Software Engineering

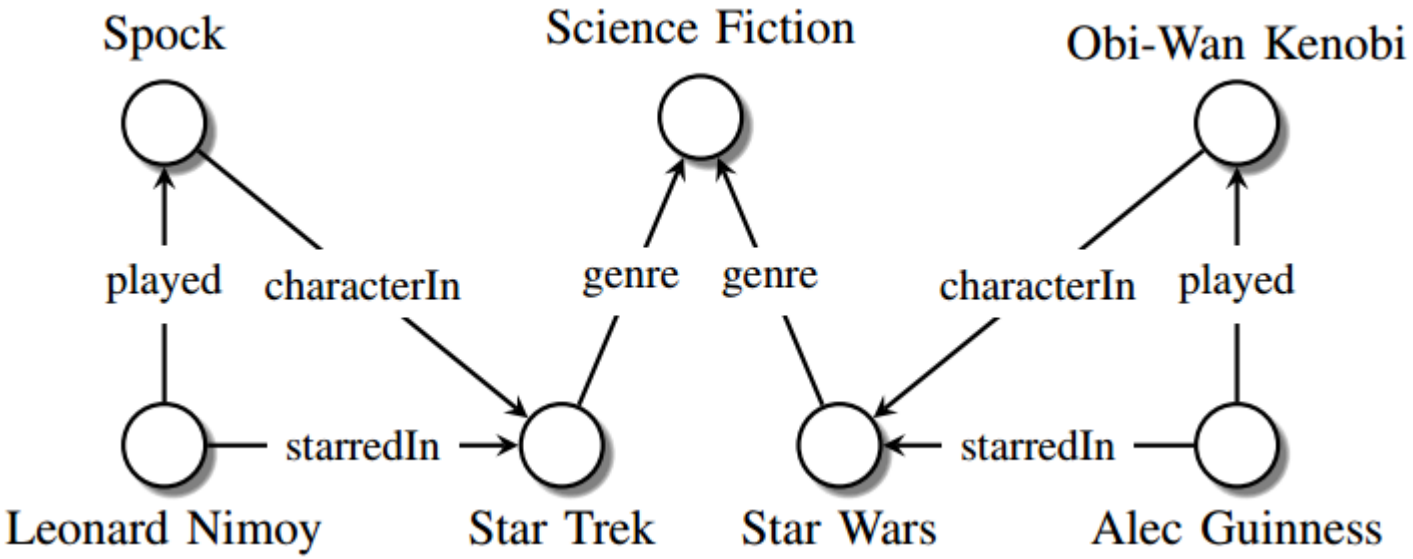
Farrukh Hasan

Email id: farrukh.hassan@nu.edu.pk

Agenda for the first few classes

- An overview of the whole course

Why?


Spock
Science Fiction
Obi-Wan Kenobi
played
characterIn
genre
genre
characterIn
played
starredIn
starredIn
Leonard Nimoy
Star Trek
Star Wars
Alec Guinness

Books

- Ian Sommerville – Software Engineering 10th Edition
- Roger Pressman – A practitioner's Approach 7th Edition

What is Success of a Software?

- When it meets the needs of the people who use it
- When it performs flawlessly over a long period of time
- When it is easy to modify
- Easier to use

- What do people use software for?

What happens if a software is not successful?

- Customer dissatisfaction
- Loss of financial assets
- Loss of human lives
- One may have to face avoidable legal consequences as you or your company could be sued for damages or injuries caused by bugs in you software.

Software Engineering: Definition

Software Engineering is a collection of techniques,

methodologies and tools that help

with the production of

- a high quality software system
- with a given budget
- before a given deadline

while change occurs.

- Software engineering exists as a discipline because much software fails to be delivered when expected or to perform as expected.
- The first step to controlling these problems is to understand them. Once the modes of failure are understood, the deficiencies in existing software can be addressed.
    - http://www.bernstein-plus-sons.com/.dowling/CSC175/SE_Failures.html

Scientist vs Engineer

- Computer Scientist
  - Proves theorems about algorithms, designs languages, defines knowledge representation schemes
  - Has infinite time…
- Engineer
  - Develops a solution for an application-specific problem for a client
  - Engineering is all about developing products, **using** well defined, scientific principles and methods.
  - Uses computers & languages, tools, techniques and methods
- Software Engineer
  - Works in multiple application domains
  - Has only 3 months...
  - …while changes occurs in requirements and available technology

- Mechanical Engineering is like looking for a black cat in a lighted room.
- Chemical Engineering is like looking for a black cat in a dark room.
- Software Engineering is like looking for a black cat in a dark room in which there is no cat.

http://sven-s.blogspot.com/2006/04/black-cats-and-software-engineering.html

- The Intangible nature of software causes problems for management in planning, estimating, scheduling and budgeting for accounting purposes.

Types of Software Products

- Generic products
  - Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
  - Examples – Word Processors, graphics programs, Information systems, ERPs
- Customized products
  - Software that is commissioned by a specific customer to meet their own needs.
  - Examples – embedded control systems, air traffic control software, NADRA software

- The difference between the FR and NFR between both

“Engineering”

- **Engineering** is the application of knowledge, typically in the form of science, mathematics, and empirical evidence, to the innovation, design, construction, operation and maintenance of structures, machines, materials, devices, systems, processes, and organizations.
- Practical use, economic value: These products need to help people do the things they need to do that customers actually want and are willing to pay real money for.
- Responsibility for correctness, suitability, and safety.

What is Software Engineering?

- Software engineering (SE) is the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software.
  - Wikipedia

- [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.
  - Fritz Bauer

- Software Engineering: (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).
  - IEEE

- One of the most pointless principles in the field of computer science. Entails nothing but common sense applied to programming, and generally takes up too much of your time to create stupid diagrams so people less intelligent than yourself can understand how it fits together.
  - Urbandictionary.com

- In short, Software engineering is a branch of computer science, which uses well defined engineering concepts required to produce efficient, durable, scalable, in budget, and on-time software products. It is concerned with all aspects of software production, from the early stages of system specification through to maintaining the system after it has gone into use.
- The first reference to the term is the 1968 NATO Software Engineering Conference and was meant to provoke thought regarding the perceived "software crisis" at the time.

- The Process
- Modeling
- Quality Management
- Project Management techniques

Software Crisis

- Software crisis was a term used in the early days of computing science.
- The term was used to describe the impact of rapid increases in computer power and the complexity of the problems that could be tackled.
- The focus of development and innovation was on hardware. Software was largely viewed as an afterthought.

- A study in United States (1979) found that:
  - 2% of software worked on delivery
  - 3% worked only after some corrections
  - 45% were delivered, but never successfully used
  - 20% were used, only after major modification/rework
  - 30% were paid for, but never completed/delivered

– Software Engineering
By: Jibitesh Mishra; Ashok Mohanty

Software Crisis (cont…)

- Systems cost exceeded the initial estimated budget
- Software over-ran delivery schedules
- Software had to be substantially modified
- Projects were unmanageable and code difficult to maintain.
- Software was never delivered.

  – These problems of software development were extensively felt in the beginning of the 1970s and are collectively referred to as the "software crisis".

www.nytimes.com/2015/08/11/opinion/bina-shah-pakistan-the-next-software-hub.html

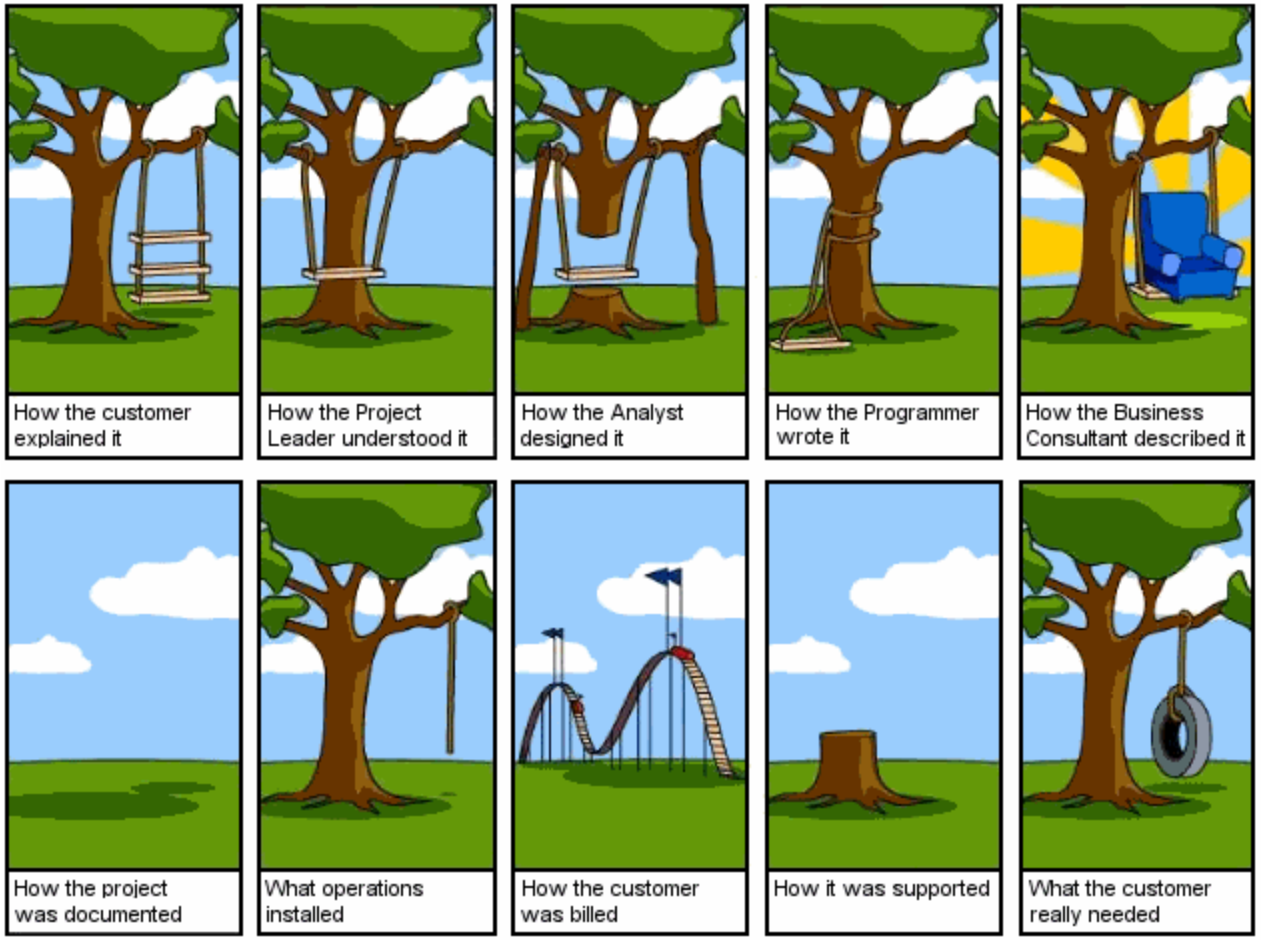
The New York Times

The Opinion Pages | CONTRIBUTING OP-ED WRITER

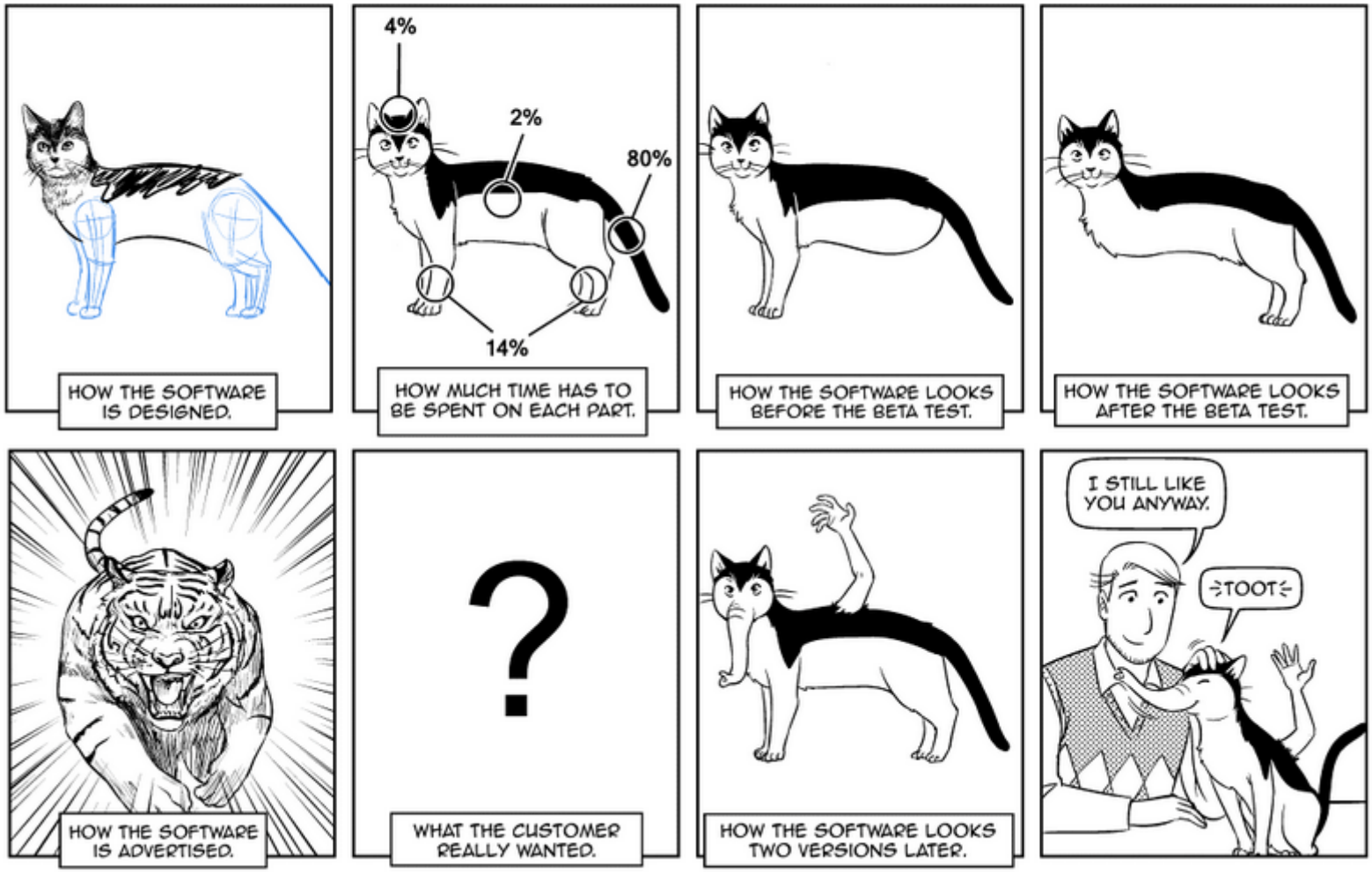
Pakistan, the Next Software Hub?

AUG. 10, 2015

Pakistan isn't usually considered one of the world's information technology powerhouses; its share of global I.T. sales is only $2.8 billion, of which $1.6 billion represents tech and I.T. services and software exported abroad. This is a tiny percentage of the expected $3.2 trillion global market for 2015, and is dwarfed by India's $100 billion worth of software exports per year.


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

Richard's guide to software development



Sandra and Woo by Oliver Knörzer (writer) and Powree (artist) – www.sandraandwoo.com

Software engineering code of ethics (ACM/IEEE-CS Joint Task Force on Software Engineering, 1999)

PREAMBLE

The short version of the code summarizes aspirations at a high level of abstraction. The clauses that are included in thefull version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.

Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

1 PUBLIC - Software engineers shall act consistently with the public interest.

2 CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer, consistent with the public interest.

3 PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

4 JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.

5 MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.

6 PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

7 COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.

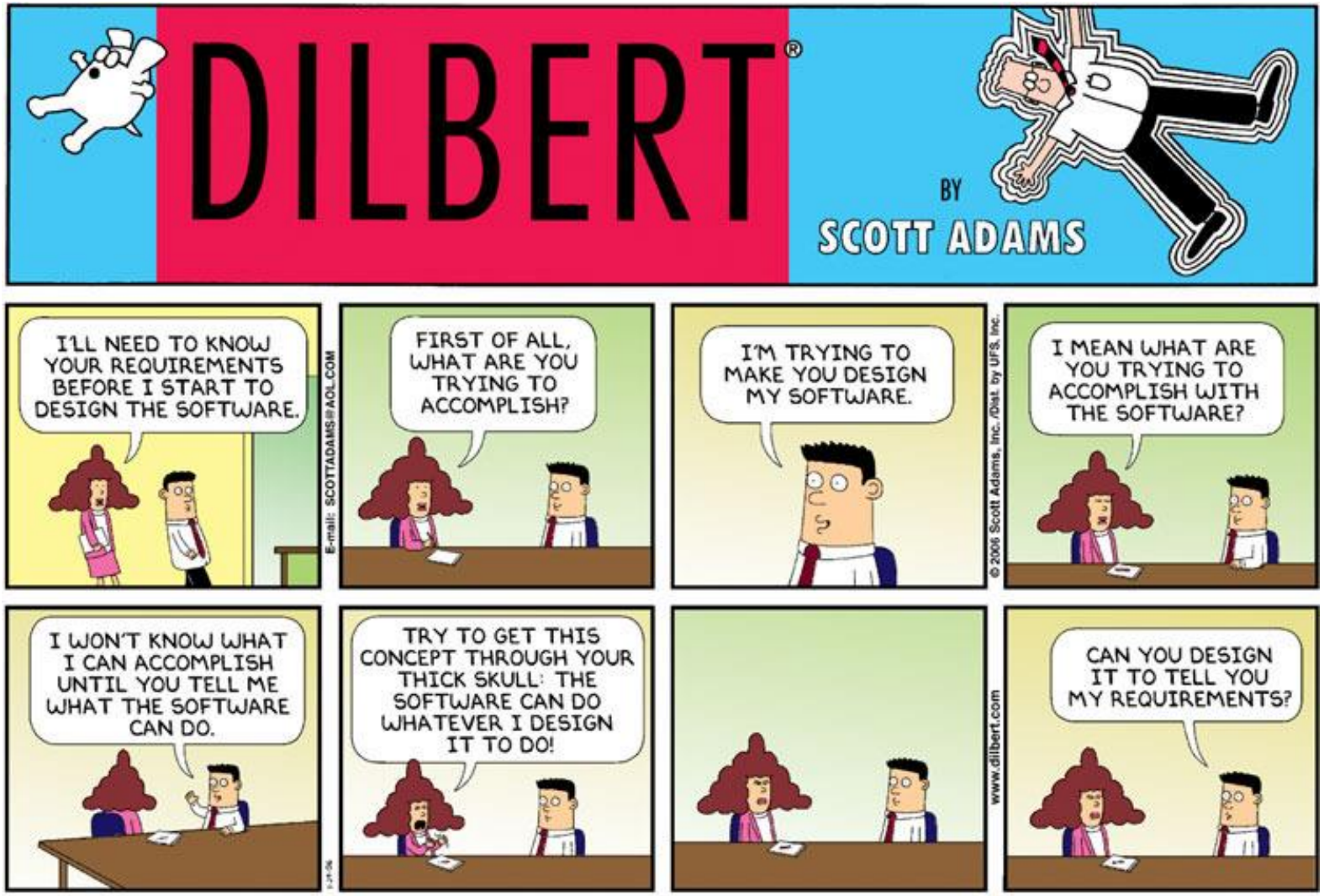
8 SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

The short form of the Code

- Mechanical Engineering is like looking for a black cat in a lighted room.
- Chemical Engineering is like looking for a black cat in a dark room.
- Software Engineering is like looking for a black cat in a dark room in which there is no cat.

http://sven-s.blogspot.com/2006/04/black-cats-and-software-engineering.html

Requirement gathering



Some famous software issues

- Year 2038 Bug: It is possible a problem (similar to Y2K) could occur in 2038 (the year 2038 problem), as many Unix systems calculate the time in seconds since 1 January 1970, and store this number as a 32-bit signed integer, for which the maximum possible value is 231 − 1 (2,147,483,647) seconds

- The Therac-25 was a radiation therapy machine produced by Atomic Energy of Canada Limited (AECL). It was involved in at least six accidents between 1985 and 1987, in which patients were given massive overdoses of radiation, approximately 100 times the intended dose.

  The accidents occurred when the high-power electron beam was activated instead of the intended low power beam, and without the beam spreader plate rotated into place. Previous models had hardware interlocks in place to prevent this, but Therac-25 had removed them, depending instead on software interlocks for safety.

Boeing 737 max

- “Investigators link both crashes, in part, to an automated flight control system that acted on erroneous information from malfunctioning sensors and put the planes into nose dives the pilots could not pull the planes out of.”
- https://www.npr.org/2019/06/26/736430419/faa-finds-new-problem-with-737-max-jets-delaying-their-return-to-flight

- Virtual Case File (or VCF) was a software application developed by the United States Federal Bureau of Investigation (FBI) between 2000 and 2005. The project was officially abandoned in April 2005, while still in development stage and cost the federal government nearly $170 million.

Reasons for failure [edit]

The project demonstrated a systematic failure of software engineering practices:[3]

- Lack of a strong technical architecture ("blueprint") from the outset led to poor architectural decisions
- Repeated changes in specification
- Repeated turnover of management, which contributed to the specification problem
- Micromanagement of software developers
- The inclusion of many FBI Personnel who had little or no formal training in computer science as managers and even engineers on the project
- Scope creep as requirements were continually added to the system even as it was falling behind schedule
- Code bloat due to changing specifications and scope creep—at one point it was estimated the software had over 700,000 lines of code.
- Planned use of a flash cutover deployment made it difficult to adopt the system until it was perfected.

Why so many issues?

- Requirement of flexibility
  - So flexible that start working with it before fully understanding what needs to be done.
  - Unknowns
  - Changes – policies, requirements, technologies.
- Intangibility/ Invisibility
  - Interfaces undecided
  - Transient hardware, software errors, race conditions, compatibility issues.
  - Visualization (E.g. The program for calculating landing for path finder mission)
- Conformity
  - Systems usually interacts with outside systems.

Issues with your projects?

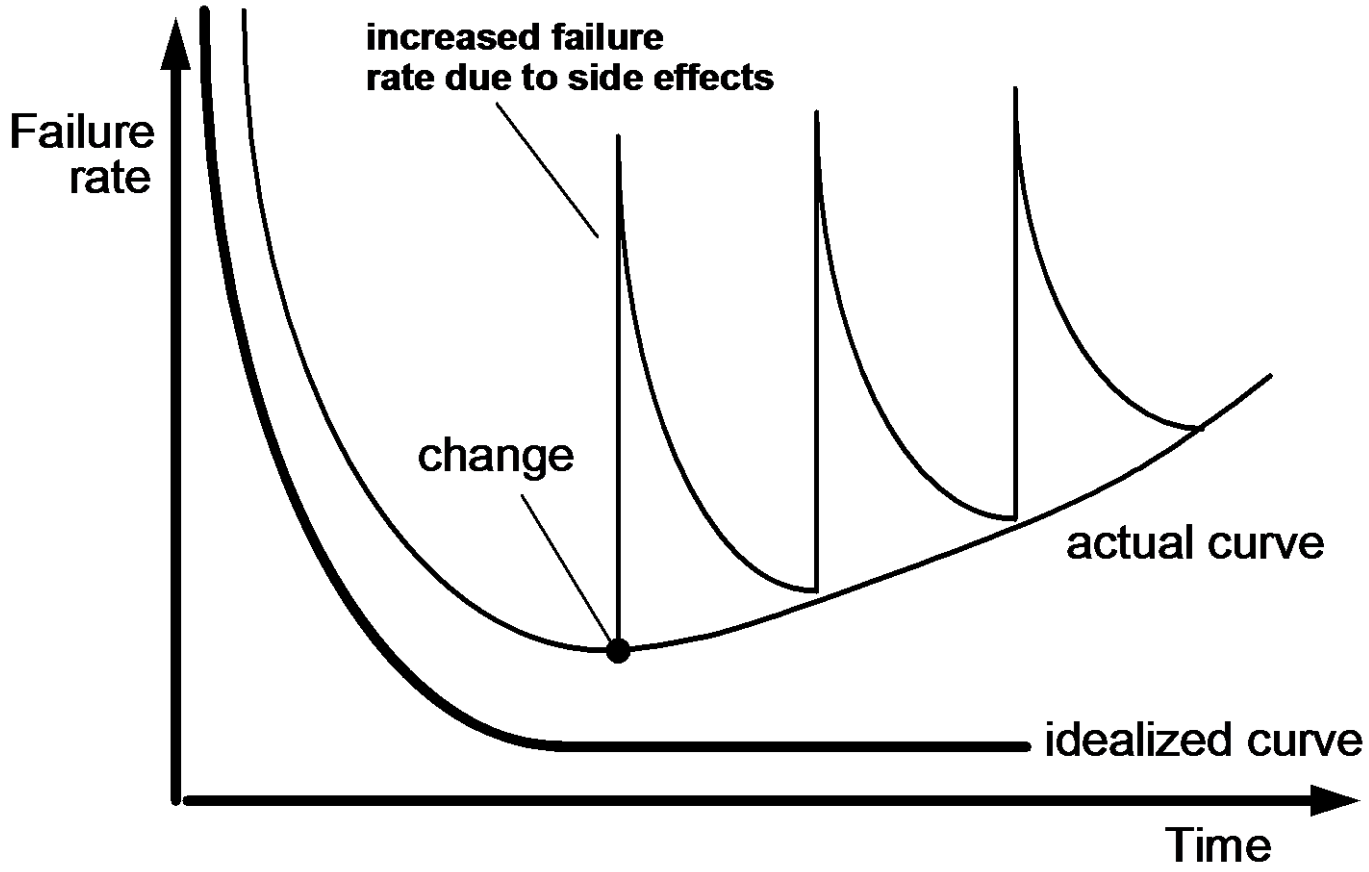
- What are the issues you have faced in your projects?

- Although the industry is moving toward component-based construction, most software continues to be custom-built.

Software – Some Categories

- Legacy Software
- Open world computing—pervasive, ubiquitous, distributed computing due to wireless networking. How to allow mobile devices, personal computer, enterprise system to communicate across vast network.
- Netsourcing—the Web as a computing engine. How to architect simple and sophisticated applications to target end-users worldwide.
- Open source—”free” source code open to the computing community (a blessing, but also a potential curse!)
- Data mining
- Grid computing
- Cognitive machines
- Software for nanotechnologies

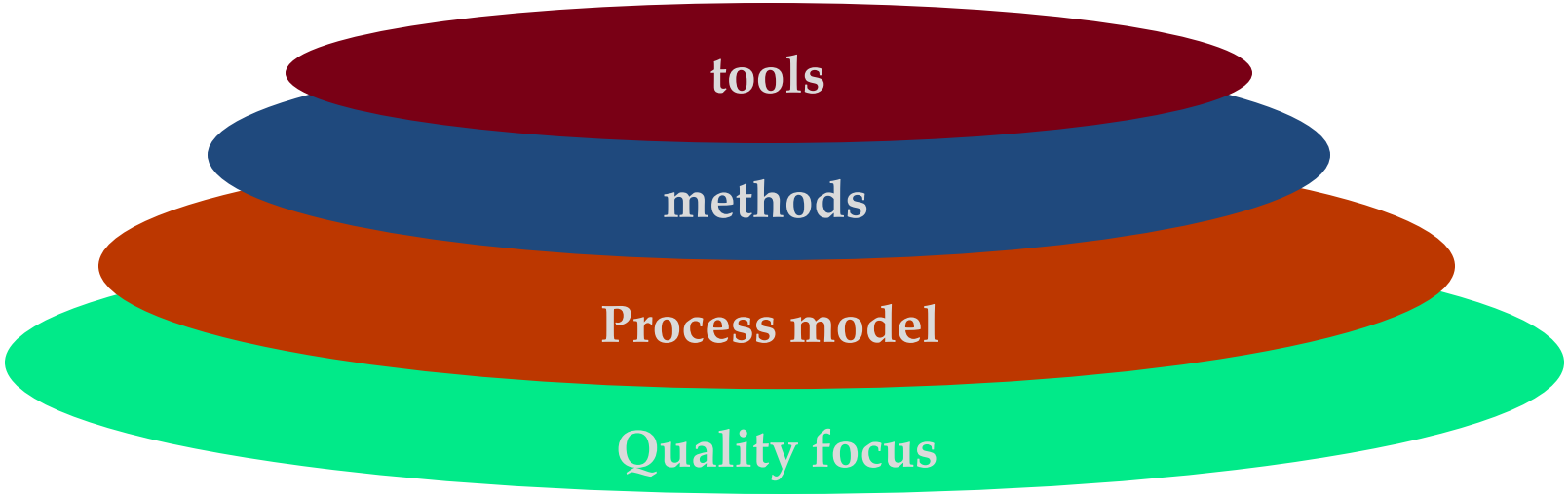
S.W. Deterioration



Some Realities

- a concerted effort should be made to understand the problem before a software solution is developed
- design becomes a pivotal activity
- software should exhibit high “quality”
- software should be “maintainable”

Software Engineering Layers



Course Description

Course Objectives

At the end of this course, students should:

- Have a sound understanding of the fundamental concepts of the software engineering paradigm.
- Understand and apply the different common practices used in software industry for the analysis, design and production of software.
- Analyze, design and implement practical systems of up to average complexity within a team.
- Become familiar with different tools used by industry in the software development process.
- Be prepared for interviews.

Tentative

Assignment	10
Quiz	5
Project	20
Mids	25
Final	40

- A process is a collection of activities, actions, and tasks that are performed when some work product is to be created. The software process forms the basis for management control of software projects and establishes the context in which technical methods are applied, work products (models, documents, data, reports, forms, etc.) are produced, milestones are established, quality is ensured, and change is properly managed.
- Software engineering methods provide the technical how-to's for building software. Methods encompass a broad array of tasks that include communication (e.g. interviews, questionnaires), requirements analysis, design modeling, program construction, testing, and support. Software engineering methods rely on a set of basic principles that govern each area of the technology and include modeling activities and other descriptive techniques.
- Software engineering tools provide automated or semiautomated support for the process and the methods.

The Essence of Software Engineering

- Understand the Problem (Communication and Analysis)
- Plan a solution (Modeling and Software Design)
- Carry out a plan (Code generation)
- Examine the result for accuracy (Testing and Quality Assurance)

Understanding the Problem

- Who has a stake in the solution to the problem? That is, who are the stakeholders?
- What are the unknowns? What data, functions, and features are required to properly solve the problem?
- Can the problem be compartmentalized? Is it possible to represent smaller problems that may be easier to understand?
- Can the problem be represented graphically? Can an analysis model be created?

Plan a Solution

- Have you seen similar problems before? Are there patterns that are recognizable in a potential solution? Is there existing software that implements the data, functions, and features that are required?
- Has a similar problem been solved? If so, are elements of the solution reusable?
- Can subproblems be defined? If so, are solutions readily apparent for the subproblems?
- Can you represent a solution in a manner that leads to effective implementation? Can a design model be created?

Carry Out the Plan

- Develop a solution.
- Does the solution conform to the plan? Is source code traceable to the design model?
- Is each component part of the solution provably correct? Have the design and code been reviewed, or better, have correctness proofs been applied to the algorithm?

Examine the Result

- Is it possible to test each component part of the solution? Has a reasonable testing strategy been implemented?
- Does the solution produce results that conform to the data, functions, and features that are required? Has the software been validated against all stakeholder requirements?

Types of Stakeholders

- Primary Stakeholders - usually internal stakeholders, are those that engage in economic transactions with the business. (For example stockholders, customers, suppliers, creditors, and employees)

- Secondary Stakeholders - usually external stakeholders, are those who - although they do not engage in direct economic exchange with the business - are affected by or can affect its actions. (For example the general public, communities, activist groups, business support groups, and the media)

-wikipedia

Stakeholders

- A person, group or organization that has interest or concern in an organization.
- Possible Stakeholders:
  - **Government** taxation, legislation, employment, legalities.
  - **Employees** job security, compensation
  - **Customers** value, quality, customer care
  - **Suppliers** providers of products and services used in the end product for the customer
  - **Creditors** credit score, new contracts, liquidity (banks?).
  - **Community** jobs, involvement, environmental protection, shares, truthful communication.
  - **Trade Unions** quality, worker protection, jobs.
  - **Owner(s)** profitability, longevity, market share, market standing, succession planning, raising capital, growth, social goals.
  - **Investors** return on investment, income.

FAQs

What is software?	***Computer programs, data structures and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.***
What are the attributes of good software?	***Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable.***
What is software engineering?	***Software engineering is an engineering discipline that is concerned with all aspects of software production.***
What is the difference between software engineering and computer science?	***Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.***
What is the difference between software engineering and system engineering?	***System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process.***