



SWE 205: Introduction to Software Engineering

Lecture 2

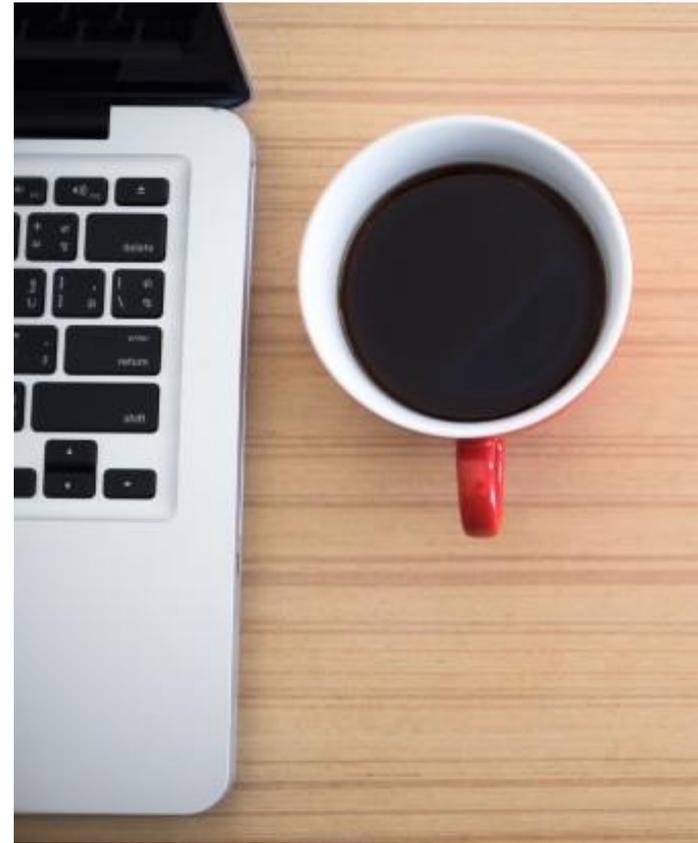
Introduction II

Course Topics

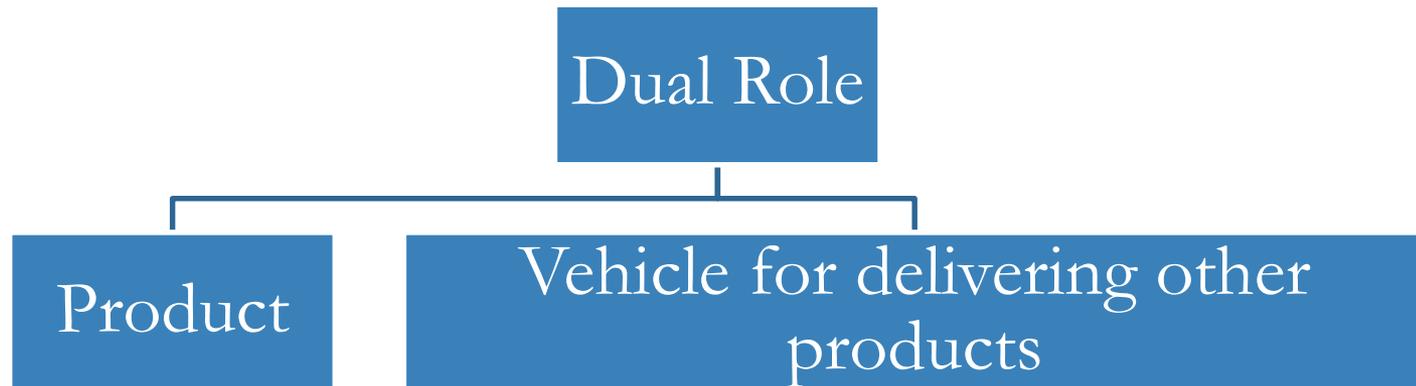
- Introduction
- Software Process Models
- Requirements Engineering
- Modeling
- Programming Languages
- Software Construction Techniques
- Testing
- Refactoring
- Project Management
- Ethical Issues

Lecture Objectives

- ✓ Software Dual Rule
- ✓ Software Vs. Hardware
- ✓ Software Quality



Software's Dual Role



- Software is a product
 - Delivers computing potential
 - Software is an information transformer: Produces, manages, acquires, modifies, displays, or transmits information

Software's Dual Role



- Software is a vehicle for delivering a product
 - Basis for the controls of the computer (e.g. operating systems)
 - Communication of Information (e.g. networking software)
 - Creation and control of other software (e.g. software tools and environments)

Software Applications

- No clear breakdown of application types, following are some generally accepted overlapping categories.
 1. *Stand-alone applications*: run on a local computer and do not need to be connected to a network (e.g., office applications, photo manipulation software).
 2. *Interactive transaction-based applications*: execute on a remote computer and accessed by users from their own PCs/terminals.
 3. *Embedded control systems*: software control systems that control and manage hardware devices (e.g., software in a mobile phone, software in a microwave oven).
 4. *Batch processing systems*: business systems designed to process data in large batches (e.g., as phone billing systems, and salary payment systems)

Software Applications



5. *Entertainment systems*: Primarily for personal use and intended to entertain the user (e.g., games)
 6. *Systems for modeling and simulation*: Developed by scientists and engineers to model physical processes. They are often computationally intensive (e.g., molecular biology, astronomy, etc.)
 7. *Data collection systems*: Collect data from their environment using a set of sensors and send that data to other systems for processing.
 8. *Systems of systems*: Systems composed of a number of other software systems. Some of these may be generic software products, such as a spreadsheet program.
- 

Software Characteristics

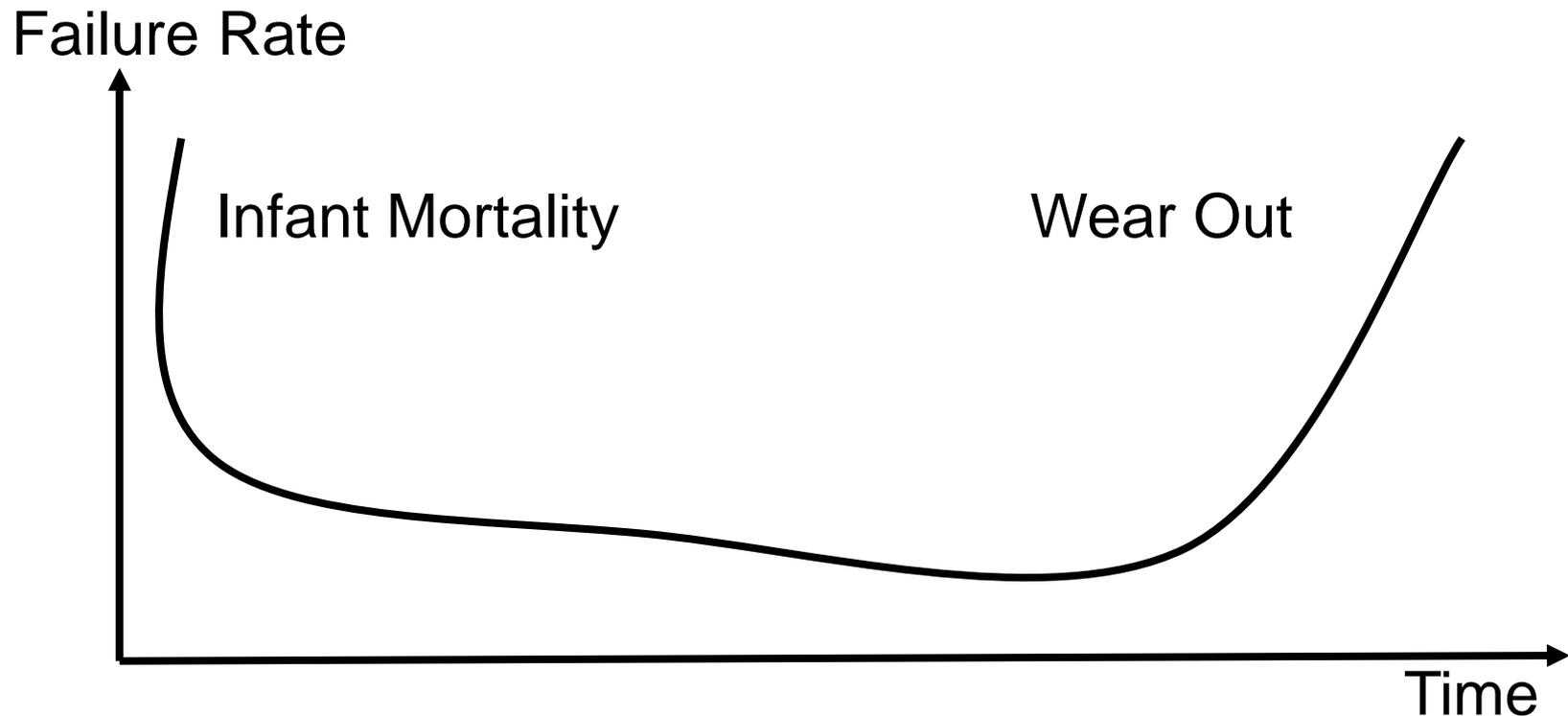


- Software is developed or engineered; it is not manufactured in the classical sense
 - Different from hardware manufacturing process.
 - Software is custom built
 - However, software industry is moving towards component-based development

- Software does not wear out,
 - but it does deteriorate

Hardware Failure

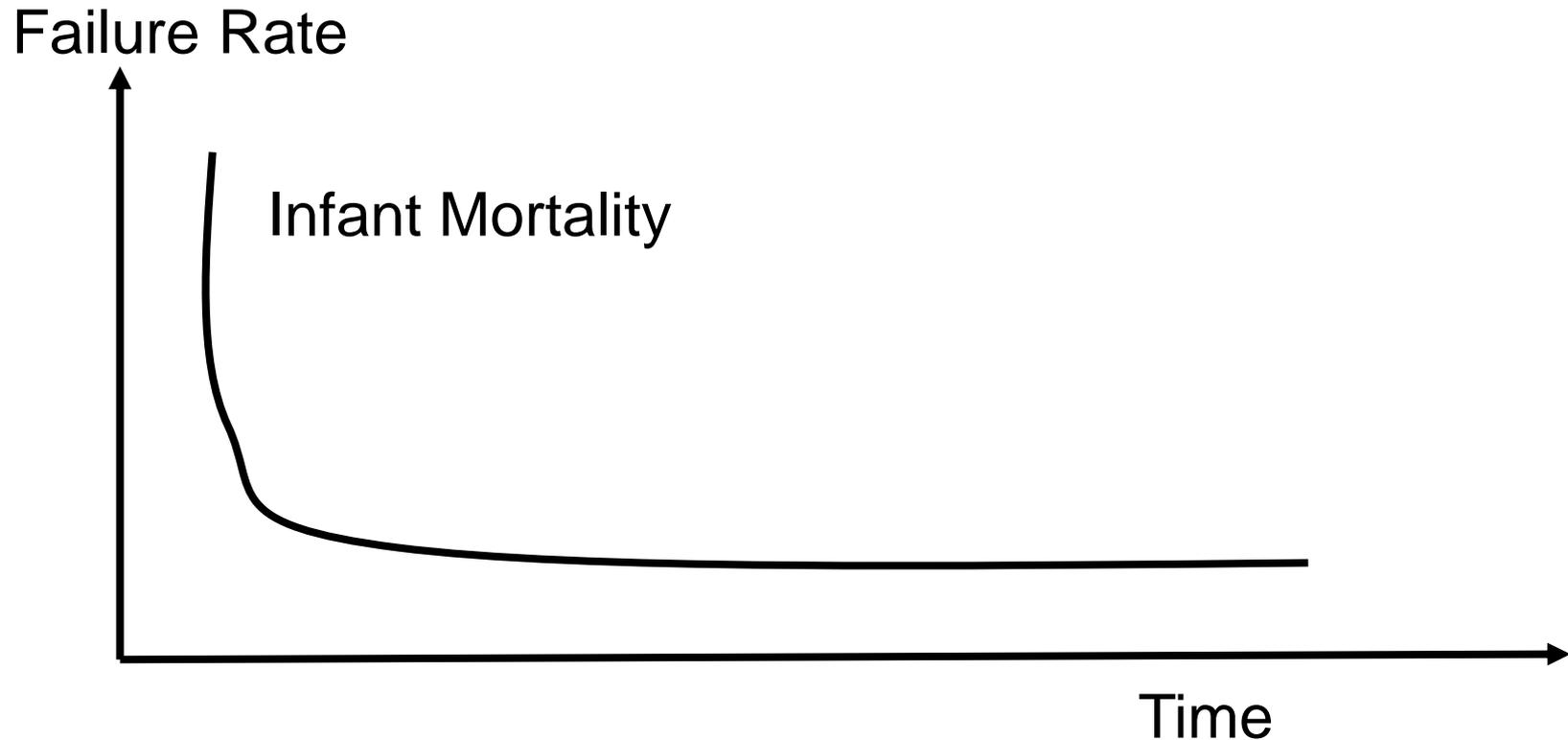
- Bathtub curve



Hardware components suffer from the cumulative effects of dust, vibration, abuse, temperature extremes, etc.

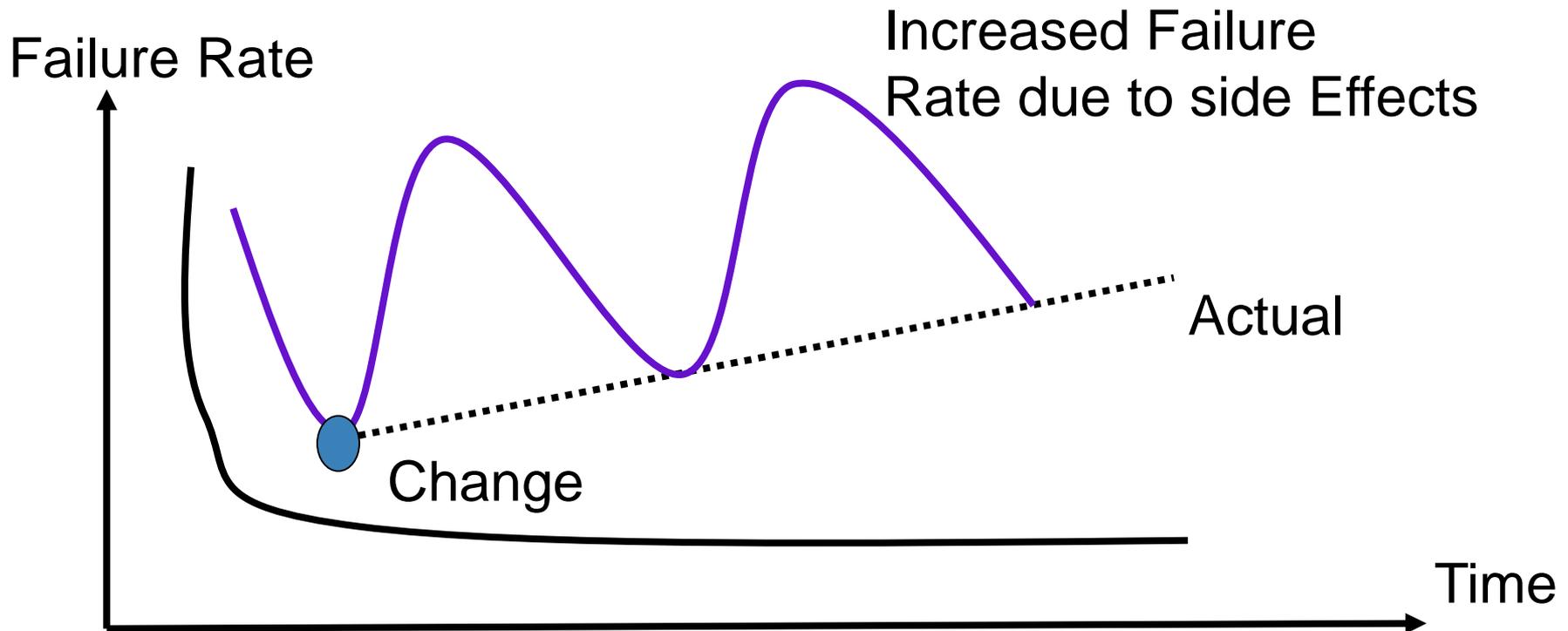
Software Failure (Ideal)

- Idealized curve (oversimplified)



Undiscovered defects will cause high failure rates early in the life of a program. These are corrected and the curve flattens.

Software Failure (Realistic)



Before the curve can return to the original steady-state failure rate, another change is requested, causing the curve to spike again. Slowly, the minimum failure rate level begins to rise—the software is deteriorating due to change.

Software Quality



- 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

Software Quality Attributes



- Usability
 - Users can learn the software to get their job done easily and fast.
 - Efficiency
 - It doesn't waste resources such as CPU time and memory
 - Dependability
 - Software must be trustworthy; e.g. reliability, security, safety
 - Maintainability
 - It can be easily changed
 - Reusability
 - Its parts can be used in other projects, so reprogramming is not needed
- 

Software Quality and Stakeholders

Customer (those who pay):

solves problems at
an acceptable cost in
terms of money paid and
resources used

User:

easy to learn;
efficient to use;
helps get work done



Developer:

easy to design;
easy to maintain;
easy to reuse its parts

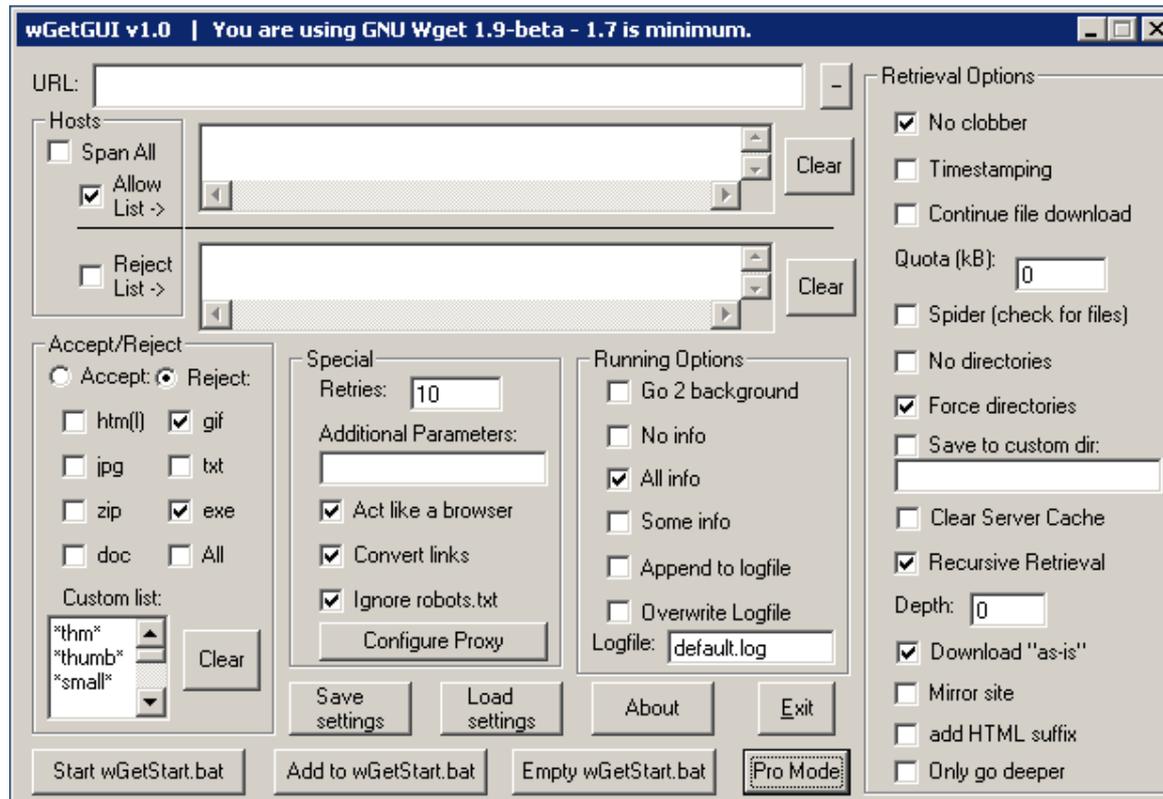
Development manager:

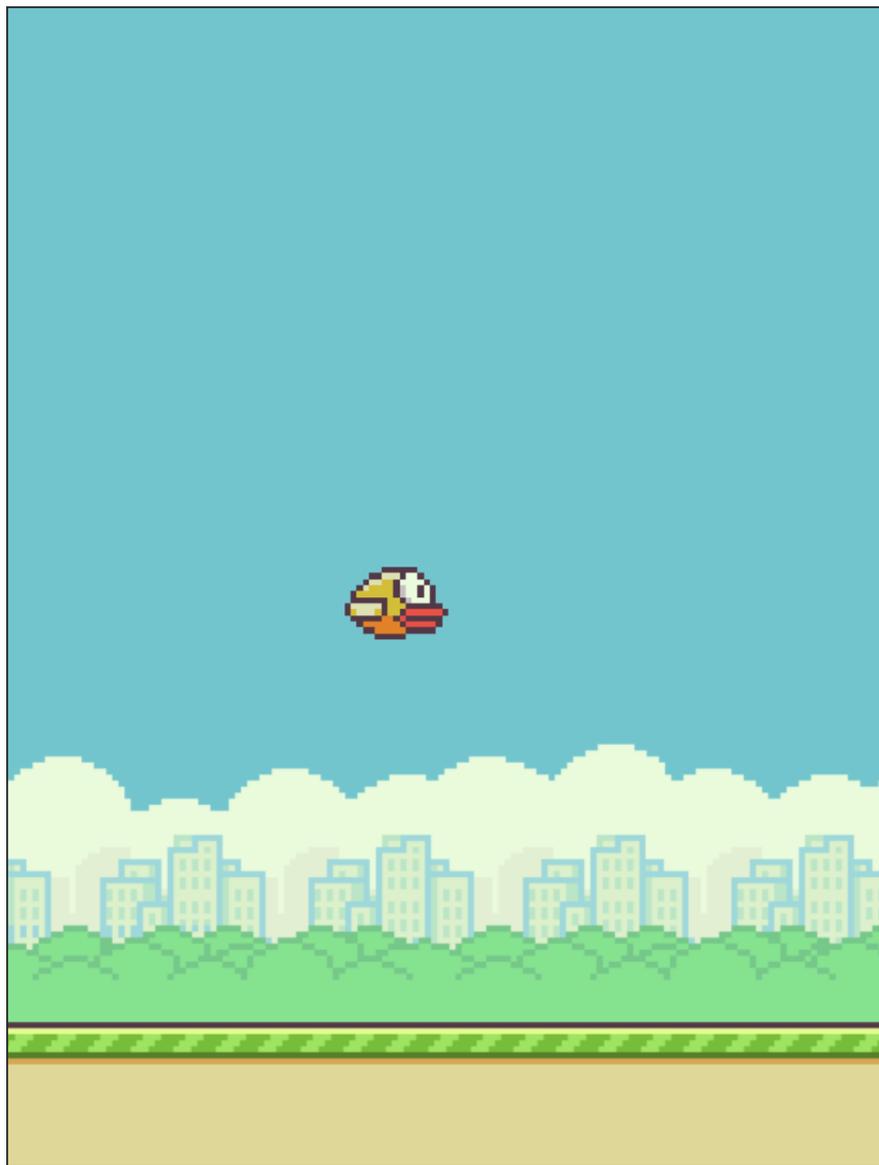
sells more and
pleases customers
while costing less
to develop and maintain

Software Quality Attribute

■ Usability

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





<http://flappybird.io/>

* - Denotes Required Information

> **1 Donation** > 2 Confirmation > Thank You!**Donor Information****First Name*** **Last Name*** **Company** **Address 1*** **Address 2** **City*** **State*** **Zip Code*** **Country*** **Phone** **Fax** **Email***

Donation Amount* None \$50 \$75 \$100 \$250 Other
 (Check a button or type in your amount) **Other Amount \$**

Recurring Donation I am interested in giving on a regular basis.
 (Check if yes) Monthly Credit Card \$ For Months

Honorarium and Memorial Donation Information

I would like to make this donation To Honor
 In Memory of

Name **Acknowledge Donation to** **Address** **City** **State** **Zip** **Additional Information**

Please enter your name, company or organization as you would like it to appear in our publications:

Name

- I would like my gift to remain anonymous.
 My employer offers a matching gift program. I will mail the matching gift form.
 Please save the cost of acknowledging this gift by not mailing a thank you letter.

Comments
 (Please type any questions or feedback here)

How may we contact you? E-mail
 Postal Mail
 Telephone
 Fax

I would like to receive newsletters and information about special events by:

- E-mail
 Postal Mail

 I would like information about volunteering with the

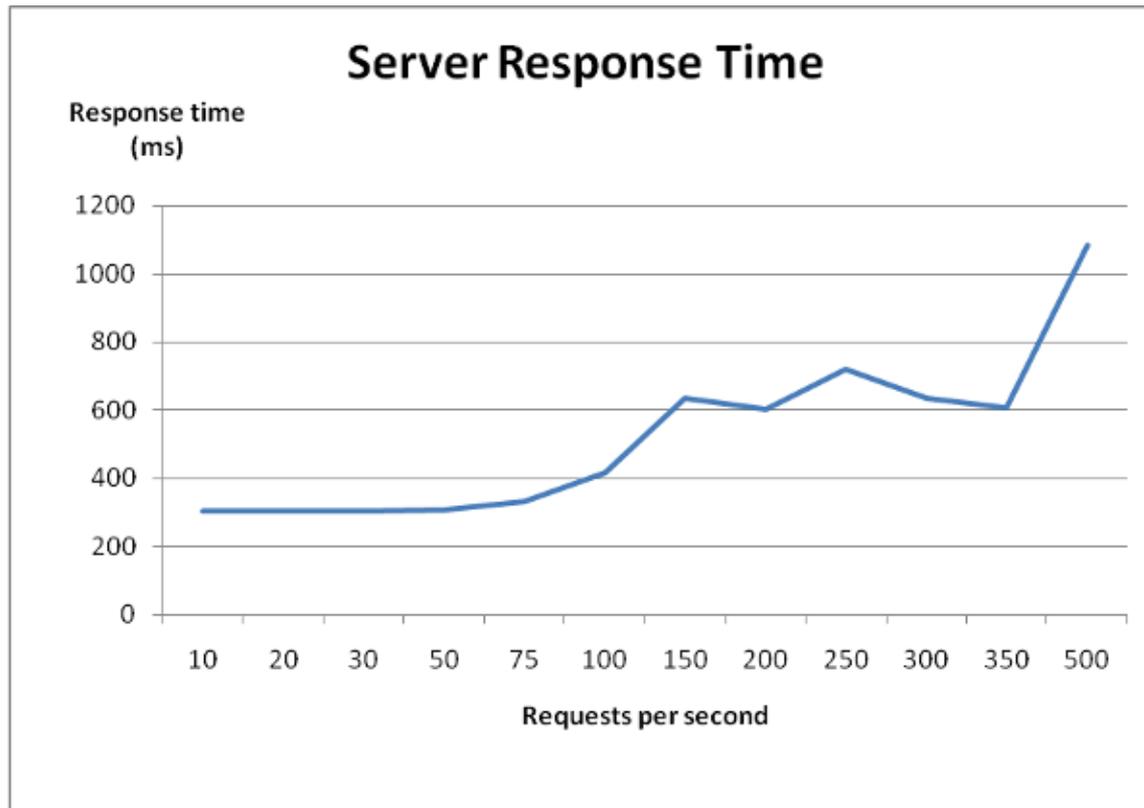
Donate online with confidence. You are on a secure server.

If you have any problems or questions, please contact [support](#).

Software Quality Attribute

■ Efficiency

- Software should not make wasteful use of system resources
- e.g., CPU, Memory, Drive space etc.



Software Quality Attribute

- Dependability
 - Software must be trustworthy; e.g. reliability, security, safety, availability.
- Principal dimensions of dependability are:
 - Availability
 - System ability to deliver services when requested
 - Reliability
 - System does what it is required to do without failing
 - Safety
 - System's ability to operate, normally or abnormally, without danger of causing human injury or death and without damage to the system's environment
 - Security
 - System ability to protect itself against intrusion

A problem has been detected and windows has been shut down to p
to your computer.

The problem seems to be caused by the following file: SPCMDCON.

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly ins
If this is a new installation, ask your hardware or software ma
for any windows updates you might need.

If problems continue, disable or remove any newly installed har
or software. Disable BIOS memory options such as caching or sha
If you need to use Safe Mode to remove or disable components, r
your computer, press F8 to select Advanced Startup Options, and
select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x000000

Software Quality Attribute

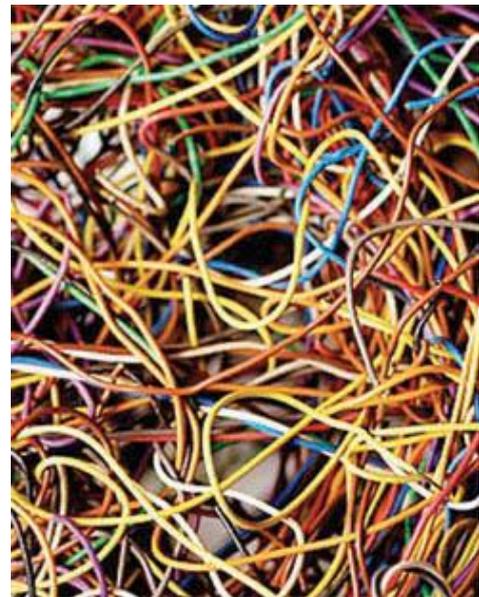
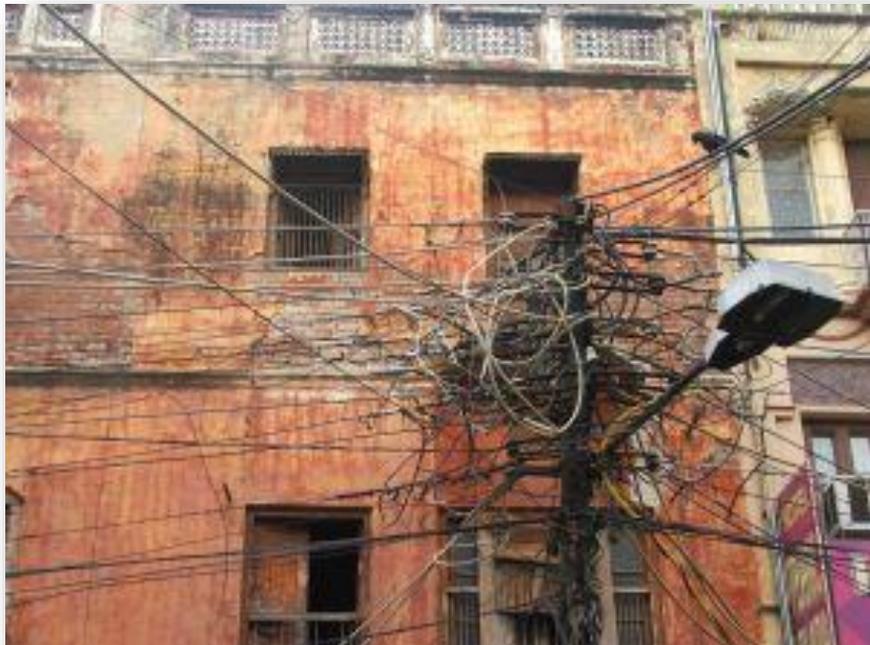
- Dependability is important for:
 - Safety-critical systems
 - Failure results in loss of life, injury or damage to the environment
 - E.g. Chemical plant protection system
 - Mission-critical systems
 - Failure results in failure of some goal-directed activity
 - E.g. Spacecraft navigation system
 - Business-critical systems
 - Failure results in high economic losses
 - E.g. Customer accounting system in a bank

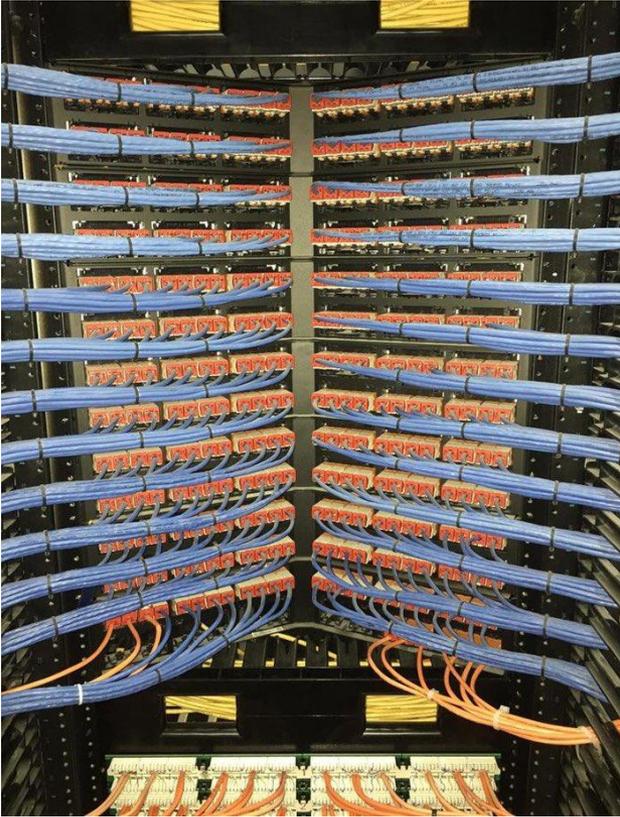


Software Quality Attribute



- Maintainability
 - Software should be written in such a way so that it can evolve to meet the changing needs of customers.





Complex
But
Organized and
Maintainable

Software Quality: Conflicts and Objectives

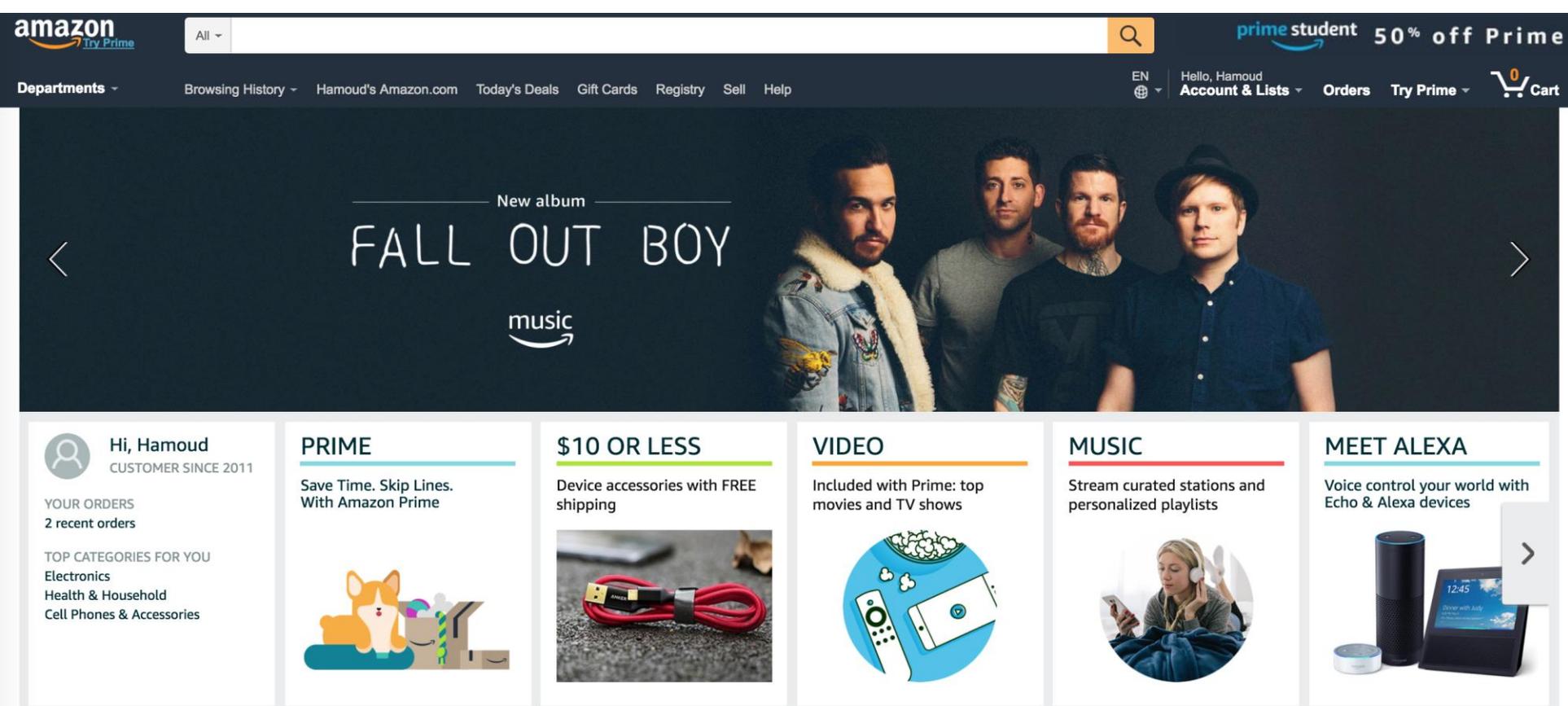
- Characteristics relate to each other

Complex Trade-Offs

- The different qualities can conflict
 - Increasing efficiency can reduce maintainability or reusability
 - Increasing usability can reduce efficiency
- Setting objectives for quality is a key engineering activity
 - You then design to meet the objectives

Discussion

- List important quality attributes for 



The screenshot displays the Amazon homepage with the following elements:

- Navigation Bar:** Includes the Amazon logo, a search bar, and links for Departments, Browsing History, Hamoud's Amazon.com, Today's Deals, Gift Cards, Registry, Sell, and Help. It also shows the user's name (Hamoud), account lists, orders, and a shopping cart with 0 items.
- Featured Music:** A banner for the new album "FALL OUT BOY" by the band Fall Out Boy, featuring a photo of the four band members.
- Customer Profile:** A tile for "Hi, Hamoud" (Customer since 2011) showing "YOUR ORDERS" (2 recent orders) and "TOP CATEGORIES FOR YOU" (Electronics, Health & Household, Cell Phones & Accessories).
- PRIME:** A tile with the text "Save Time. Skip Lines. With Amazon Prime" and an illustration of a dog sitting in a cardboard box.
- \$10 OR LESS:** A tile with the text "Device accessories with FREE shipping" and an image of a red USB cable.
- VIDEO:** A tile with the text "Included with Prime: top movies and TV shows" and an illustration of a game console and a smartphone.
- MUSIC:** A tile with the text "Stream curated stations and personalized playlists" and an image of a woman listening to music on headphones.
- MEET ALEXA:** A tile with the text "Voice control your world with Echo & Alexa devices" and an image of an Echo smart speaker and an Echo Show smart display.

Key Points



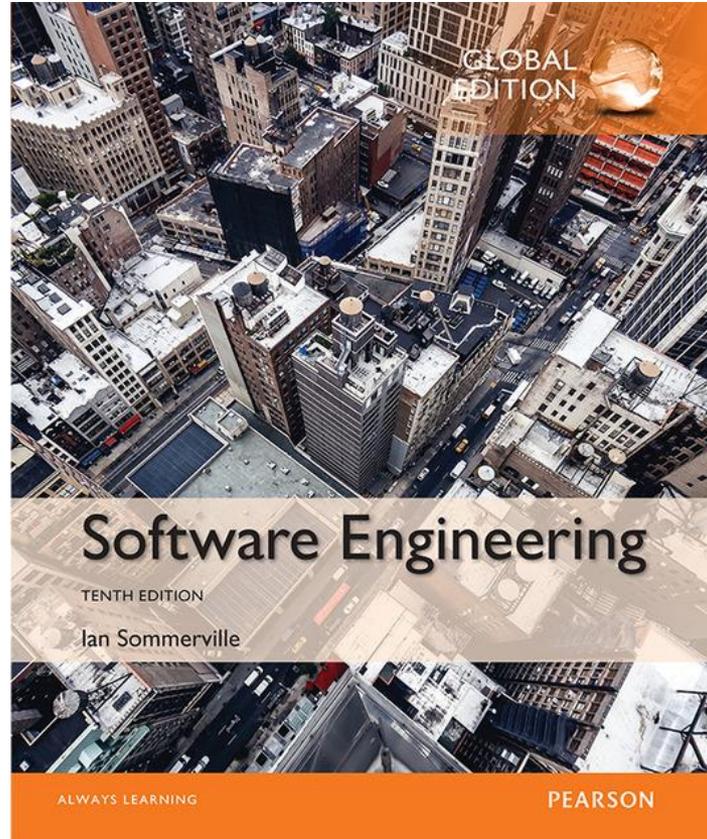
- Software's dual roles
 - Software is a product; and
 - Software is a vehicle for delivering a product

 - Software does not wear out,
 - but it does deteriorate

 - It is not enough just to produce software
 - Software should deliver the required functionality
 - Software characteristics also relate to each other
- 

Read

Chapter 1



References



- Ian Sommerville, “Software Engineering”, 10th Edition, Addison-Wesley, 2015.
 - Timothy C. Lethbridge and Robert Laganière, “Object-Oriented Software Engineering: Practical Software Development using UML and Java”, 2nd Edition, McGraw Hill, 2001.
 - R. S. Pressman, Software Engineering: A Practitioner’s Approach, 10th Edition, McGraw-Hill, 2005.
- 

Course Topics

- Introduction
- Software Process Models
- Requirements Engineering
- Modeling
- Programming Languages
- Software Construction Techniques
- Testing
- Project Management
- Refactoring
- Ethical Issues