Chapter 1

- **Software**

Slide Set to accompany

Software Engineering: A Practitioner’s Approach, 7/e

by Roger S. Pressman

For non-profit educational use only

Table of Contents

1.1 The Nature of Software

1.2 The Changing Nature of Software

1. The Nature of Software

1.1 What is Software?

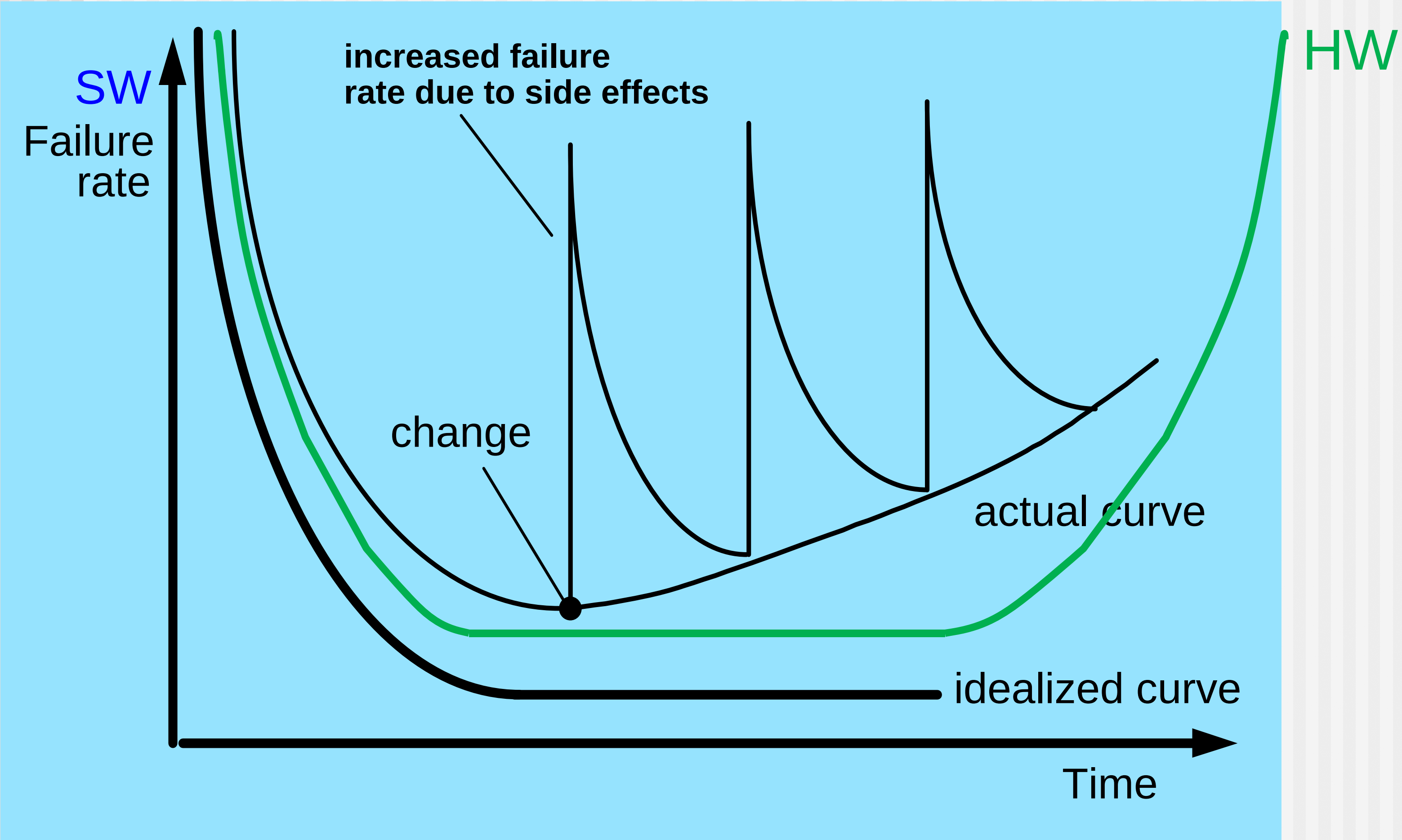
Definition Software is:

(1) computer programs that when executed provide desired features, function, and performance; together with data structures that enable the programs to adequately manipulate information

(2) documentation that describes the operation and use of the programs.

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

Software doesn't "wear out."



These slides are designed to accompany *Software Engineering: A Practitioner’s Approach, 7/e* (McGraw-Hill 2009). Slides copyright 2009 by Roger Pressman.

1.1.2 Software Applications

- Categories
  - system software
  - application software
  - engineering/scientific software
  - embedded software
  - product-line software
  - WebApps (Web applications)
  - AI software

WebApps

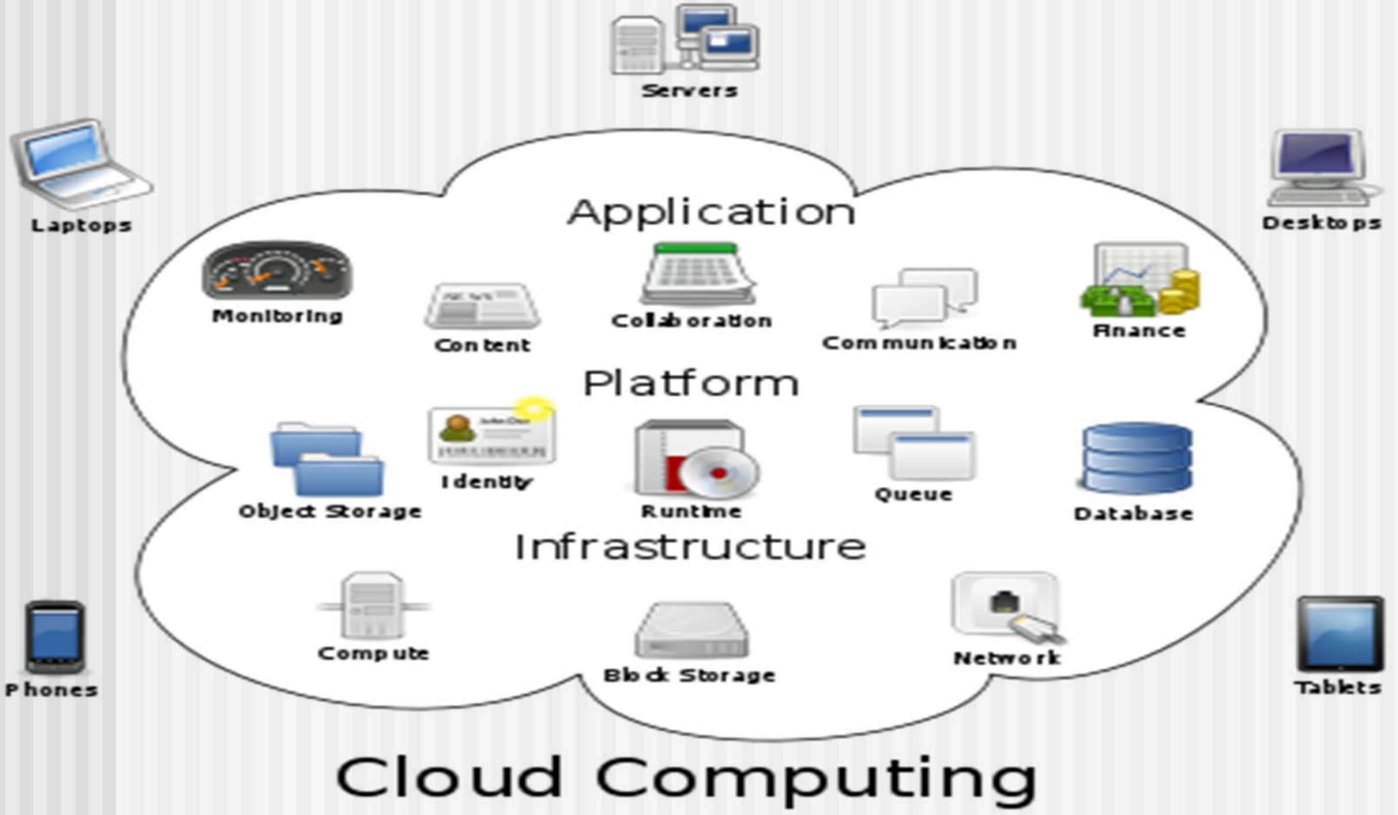
- Characteristics
  - **Network intensiveness.** A WebApp resides on a network and must serve the needs of a diverse community of clients.
  - **Concurrency.** A large number of users may access the WebApp at one time.
  - **Unpredictable load.** The number of users of the WebApp may vary by orders of magnitude from day to day.
  - **Performance.** If a WebApp user must wait too long (for access, for server-side processing, for client-side formatting and display), he or she may decide to go elsewhere.

Mobile Apps

- Reside on mobile platforms such as cell phones or tablets
- Contain 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
- Provide persistent storage capabilities within the platform

- A *mobile web application* allows a mobile device to access to web-based content using a browser designed to accommodate the strengths and weaknesses of the mobile platform
- A *mobile app* can gain direct access to the hardware found on the device to provide local processing and storage capabilities
- As time passes these differences will become blurred

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
- Frontend services include the client devices and application software to allow access
- Backend services include servers, data storage, and server-resident applications
- Cloud architectures can be segmented to restrict access to private data

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 shares a set of assets that include *requirements, architecture, design patterns, reusable components, test cases,* and other work products
- A software product line allow in the development of many products that are engineered by capitalizing on the commonality among all products with in the product line

1.1.3 Legacy Software

Legacy Software should evolve

Why must it change?

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

1.2 The Changing Nature of Software

Example) WebApps

- **Continuous evolution.** Unlike conventional application software that evolves over a series of planned, chronologically-spaced releases, Web applications evolve continuously.

"By the time we see any sort of stabilization the Web will have turned into something completely different." Louis Monier