The Product and the Process

Chapter 1-3 (Pressman 8E)

CSCI-P465/565

Kurt Seiffert

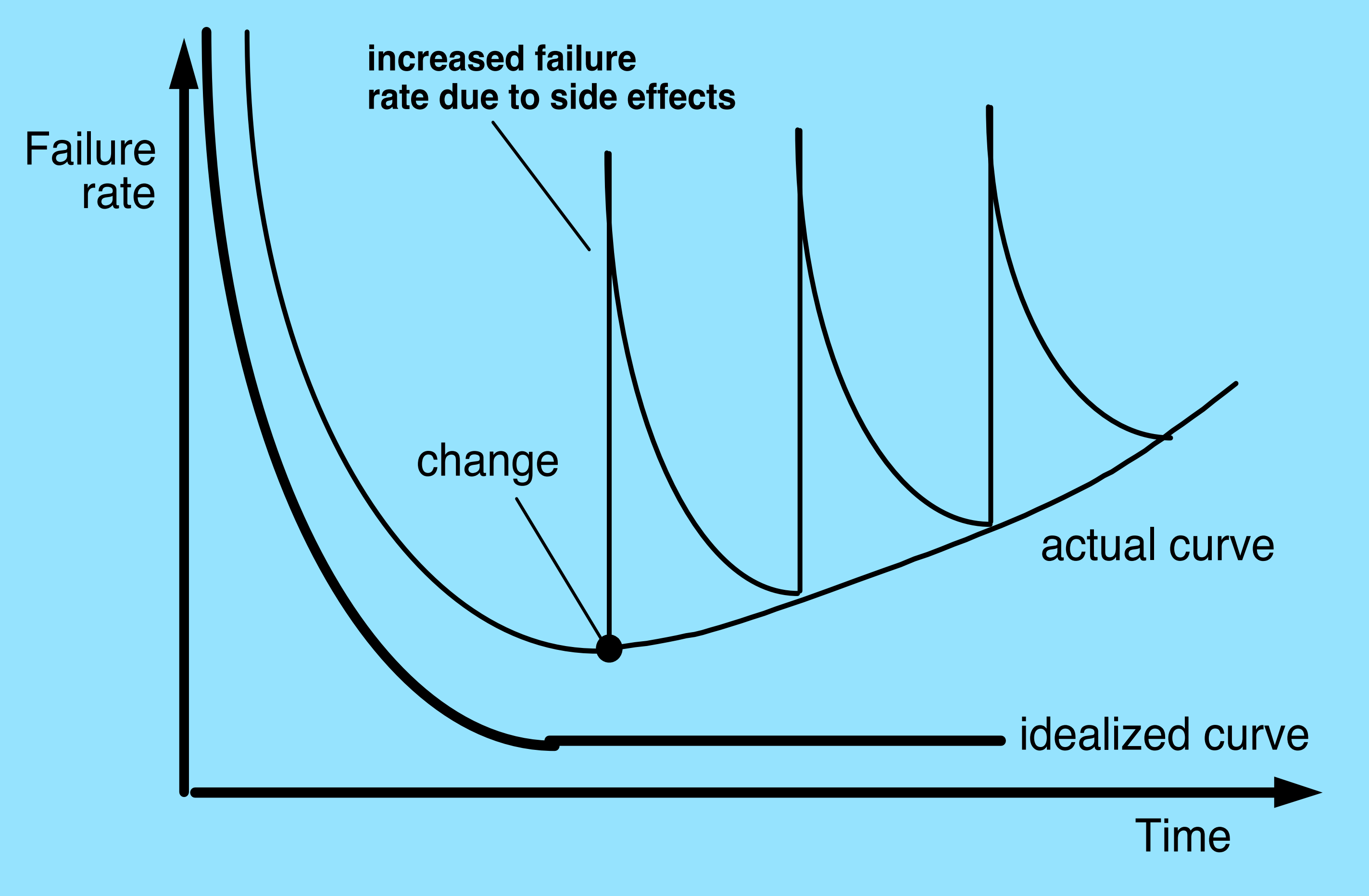
What is Software?

Software is: *(1) instructions (computer programs) that when executed provide desired features, function, and performance; (2) data structures that enable the programs to adequately manipulate information and (3) documentation that describes the operation and use of the programs.*

What is Software?

- Software is developed or engineered, it is not manufactured in the classical sense.
- Software doesn't "wear out"
- Although the industry is moving toward component-based construction, most software continues to be custom-built.

Wear vs. Deterioration



Software Applications

- System software (VOIP, OS, networking)
- Application software (standalone apps)
- Engineering/Scientific software (genetics, bio)
- Embedded software
- Product-line software (mass production)
- Web/Mobile applications
- AI software (robotics, neural nets, game playing)

Legacy Software

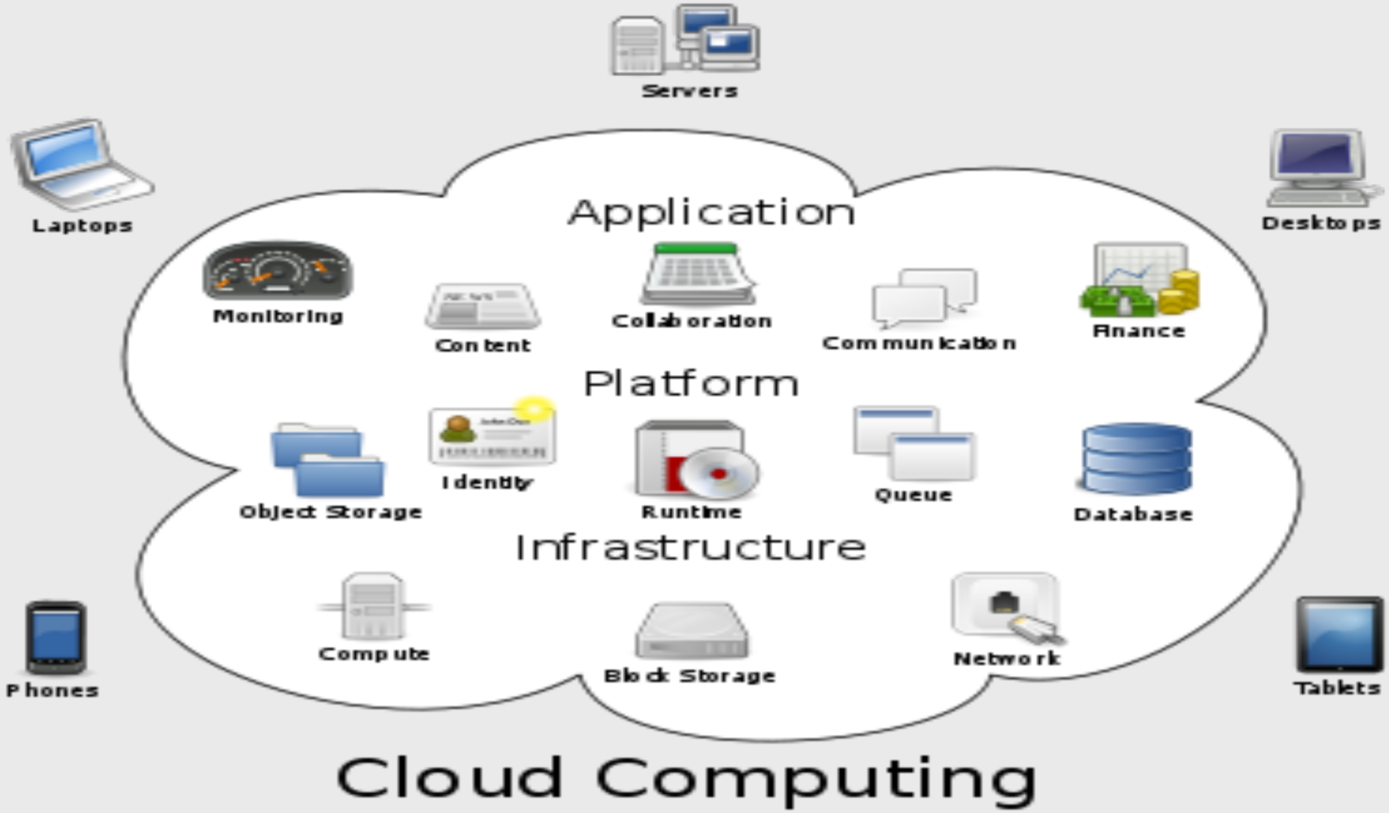
Why must it change?

- software must be adapted to meet the needs of new computing environments or technology.
- software must be enhanced to implement new business requirements.
- software must be extended to make it interoperable with other more modern systems or databases.
- software must be re-architected to make it viable within a network environment.

WebApps & Mobile Apps

- Modern WebApps are Data driven, Content sensitive, Continuous evolution, Immediacy, Security, Aesthetics
  - Software reuse is important
  - Mostly developed incrementally (Agile)
- Mobile Apps reside on mobile platforms such as cell phones or tablets, have user interfaces that take both device characteristics and location attributes
- Often provide access to a combination of web-based resources and local device processing and storage capabilities

Cloud Computing



Cloud Computing

- *Cloud computing* provides distributed data storage and processing resources to networked computing devices
- Computing resources reside outside the cloud and have access to a variety of resources inside the cloud
- Cloud computing requires developing an architecture containing both frontend and backend services
- SaaS, PaaS, IaaS

Product Line Software

- *Product line software* is a set of software-intensive systems that *share a common set of features* and satisfy the needs of a particular market
- These software products are developed using the same application and data architectures using a common core of reusable software components
- A software product line allow in the development of many products that are engineered by capitalizing on the commonality among all products within the product line

Software Engineering

- Some realities:
  - *a concerted effort should be made to understand the problem before a software solution is developed*
  - *design becomes a pivotal activity*
  - *software should exhibit high quality*
  - *software should be maintainable*

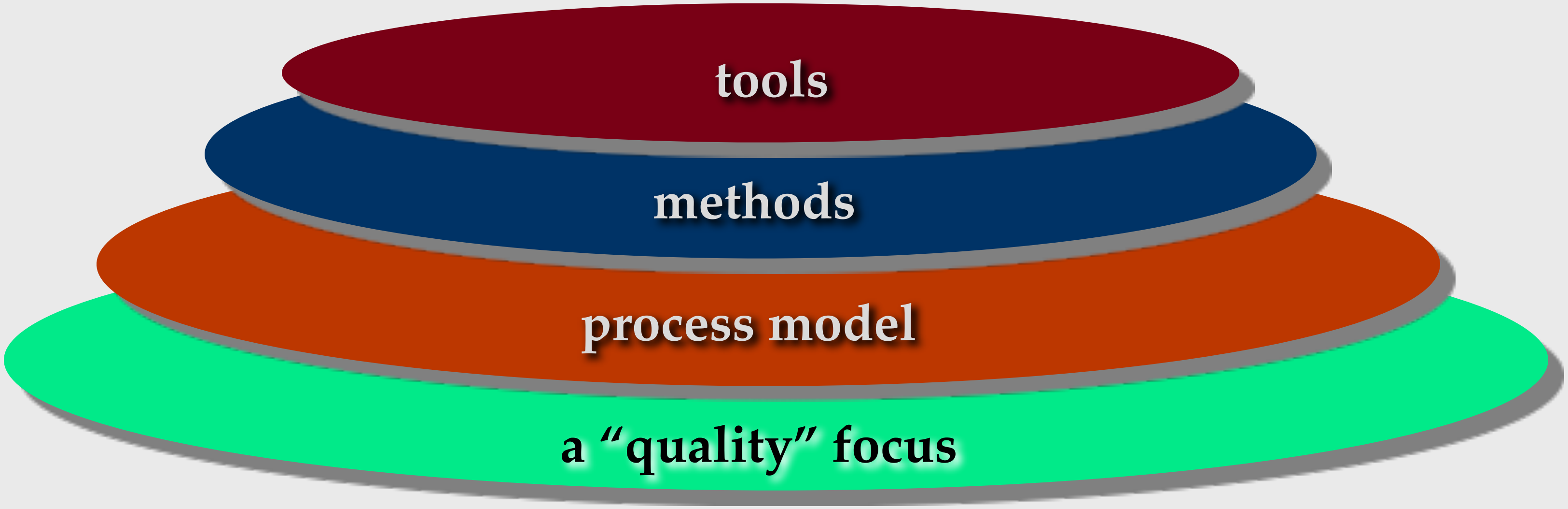
Software Engineering

- The seminal definition:
  - *[Software engineering is] the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.*

Software Engineering

- The IEEE definition:
    - *Software Engineering:*
    - *(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*
    - *(2) The study of approaches as in (1).*

A Layered Technology



Software Engineering

A Process Framework

Process framework

Framework activities

work tasks

work products

milestones & deliverables

QA checkpoints

Umbrella Activities

Framework Activities

- Communication
- Planning
- Modeling
  - Analysis of requirements
  - Design
- Construction
  - Code generation
  - Testing
- Deployment

- Software Cost:
  - 60% development, 40% Testing

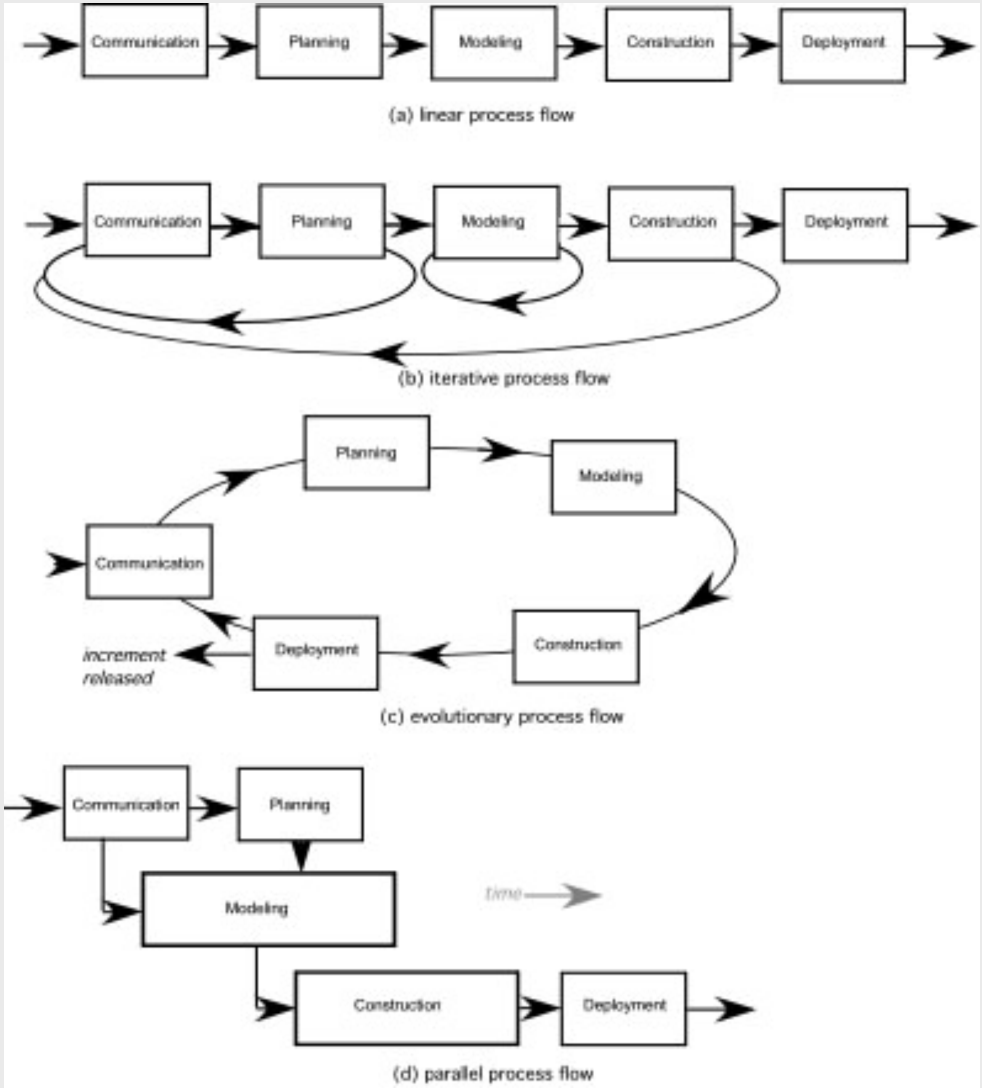
Umbrella Activities

- Software project tracking and control
- Risk management
- Software quality assurance
- Technical reviews
- Measurement
- Software configuration management
- Reusability management
- Work product preparation and production

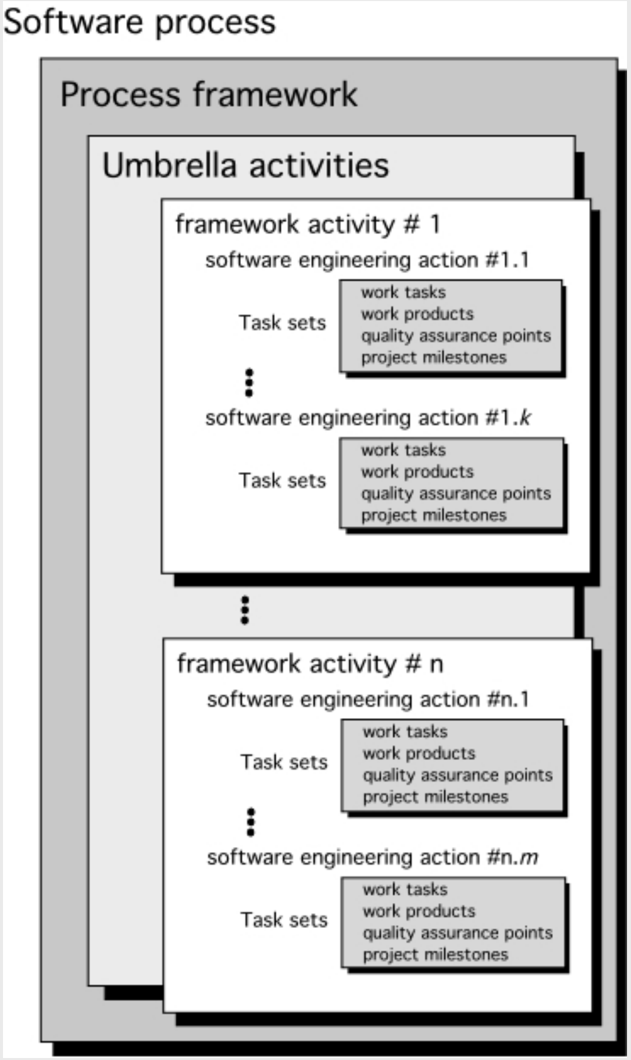
Adapting a Process Model

- the overall flow of activities, actions, and tasks and the interdependencies among them
- the degree to which actions, tasks, products, process are defined
- the manner which other activities are applied
- the level of autonomy given to the software team
- the degree to which team organization and roles are prescribed

Process Flow



A Generic Process Model



Software process

Process framework

Umbrella activities

framework activity # 1

software engineering action #1.1

Task sets

work tasks
work products
quality assurance points
project milestones

software engineering action #1.*k*

Task sets

work tasks
work products
quality assurance points
project milestones

framework activity # n

software engineering action #n.1

Task sets

work tasks
work products
quality assurance points
project milestones

software engineering action #n.*m*

Task sets

work tasks
work products
quality assurance points
project milestones

Identifying a Task Set

- A task set defines the actual work to be done to accomplish the objectives of a software engineering action.
    - A list of the task to be accomplished
    - A list of the work products to be produced
    - A list of the quality assurance filters to be applied

The Essence of Practice

- Polya suggests:

1. *Understand the problem* (communication and analysis).
2. *Plan a solution* (modeling and software design).
3. *Carry out the plan* (code generation).
4. *Examine the result for accuracy* (testing and quality assurance).

Hooker’s General Principles

- 1: *The Reason It All Exists*
- 2: *KISS (Keep It Simple, Stupid!)*
- 3: *Maintain the Vision*
- 4: *What You Produce, Others Will Consume*
- 5: *Be Open to the Future*
- 6: *Plan Ahead for Reuse*
- 7: *Think!*

Software Myths

- Affect managers, customers (and other non-technical stakeholders) and practitioners
- Are believable because they often have elements of truth,

but …

- Invariably lead to bad decisions,

therefore …

- Insist on reality as you navigate your way through software engineering

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.
- Stated in more general terms, a process pattern provides you with a *template* [Amb98]—a consistent method for describing problem solutions within the context of the software process.