

SWE 205 - Introduction to Software Engineering

Lecture - 1



Lecture Objectives

- Introduce software engineering and to explain its importance
- Develop a broad understanding of the software engineering domain



Student Activity

- What makes software so important?
 - A software product's impacts our life
- Behind the Scene Impact
 - Think of non-computer related business where software has a significant role



General Observations

- Software is used by virtually everyone in society either directly or indirectly
- The economies of the world are depend on software



Software Engineering Definition

- Software Engineering is concerned with theories, methods and tools for professional software development



What is software?

- Computer programs and associated documentation; such as
 - requirements,
 - design models; and
 - user manuals
- Software products may be developed for a particular customer or may be developed for a general market



Software Types

- Generic - developed to be sold to a range of different customers,
 - e.g. PC software such as Word or Excel
- Custom - developed for a single customer according to their specification
- Cooperative Solutions
 - Starting with generic system and customizing it to the needs of a particular customer. For example, Resource Planning (ERP) system



Software's Dual Role

- Software is a product
 - Delivers computing potential
 - Produces, manages, acquires, modifies, displays, or transmits information
- Software is a vehicle for delivering a product
 - Supports or directly provides system functionality
 - Controls other programs (e.g. operating systems)
 - Effects communications (e.g. networking software)
 - Helps build other software (e.g. software tools)

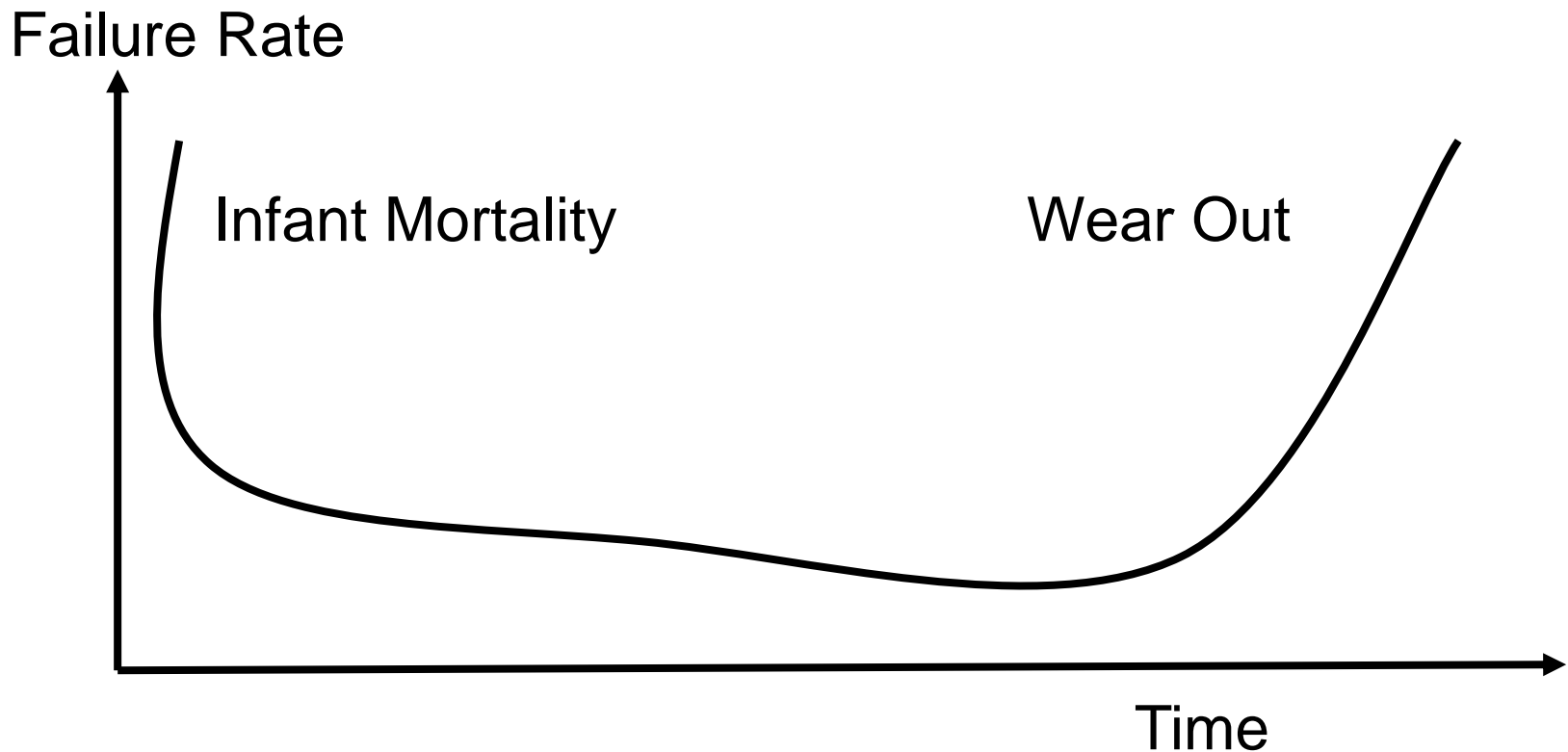


Software Characteristics

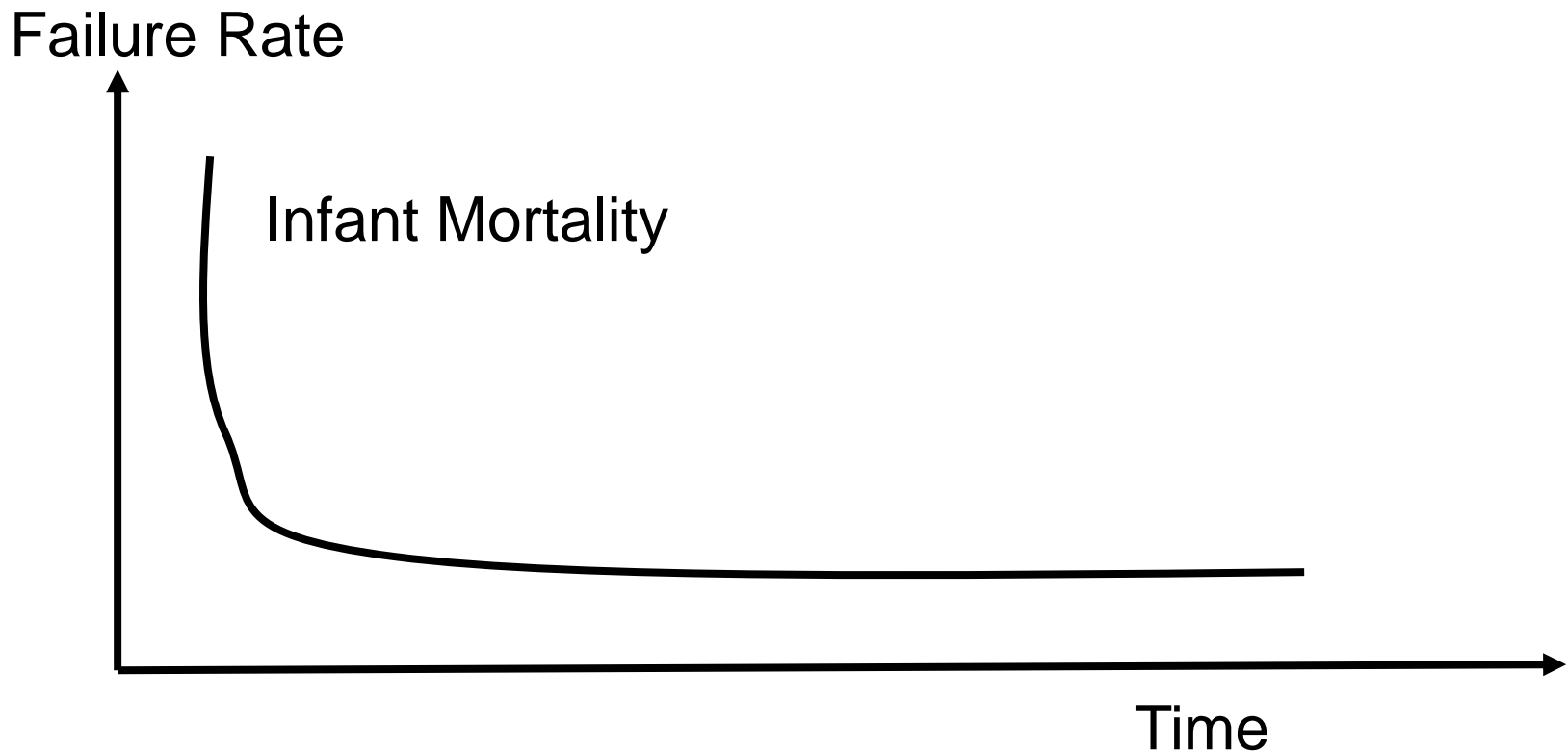
- To date, most software is still custom build.
 - Different to hardware manufacturing process.
 - However, software industry is moving towards component-based development
- Software does not wear out,
 - but it does deteriorate



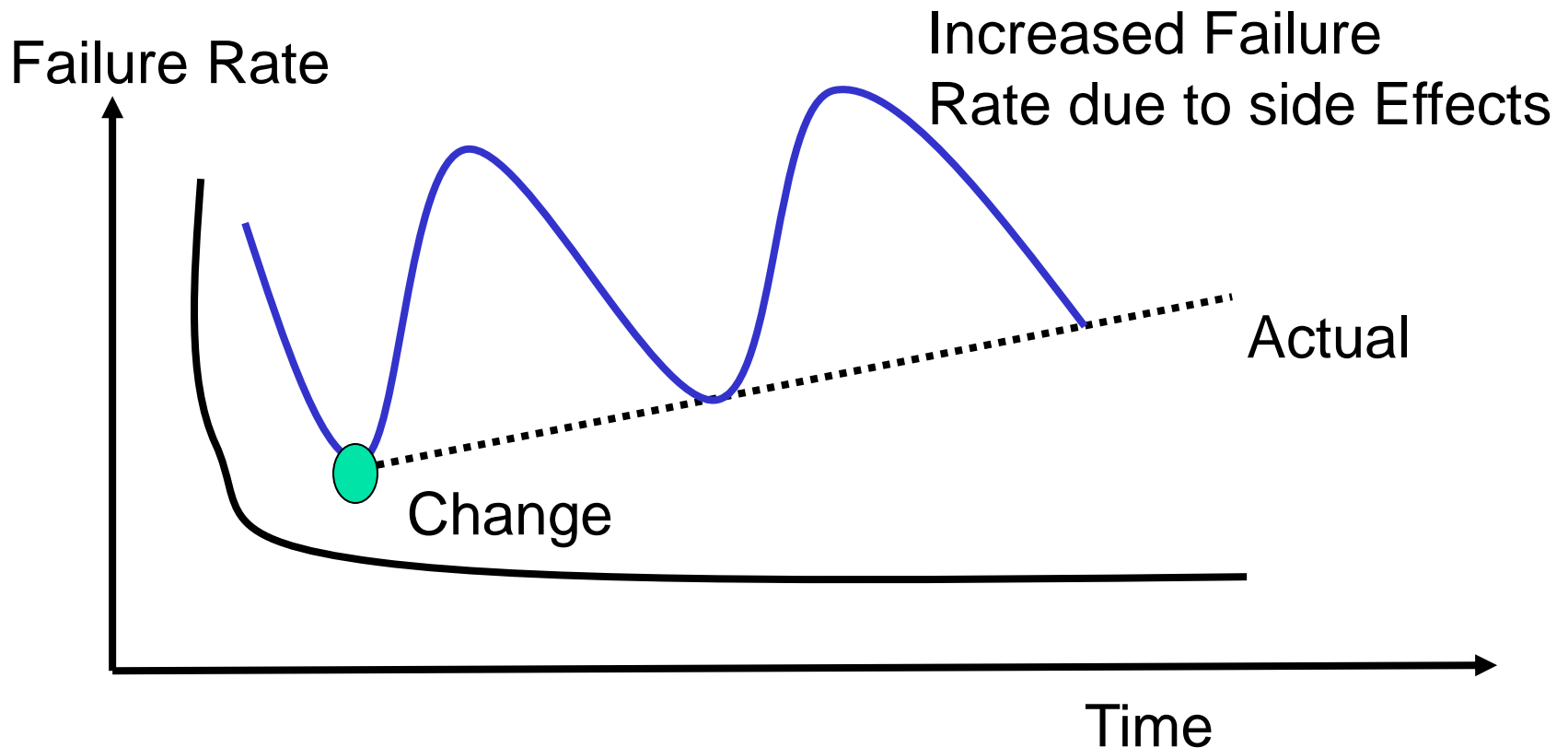
Hardware Failure



Software Failure (Ideal)



Software Failure (Realistic)





Software Applications

- No clear breakdown of application types, following are some generally accepted overlapping categories
 - System software
 - Real-time software
 - Business information software
 - Engineering & scientific software
 - Embedded software
 - Personal application software
 - Communication software



Attributes of a Good Software

- It is not enough just to produce software
 - Software should deliver the required functionality
- Software should have the appropriate product characteristics
 - The relative importance of these characteristics varies from product to product

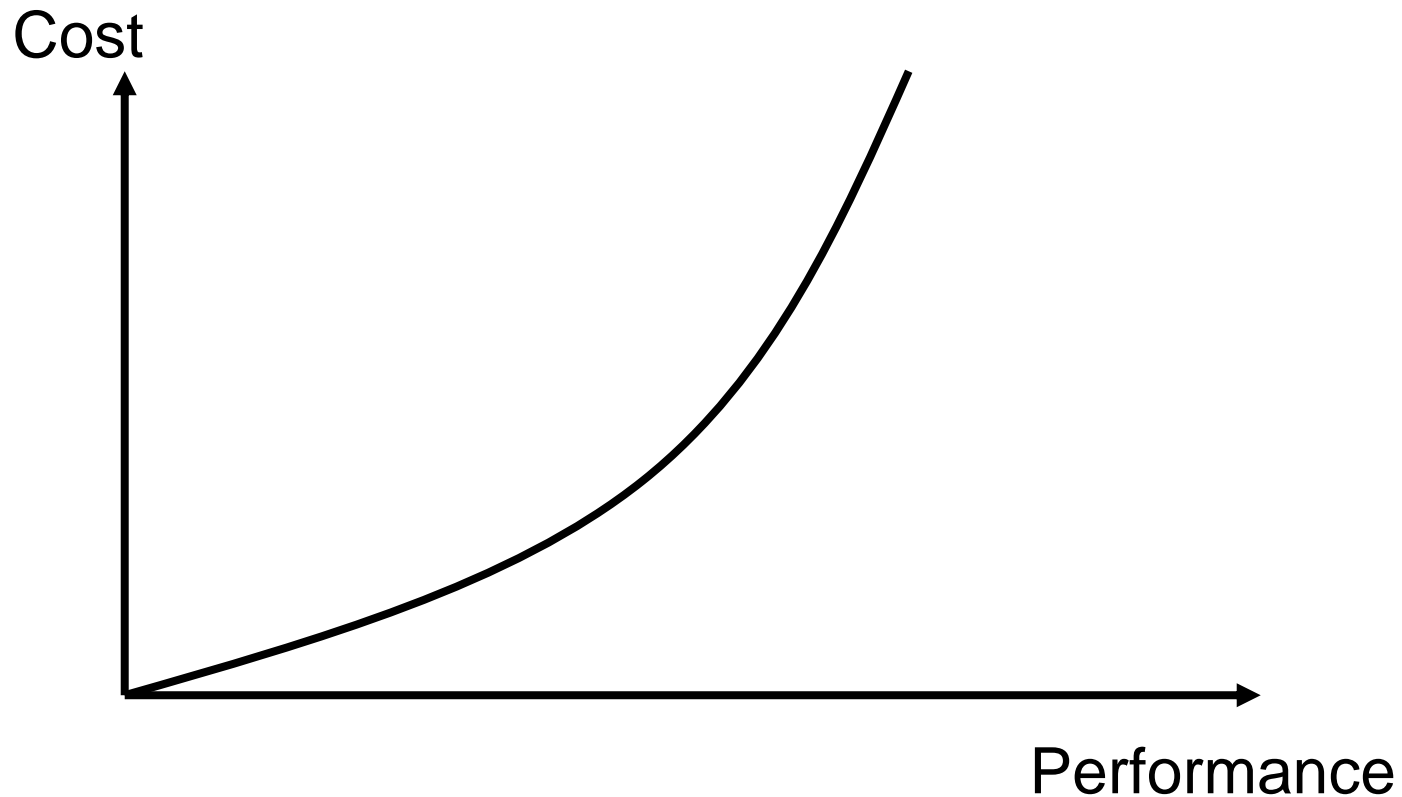


Attributes of a Good Software

- Characteristics relate to each other
 - Complex 'trade-offs'



Attributes of a Good Software





Attributes of a Good Software

- Maintainability
 - Software must evolve to meet changing needs;
- Dependability
 - Software must be trustworthy; e.g. reliability, security, safety.
- Efficiency
 - Software should not make wasteful use of system resources;



Attributes of a Good Software

- Usability

- Software must be accepted by the users for what it was designed.
- Appropriate user interface & adequate documentation.



Important Questions

- Why does it takes so long to get software finished?
- Why are development costs so high?
- Why can't we find all errors before we give the software to our customers?
- Why do we spend much time and effort in maintaining existing programs?



Software Development Process

- A set of activities whose goal is the development or evolution of software.
- Generic activities in all software processes are
 - Specification - what the system should do and its development constraints
 - Development - production of the software system
 - Validation - checking that the software is what the customer wants
 - Evolution - changing the software in responses to changing demands



Software Process Model

- A simplified representation of a software process, presented from a specific perspective.
- Examples of process perspectives are
 - Workflow - sequence of activities
 - Data-flow - information flow
 - Role/action - who does what



Key Points

- Software engineering is an engineering discipline that is concerned with all aspects of software production
- Software production consists of developed programs and associated documentation.
- Basic software activities are software specification, development, validation and evolution.