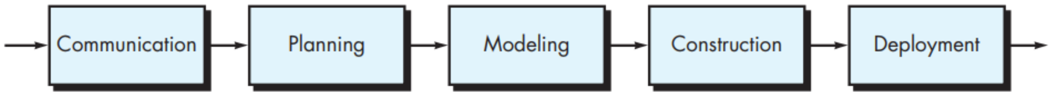
- technical reviews and others—are applied throughout the process

PROCESS FLOW

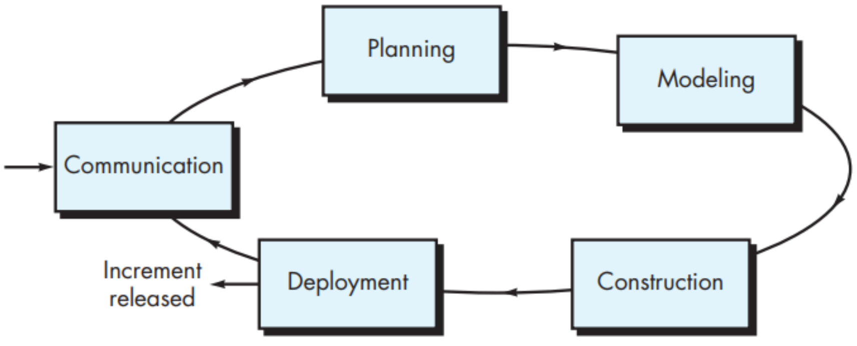
A process flow— describes how the framework activities and the actions and tasks that occur within each framework activity are organized with respect to sequence.

A linear process flow - executes each of the five framework activities in sequence, beginning with communication and culminating with deployment.

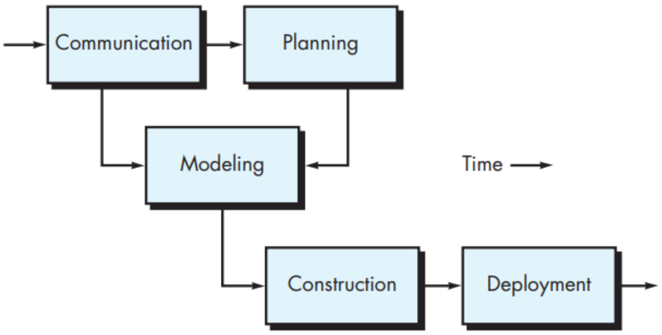
FIGURE 3.2 Process flow



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 (e.g., modelling for one aspect of the software might be executed in parallel with construction of another aspect of the software



IDENTIFYING A TASK SET

Different projects demand different task sets. The software team chooses the task set based on problem and project characteristics.

A task set defines the actual work to be done to accomplish the objectives of a software engineering action.

For a small, relatively simple project, the task set for requirements gathering might look like this:

- Make a list of stakeholders for the project.
- Invite all stakeholders to an informal meeting.
- Ask each stakeholder to make a list of features and functions required.
- Discuss requirements and build a final list.
- Prioritize requirements.
- Note areas of uncertainty.

PROCESS PATTERNS

A process pattern describes a process-related problem that is encountered during software engineering work, identifies the environment in which the problem has been encountered, and suggests one or more proven solutions to the problem.

A process pattern provides you with a template, a consistent method for describing problem solutions within the context of the software process. By combining patterns, a software team can solve problems and construct a process that best meets the needs of a project.

Ambler [Amb98] has proposed a template for describing a process pattern:

Pattern Name. a meaningful name describing it within the context of the software

Forces. The environment in which the pattern is encountered and the issues that make the problem visible and may affect its solution.

Type. The pattern type is specified

1. **Stage pattern**—defines a problem associated with a framework activity for the process.
2. **Task pattern**—defines a problem associated with a software engineering action or work task and relevant to successful software engineering practice .
3. **Phase pattern**—define the sequence of framework activities that occurs within the process, even when the overall flow of activities is iterative in nature.

Initial Context. Describes the conditions under which the pattern applies. Prior to the initiation of the pattern.

Resulting Context. Describes the conditions that will result once the pattern has been successfully implemented.

Related Patterns. Provide a list of all process patterns that are directly related to this one.

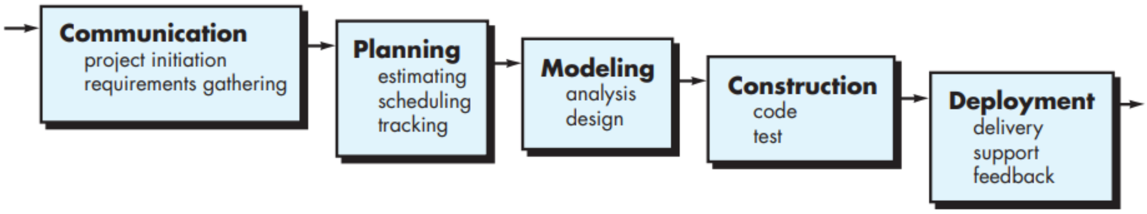
Known Uses and Examples. Indicate the specific instances in which the pattern is applicable.

PROCESS MODELS

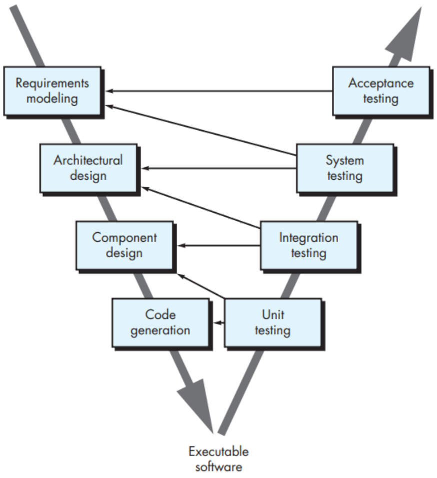
A process model provides a specific roadmap for software engineering work. It defines the flow of all activities, actions and tasks, the degree of iteration, the work products, and the organization of the work that must be done.

A prescriptive process model1 strives for structure and order in software development.

The Waterfall Model

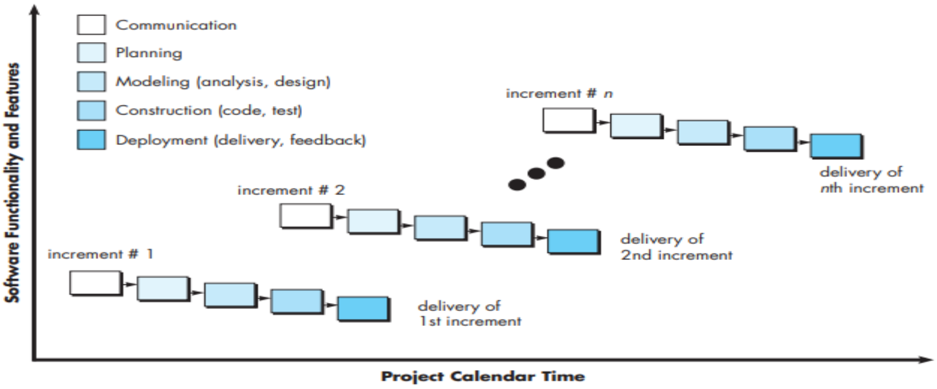


The waterfall model, sometimes called the classic life cycle, suggests a systematic, sequential approach to software development that begins with customer specification of requirements and progresses through planning, modelling, construction, and deployment, culminating in ongoing support of the completed software. A variation in the representation of the waterfall model is called the V-model.



INCREMENTAL PROCESS MODELS

The incremental model delivers a series of releases, called increments, that provide progrea22ssively more functionality for the customer as each increment is delivered.



The incremental model combines the elements’ linear and parallel process flows.