1. Introduction to Software Engineering

What are the expectations?

- It should meet the needs of the people who use it
- It should perform flawlessly over a long period of time
- It should be easy to modify
- It should be easy to use

What things are not expected?

- When its users are dissatisfied
- When it is error prone
- When it is difficult to change
- When is it harder to use

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.

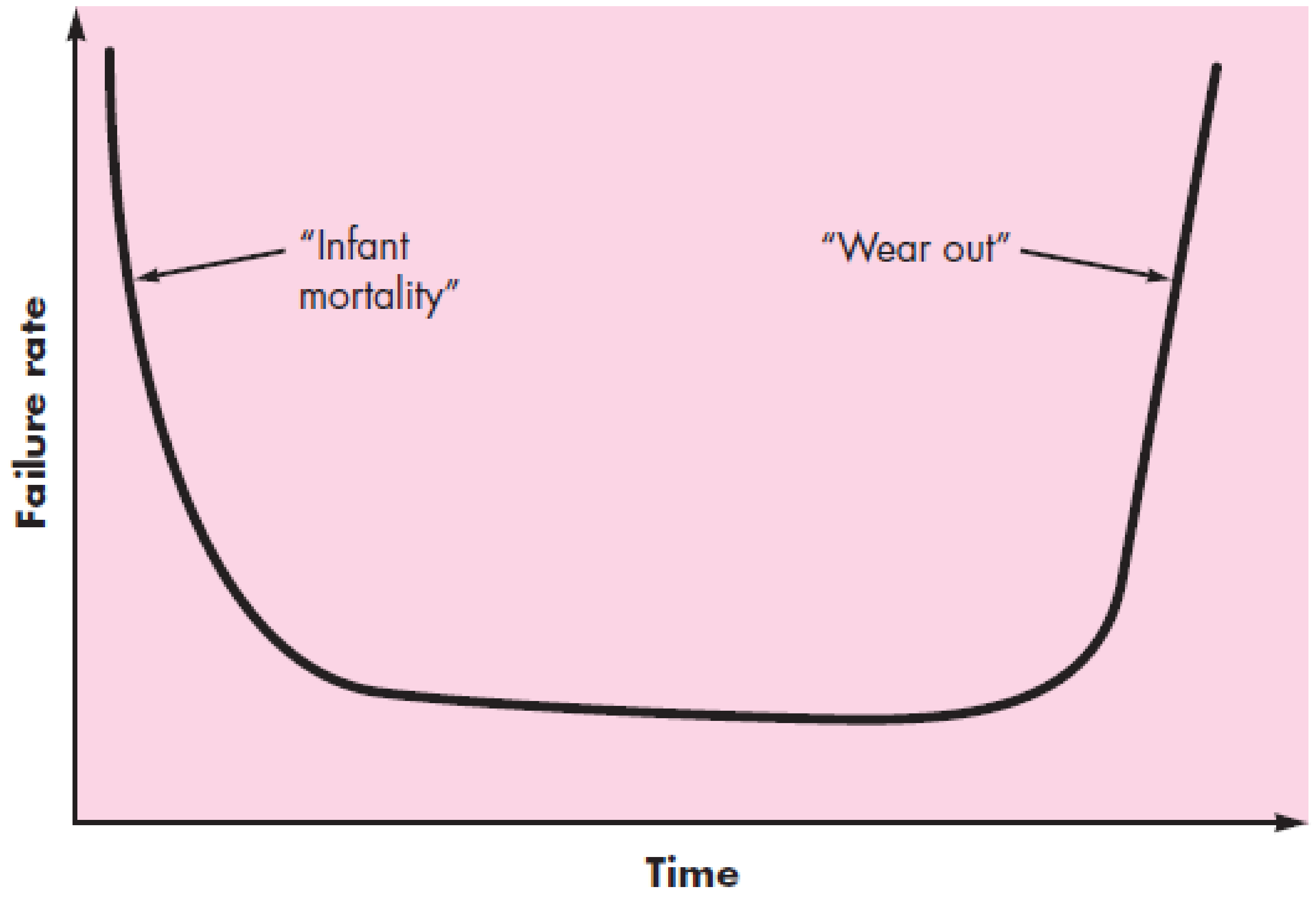
What is Software?

- Software is developed or engineered, it is not manufactured in the classical sense.
  - Common goal in both is to achieve high quality
    - But manufacturing process of hardware can have quality problems which are non existent in software (easily corrected)
  - Both activities depend on people
    - but the relationship between people applied and work accomplished is entirely different
  - Both construct a product
    - But the approaches are different

What is Software?

- *Software doesn't "wear out."*
  - hardware exhibits relatively high failure rates early in its life; defects are corrected and the failure rate drops to a steady-state level for some period of time.
  - As time passes, however, the failure rate rises again as hardware components suffer from the cumulative effects of dust, vibration, abuse, temperature extremes, and many other environmental maladies.
  - Stated simply, the hardware begins to *wear out.*

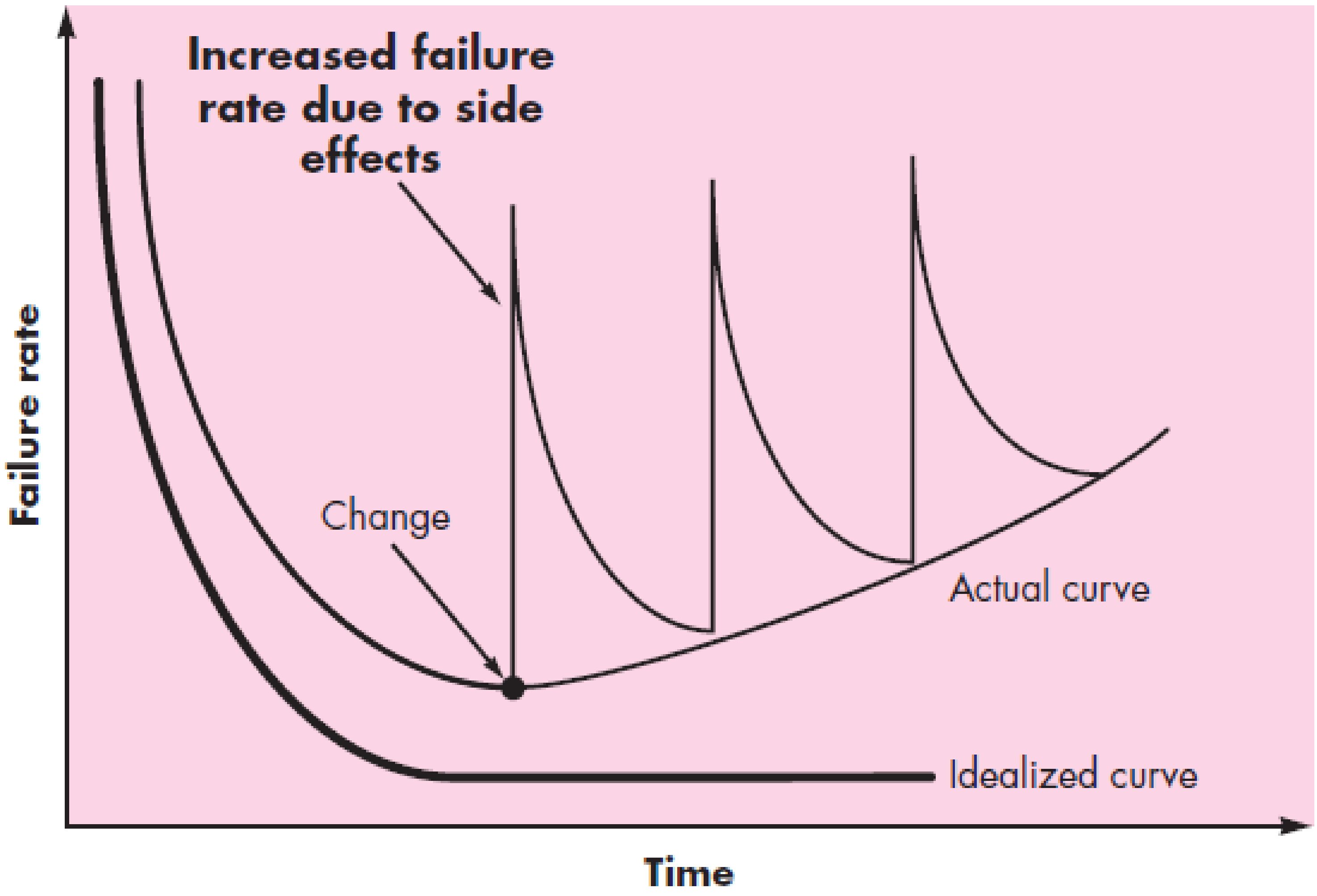
What is Software?



What is Software?

- For a software, undiscovered defects will cause high failure rates early in the life of a program.
- However, these are corrected and failure rate drops.
- "Software doesn't wear out. But it does *deteriorate!*"
  - When a hardware component wears out, it is replaced by a spare part. There are no software spare parts.

What is Software?



What is Software?

- *Although the industry is moving toward component-based construction, most software continues to be custom built.*
  - In the hardware world, component reuse is a natural part of the engineering process.
  - In the software world, it is something that has only begun to be achieved on a broad scale.

Software Application Domains

- System software
  - Collection of programs written to service other programs.
- Application software
  - Stand-alone programs that solve a specific business need
- Engineering/scientific software
  - Characterized by “number crunching” algorithms
- Embedded software
  - Resides within a product or system and is used to implement and control features and functions for the end user and for the system itself.

Software Application Domains

- Product-line software
  - Designed to provide a specific capability for use by many different customers.
- WebApps (Web applications)
  - Network-centric software category spans a wide array of applications.
- AI software
  - Makes use of nonnumeric algorithms to solve complex problems

Software—New Categories

- Open world computing—distributed computing
- Ubiquitous computing—wireless networks
- Open source—"free" source code
- Data mining
- Grid computing
- Cognitive machines
- Software for nanotechnologies

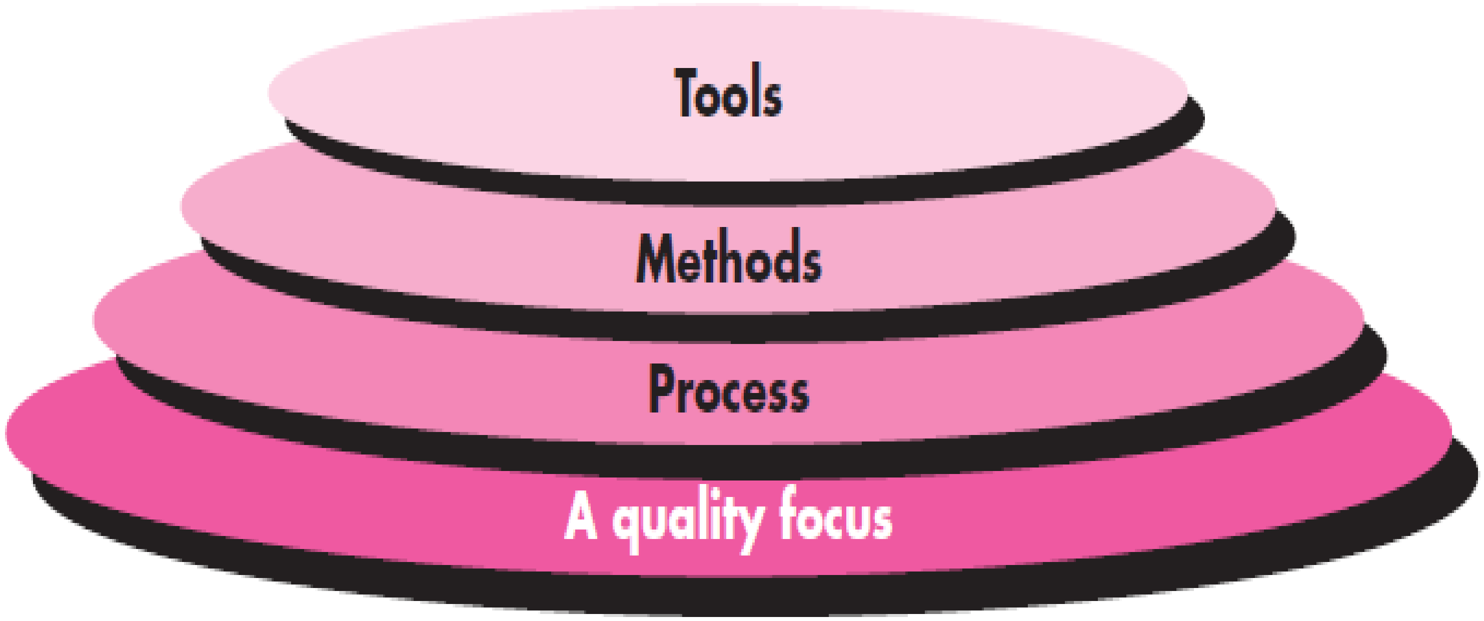
Legacy Software

- ***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.

Software Engineering

- 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.

SE: A Layered Technology



SE: A Layered Technology

- The bedrock that supports software engineering is a quality focus.
- The foundation for software engineering is the process layer.
- Software engineering methods provide the technical how-to's for building software.
- Software engineering tools provide automated or semi automated support for the process and the methods.

The Software Process

- A process is a collection of activities, actions, and tasks that are performed when some work product is to be created.
- Activity: To achieve broad objective regardless of size, domain or complexity
- Action: Set of tasks to produce major a work product
- Task: Focus on small and well defined objective

A Process Framework

- **Process framework**
  - **Framework activities**
    - Communication
    - Planning
    - Modeling
    - Construction
    - Deployment
  - **Umbrella Activities**

Framework Activities

- Communication: (To understand the domain)
  - Before any technical work can commence, it is critically important to communicate and collaborate with the customer and other stakeholders.
  - The intent is to understand stakeholders' objectives for the project and to gather requirements that help define software features and functions.

Framework Activities

- Planning (Like creating a map)
  - Create software project plan which defines the software engineering work by describing
    - the technical tasks to be conducted,
    - the risks that are likely,
    - the resources that will be required,
    - the work products to be produced,
    - and a work schedule.

Framework Activities

- Modeling: (Like creating a sketch)
  - During this phase, software engineers create models to better understand software requirements and the design that will achieve those requirements.
- Construction: (Implementation and Testing)
  - This activity combines code generation (either manual or automated) and the testing that is required to uncover errors in the code.

Framework Activities

- Deployment:
  - The software (as a complete entity or as a partially completed increment) is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

Framework Activities

- These five generic framework activities can be used during the development of
  - small, simple programs,
  - the creation of large Web applications,
  - and for the engineering of large, complex computer-based systems.

- The details of the software process will be quite different in each case, but the framework activities remain the same.

Umbrella Activities

- Software project management
- Formal technical reviews
- Software quality assurance
- Software configuration management
- Work product preparation and production
- Reusability management
- Measurement
- Risk management