

\$440 Million



SPENCER PLATT/GETTY IMAGES

28 Soldiers Killed, 100 wounded



Software Systems Engineering

Why?

Software Engineering Review

CSCI5200 SSE

Jeff Roach

Plan

- Review Software and Software Engineering
 - You will:
 - Gain an understanding of software systems
 - Gain an understanding of software engineering
- Review the Process of Building Software
 - You will:
 - Gain an understanding of the software development process

Software System

- An assembly or combination of interrelated software components working together toward a common objective.

Software Systems Engineering

- It is concerned with the **processes, methods,** and **tools** for the development of large and complex software systems in an economic and timely manner.

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, 1993]

Discussion

- Why is Software Engineering important?

Software Development Process

- “A framework for the tasks that are required to build high-quality software.” [Pressman, 2005]
- It “defines the approach that is taken as software is engineered.” [Pressman, 2005]

Development Process Motivations

- Reducing costs while increasing quality of the product
- Software Criticality
- Enabling effective team work
- Anticipating and managing change
- Meeting deadlines



PROCESS QUALITY
PRODUCT QUALITY

Software Development Generic Phases

Phases

Problem Identification and Definition

Software Specification

Software Design and Implementation

Software Validation

Software Deployment and Evolution

AGILE SOFTWARE PROCESS



A group of software development methodologies based on iterative development, where solutions evolve through collaboration between self-organizing cross-functional teams.

Agile Software Manifesto

- <http://agilemanifesto.org/>
- “... while there is value in the items on the **right**, we value the items on the **left** more.”
 - **Individuals and interactions** over **processes and tools**
 - **Working software** over **comprehensive documentation**
 - **Customer collaboration** over **contract negotiation**
 - **Responding to change** over **following a plan**

General Principles of Software Engineering Practice

1. Determine the value of the system
2. KISS!
3. Maintain a clear vision
4. Always keep in mind that someone else will have to understand what you are doing and have done
5. Always keep in mind that systems change, so design accordingly
6. Plan ahead for reuse
7. Think ahead before acting

Management Advice

- Standards and procedures that work for one project may not work for another.
- Adding people to a late project will not complete it sooner.
- Outsourcing needs to be managed just as rigorously as an in-house project

Customer Advice

- Unambiguous requirements are derived only through effective and continuous communication among stakeholders.
- Requirements do change, but the cost of making that change increases rapidly as time passes.

Developer Advice

- The larger percent of effort expended will be after the software is delivered for the first time.
- Technical reviews are some of the most effective software quality assurance mechanisms.
- A variety of project artifacts provide a foundation for successful engineering and guidance for software support.