

2008 Spring

Software Special Development 1

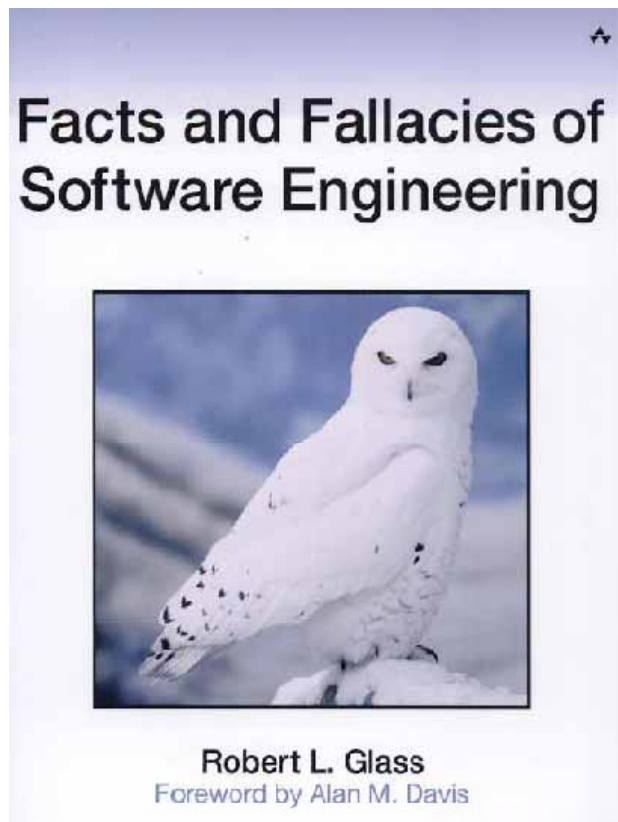
Discussion

- Facts and Fallacies of Software Engineering
(Part I)

Lecturer: JUNBEOM YOO
jbyoo@konkuk.ac.kr

Facts and Fallacies of Software Engineering

- 55 Facts & 10 Fallacies



Management

Fact
1

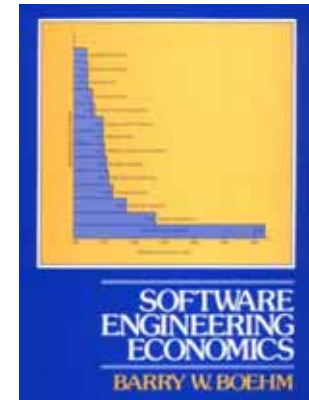
The most important fact in software work is the quality of the programmers.

Fact
5

Hype (about tools and techniques) is the plague on the house of software.

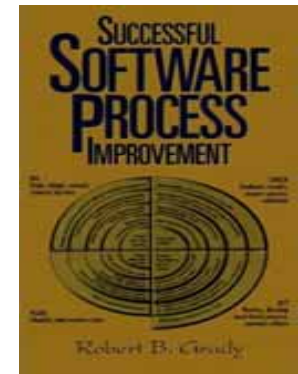
Fact 1: The most important fact in software work is the quality of the programmers.

- The most important factor in software work is not the tools, processes, methods, or techniques used by the programmers, but rather the quality of the programmers themselves.
- “The major problems of our work are not so much technological as sociological in nature.” (*Peopleware*, 1999)
- “The prime factor in affecting the reliability of software is in the selection, motivation, and management of the personnel who design and maintain it.” (Rubey, 1978)
- Question: “If your life depended on a particular piece of software, what would you want to know about it?”
 - Answer:
- Issues: “We don’t know how to identify the ‘best’ people.”



Fact 5: Hype (about tools and techniques) is the plague on the house of software.

- Exaggerated advertisements are the plague on the house of software. They claim that improvement in tools and technologies can increase productivity and quality of software up to 5~35%. But, the person who is superior to others by 10 times can also achieve the same improvement.
- It has been a long time since the last true breakthrough in 1950s.
- There is no silver bullets in this age.
- There is very little supporting evidence for most claims.
- The highest benefit from reuse in process improvement was 10 ~ 35%. (Grady, 1997)
- Question: Why we've kept believing exaggerated advertisements for a long time?



Requirements & Estimation

Fact
9

Software estimation usually occurs at the wrong time.

Fact
13

There is a disconnect between software management and their programmers.

Fact
26

Explicit requirements "explode" as implicit (design) requirements for a solution evolve.

Fact 9: Software estimation usually occurs at the wrong time.

- Estimation usually occurs before determining what the requirements are and understanding the problem itself, but no one found any problem until noticing this fact.
- The first phase of a project is to determine what the requirements are.
 - Why do we make estimates before we understand what the problem is that we are trying to solve?
- Question: “What causes run-away projects?”
 - Answer: unstable requirements and poor estimation.
- Marketing manager said to project manager, “You don’t have to understand ... we’ve already announced the release date.”
- Issues: How and When we can estimate the project well?

Fact 13: There is a disconnect between software management and their programmers

- In one research study of a project that failed to meet its estimates and was seen by its management as a failure, the technical participants saw it as the most successful project they had ever worked on.
- Many problems have their origin at the very beginning of project : ill-defined scope, unrealistic schedule, missing requirements
- Engineers can forecast that the die was cast on this project from day one.
- “Projects where no estimates were prepared at all fared best on productivity.”(Jeffery and Lawrence, 1985)
- “A very strong correlation between level of productivity and a feeling of control” (Landsbaum and Glass, 1992)
- Question: What do you think the most important factor from an aspect of management?

Fact 26: Explicit requirements "explode" as implicit (design) requirements for a solution evolve.

- When things gets messy from requirements to design, it's derived requirements grow factor of up to 50.
- Traceability from requirements to design, code, test cases, and all documents are considered desirable.
- But, traceability has proven to be an illusive Grail because of its complexity.
- Question:
 - How do we manage requirements when they are continuously being discovered?
 - What is the use of traceability?