

COSC 412

Introduction

Spring 2025

Dr. Akshita Maradapu Vera Venkata Sai
Department of Computer and Information Sciences
Towson University

Some slides are adapted from slides provided by Dr. Josh Dehlinger and Prof. Rebecca Broadwater.

Why learn software engineering ?

- What is software ?
- What is software engineering ?
- Why do we learn software engineering?
- Facts and failures in the history
- of software engineering
- Goal of the course



Definition

- What is software?
 - (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.

Definition

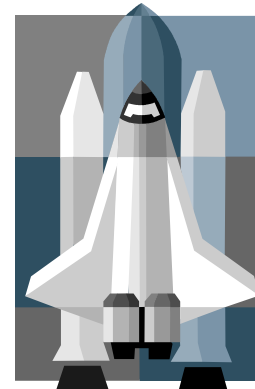
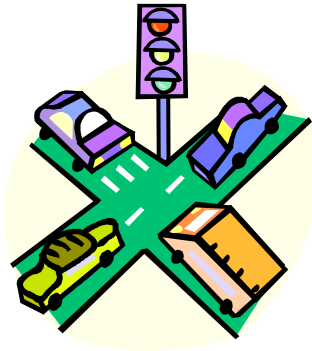
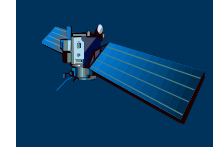
- What is software engineering?

IEEE definition :

(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).

Software is Everywhere



Problems

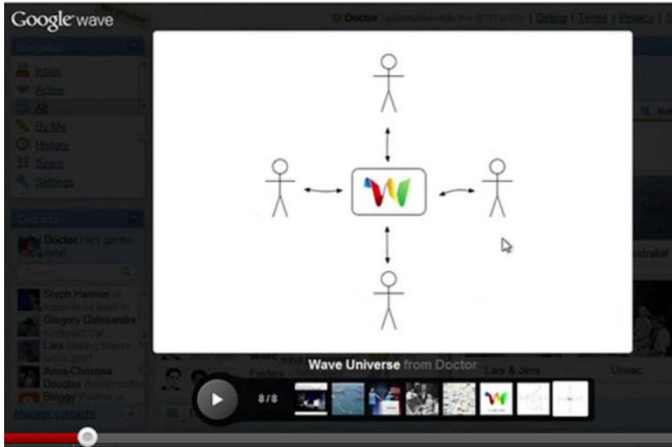
- However, software project failures are everywhere...
- A UK government project to develop a software system that provided national support to courts
 - Project estimated to cost ~\$300 million and be finished in 2001
 - By 2004, it was still not finished and had cost ~\$750 million
- This is not uncommon. Too many projects are over budget, late and don't deliver the intended features.

16% success rate on average; 9% success rate for large companies

Failed Google Projects

1 Google Wave

< PREV NEXT >



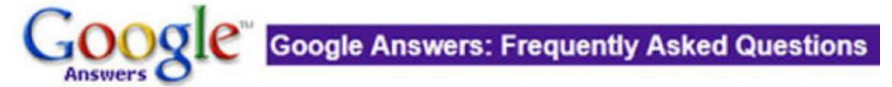
2 Google Video

< PREV NEXT >



9 Google Answers

< PREV NEXT >



Google Answers: Frequently Asked Questions

About Google Answers

1. [What is Google Answers?](#)
2. [Do I need to create an account to use Google Answers?](#)
3. [Do I need to log in to Google Answers every time I visit?](#)
4. [What happens if I check the "Remember me" box on the login page?](#)
5. [Where do I find Google Answers' Terms of Service?](#)

Asking a question

4 Google Buzz

< PREV NEXT >



5 Google Notebook and Shared Stuff



Stopping development on Google Notebook

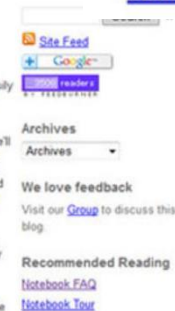
Wednesday, January 14, 2009 6:30 PM
Posted by Raj Krishnan, Product Manager, Google Notebook

At Google, we're constantly working to innovate and improve our products so people can easily find and manage information. At times though, we have to decide where to focus our efforts and which technologies we expect will yield the most benefit to users in the long run.

Starting next week, we plan to stop active development on Google Notebook. This means we'll no longer be adding features or offering Notebook for new users. But don't fret, we'll continue to maintain service for those of you who've already signed up. As part of this plan, however, we will no longer support the Notebook Extension, but as always users who have already signed up will continue to have access to their data via the web interface at <http://www.google.com/notebook>.

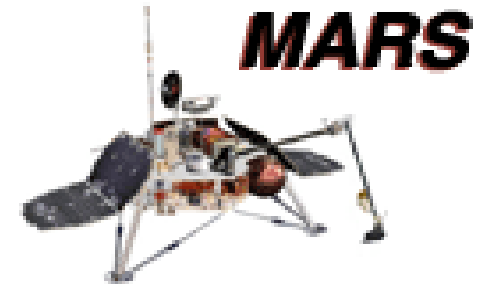
If you haven't used Notebook in the past, we invite you to explore the other Google products that offer Notebook-like functionality. Here are a few examples, all of which are being actively improved and should meet your needs:

- [SearchWiki](#) - We recently launched a feature on Search that will let you re-rank, comment, and personalize your search results. This is useful when you've found some



Failures Cost Money

- In September 1999, NASA's Mars lander crashed due to a units integration fault, over \$50 million !
- Huge losses due to web application failures
 - Financial services : \$6.5 million per hour
 - Credit card sales applications : \$2.4 million per hour
- In August 2013, Amazon was down for 30 minutes
 - Lost \$2 million
- 2007 : Symantec says that most security vulnerabilities are due to faulty software
- Better software engineering and testing could solve most of these problems



Failures Cost Money

- Example : CrowdStrike's faulty update in July 2024
 - Erroneous update pushed on the Falcon endpoint for Windows systems
 - Led to a Denial of Service issue – IT outage of 8.5 million Microsoft devices
 - Costed \$5.4 billion to the companies that used it:
 - Healthcare – procedures got delayed
 - Transport – Flights got cancelled
 - Emergency – 911 centers were not functioning properly

CROWDSTRIKE OUTAGE

A wave of IT outages swept across the globe Friday morning, sending airports, airlines, banks and other institutions into a screeching halt as some Microsoft-based computers ceased to work.

WHAT IS CROWDSTRIKE?

An American cybersecurity technology firm that provides cloud workload protection, threat intelligence and cyberattack response services.

WHEN DID THE OUTAGE OCCUR?

American Airlines, United and Delta asked the FAA for global ground stop on all flights, according to an alert from the FAA on Friday morning and, as of 7:54 AM EST, over 1,000 flights have been cancelled in the U.S.

SOME HOSPITALS HAVE ALSO BEEN AFFECTED

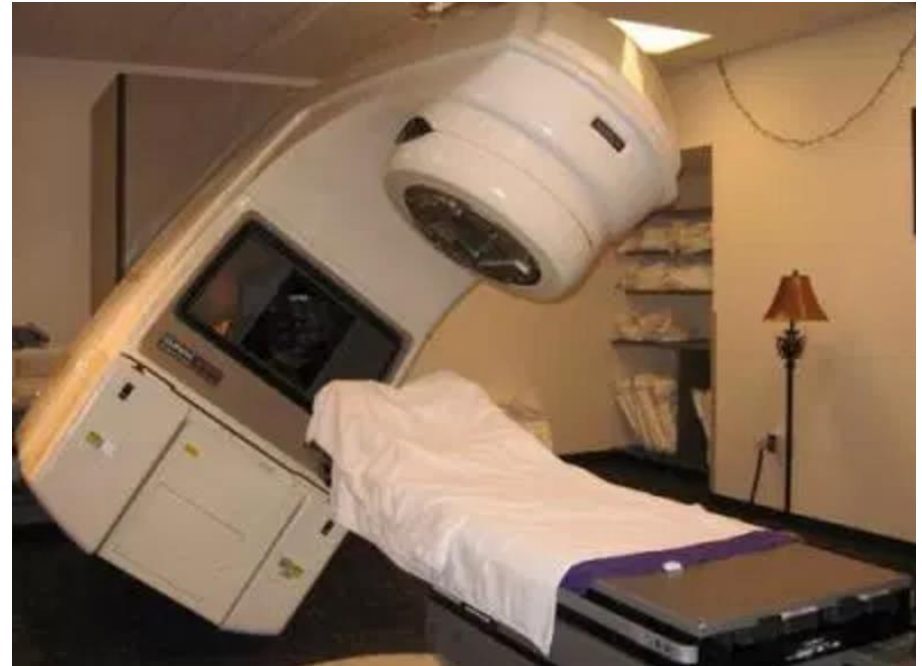
Mass General Brigham in Boston, Massachusetts, cancelled all elective surgeries on Friday and Cincinnati Children's Hospital Medical Center in Avondale, Ohio, also reported some disruptions.

WHO ELSE WAS AFFECTED BY THE OUTAGE?

- Some Microsoft apps and services
- The Department of Justice
- Berlin airport in Germany
- Gatwick airport in the UK
- The London Stock Exchange

Failures Cost Lives

- Therac-25
 - Radiation treatment machine
 - Designed to deliver small doses of radiation through filters to treat cancers, tumors
 - Has a bug in its software
 - Send lethal dose of radiation
 - Six people died due to the bug



Northeast Blackout of 2003

508 generating units and 256 power plants shut down

Affected 10 million people in Ontario, Canada

Affected 40 million people in 8 US states

Financial losses of \$6 Billion USD

The **alarm system** in the energy management system **failed due to a software error** and operators were not informed of the power overload in the system



Why does Software Fail?



How the customer explained it



How the Project Leader understood it



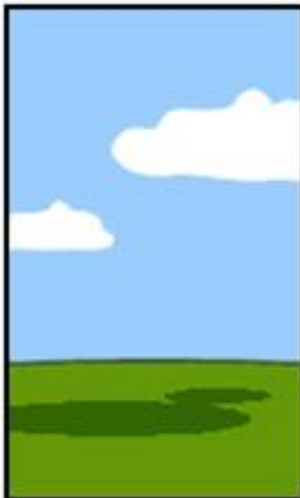
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



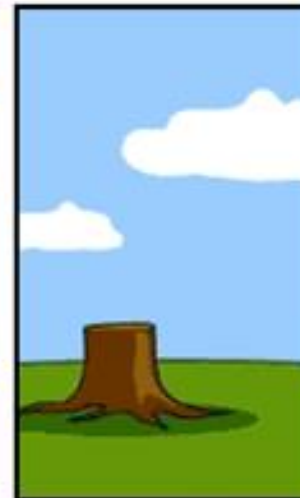
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

Why Does Software Fail?

- How many faults are there?
 - Hatton 1998, 6-30 faults per 1000 lines of code
 - DoD, 5-15 faults per 1000 lines of code
 - 1.25 hours to find, 2-9 hours to fix
 - Windows XP : 50 million lines of code

Why Does Software Fail?

- Unrealistic project goals
- Inaccurate estimates of needed resources
- Poor system requirements
- Poor reporting of the project's status
- Unmanaged risks
- Inability to handle complexity
- Use of immature technology
- Sloppy development practices
- Poor project management
- Stakeholder politics

Lessons from Faults

- Many of these faults could have been discovered through:
 - Better requirements/design specifications
 - So design your projects carefully!
 - Testing
 - Unit-level testing
 - System-wide and regression testing
 - Because a test was successful in the past doesn't mean it will stay that way!
 - Changes in one module might have subtle influence on another
 - Entire suite of tests must be re-run when a single module is changed

Consider This...

- Software engineering will make us create voluminous and unnecessary documentation and will invariably slow us down.

FALSE

- Reality: Software engineering is not about creating documents! It is about creating quality. Better quality leads to reduced rework. Reduced rework results in faster delivery times.
- Goal: Convince you that software engineering is necessary and helpful for successful projects.

Consider This...

- A general statement of objectives is sufficient to begin writing programs – we can fill in the details later

FALSE

- Reality: A poor up-front definition is the major cause of failed software efforts. A formal and detailed description of the information domain, function, behavior, performance, interfaces, etc. is essential. These can be determined only after thorough communication between customer and developer.
- Goal: Provide the techniques and tools during requirements engineering to get the needed details.

Consider This...

- The only deliverable work product for a successful project is the working program

FALSE

- Reality: A working program is only one part of a software configuration that includes many elements. Documentation provides a foundation for successful engineering and, more important, guidance for software support.
- Goal: Demonstrate that software engineering assets are valuable to the resulting system.

Consider This...

- Once we write the program and get it to work, our job is done

FALSE

- Reality: Someone once said that “the sooner you begin ‘writing code’, the longer it’ll take you to get done”. Industry data says that between 60 and 80 percent of all effort expended on software will be expended after it is delivered to the customer for the first time.
- Goal: Prove to you that software development does not start and stop at implementation.

Activity

- Pick a company of your choice
- Select one product from the company
- Discuss how this product evolved or changed over time

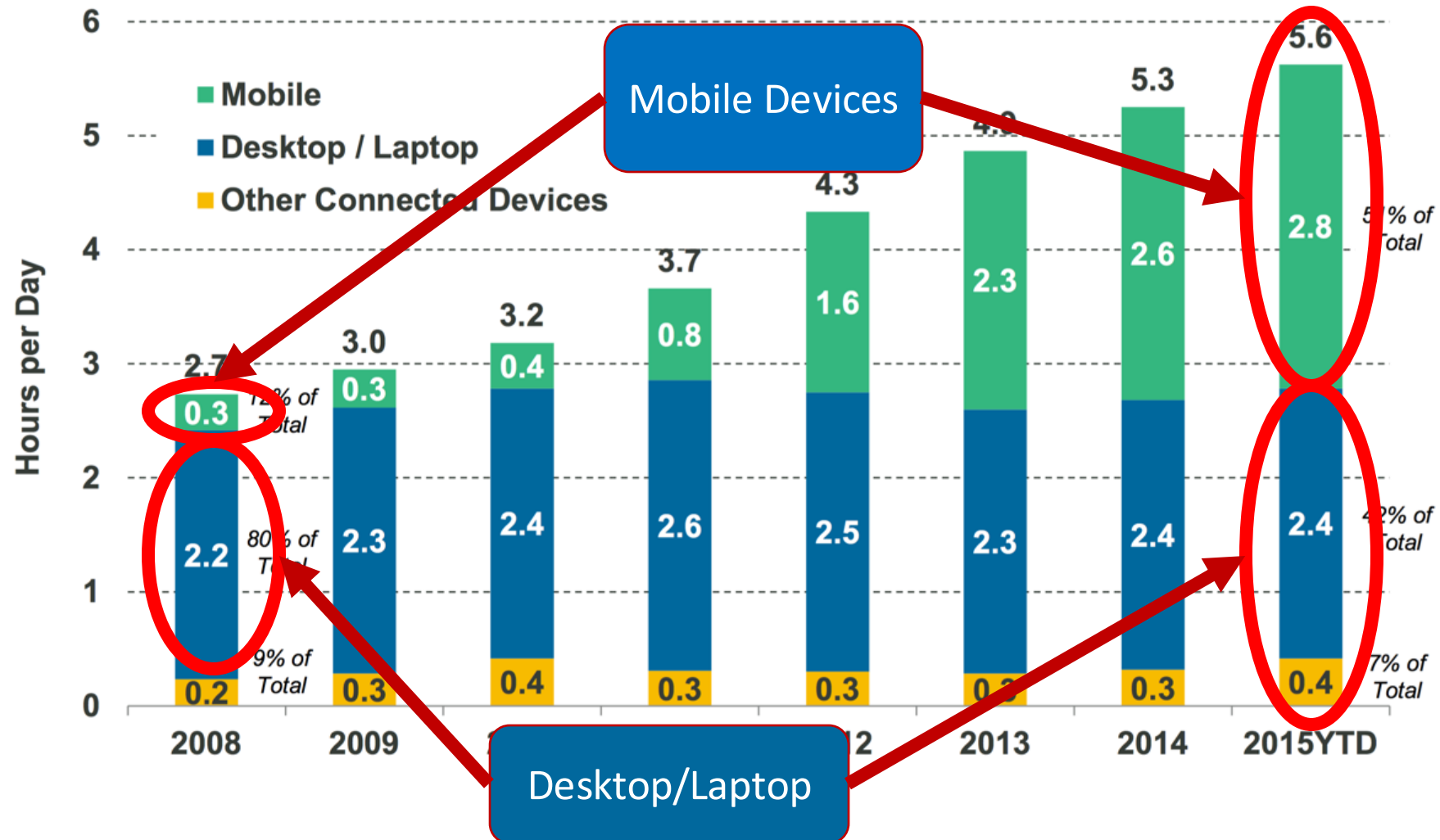
Changes in Software

Due to Changes in Technology

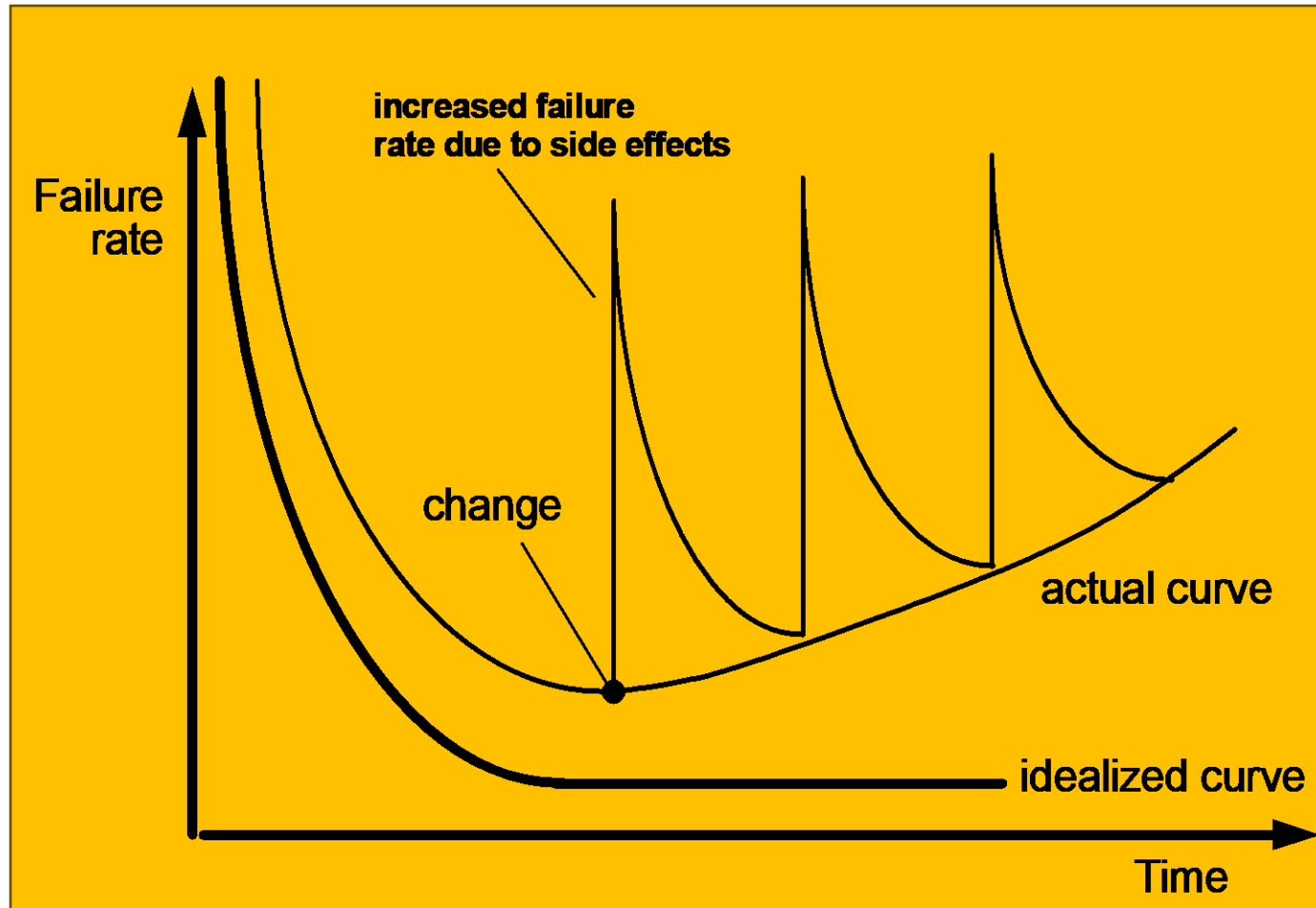


Changes in Software

Time Spent per Adult User per Day with Digital Media, USA, 2008 – 2015YTD



Wear vs. Deterioration



No software
spare parts

How to deal with this?

- Break down the product into several steps
- Focus on component-driven development
- Follow an incremental / prototype-based process
- Assess the changes and the compatibility of your current system
- Software Isolation

Goals of This Class

Provide you the fundamental knowledge, techniques and tools that contribute to successful software development projects.