

Software Engineering

multi-person construction of multi-version software

Author: Daria Shutina

Software Engineering

23-02-02

Organization stuff

Failures and Catastrophes

How to Program?

23-02-07

Socio-technical systems

Emergent Properties /todo

Requirements Engineering

Schematic RE Procedure

23-02-02

Organization stuff

Course link: <https://peter-baumann.org/Courses/SoftwareEngineering/index.php>

Project:

A group of 4-5 people submit a design doc, then there are 2-week phases:

- Specification & design (teams of 5 – free to join up)
- Code sprints Implementation (teams of 2 – random)

Grading:

Software Engineering *lecture* (2.5 CP) = 33%

Software Engineering *project* (5 CP) = 67%

Failures and Catastrophes

<http://catless.ncl.ac.uk/Risks/>

Lessons from cases:

1. ***don't optimize to death.***

Y2K: "1997" in COBOL stored as "97", and "2000" -- as "00".

2. ***be careful with guessing user intent. Users need guidance.***

a bug in FORTRAN code: "DO 20 I = 1,100" and "DO 20 I = 1.100"

How to Program?

1. Write code
2. Test it on a few samples
3. Bug fixing, improving efficiency
4. GOTO 1

This is usually appropriate for 1-person projects. In real-life projects there are different people \Rightarrow different ideas through years. Also, if developer \neq user, there is a frequent dissent about expected vs. implemented functionality.

Common problems for SE projects are mostly about organization:

- Complexity of the idea
- Communication (b2b, b2c)
- Flexibility: change of requirements, components, methods, tools over lifetime
- Lack of education
- Bad project management

23-02-07

Socio-technical systems

Socio-technical systems are hardware & software + operational processes + people. It applies an understanding of the social structures, roles and rights to design systems that involve communities of people and technology. Examples of STSs include emails, blogs, and social media sites such as Facebook and Twitter.

STSs seek to merge people and technology, viewing the integration of computers into societal systems. It results in designing a complex system for the product, and properties of the system depend on components and their relationships. The computer will always do something, but it will do it correctly only in case it has been programmed for that situation.

System's behavior partially dependent on human operators and a time-varying environment. It does not always produce same output when presented with same input.

If you do not understand the organizational environment where a system is used, the system is less likely to meet the real needs of the business and its users.

Emergent Properties /todo

Emergent Properties are the result of various system components working together, they do not belong to an individual component.

- Volume: total space occupied depends on how component assemblies are arranged & connected.
- Reliability: unexpected interactions can cause new types of failure.
- Security: attacks not anticipated by system designers may defeat built-in safeguards.
- Repairability: in order to fix the problem, can the system be rebooted or turned off while being repaired?
- Usability: it is about how easy it is to use the system (example: presenting info in different languages).

Requirements Engineering

Requirements engineering is the process of defining, documenting, and maintaining requirements in the engineering design process. It is the process of eliciting the services that the customer requires from a system and the constraints under which it operates and is developed

Types of Requirements:

- User requirements: statements in natural language and diagrams of the system's services and its operational constraints.

- System requirements: a structured document with detailed descriptions of the system's functions, services, operational constraints and technical effects.

Schematic RE Procedure

1. Inception — ask questions
 - basic problem understanding
 - Identify stakeholders
 - Establish trustful communication
2. Elaboration — create analysis mode
3. Negotiation — determine each stakeholder's "win conditions", negotiate "win-win"
4. Requirements management — a set of techniques for documenting, analyzing, prioritizing, and agreeing on requirements
 - review mechanism (errors, missing info, feedback)
 - document versioning