

Historical Context

No Silver Bullet

- *Brooks*

Software's Chronic Crisis

- *Gibbs*

Thomas Fritz

Katja Kevic

Announcement

If you haven't done so:

- Sign up for moderating a discussion
- Sign up for paper reviews

Project Proposals

- One page due by **October 9th**

No Silver Bullet

Essence and Accidents of Software Engineering

I believe the hard part of building software to be the specification, design, and testing of this conceptual construct, not the labor of representing it and testing the fidelity of the representation.

Inherent properties: complexity, conformity, changeability and invisibility
Past breakthroughs (e.g. OOP, IDEs) and promising attacks (e.g. reuse, incremental development)

No Silver Bullet

- What did you think about the paper?
- What do you agree/disagree with?
- Do you think anything substantial has changed since 1987?
- ...

Building an application that allows a user to look at hiking trails and maps in the alps:

How can you take advantage of existing approaches/ techniques/... to tackle the **essential** and **accidental** difficulties?

Software's Chronic Crisis

What is the Software's Chronic Crisis?

Lack of maturity of software engineering discipline

Denver Int. Airport Baggage System Example

- Complex distributed system (3100 telecars, oversize cars, bar code readers, network computers,...)
- over budget, operating failure (~ \$1Mio per day)
- canceled in 2005

Failures

What are the challenges in developing software?

IBM Survey

- 55% of projects cost more than expected
- 68% took longer than estimated
- 88% of projects had to be redesigned substantially

Impact of Software Failures

- Affect on public safety (train control, flight monitoring, nuclear power plants,...)
- Big cost of failures (eg. Ariane 5)

Software Engineering Vision

Software Engineering is the application of a systematic, disciplined, quantifiable approaches to the development, operation and maintenance of software.

1968 NATO Science Committee

Questions

Any questions/comments on the paper?

Spend 15mins to answer the questions your group is assigned!

What can you do to build better software and avoid failures?

What does the author suggest?

What are you doing (employing) to build better software?

What have you done in companies / on projects?

What have you heard could help?

Why do you think certain techniques might help or not?

Formal Methods

Which formal methods does the author talk about?

Which areas are particularly suited for formal methods and why?

Do you think they are applicable to other areas too?

Which ones and why or why not are they applicable to certain areas?

Component-based Software Development

What are the problems/challenges and benefits according to the article and Brad Cox?

Why might component-based software help?

Have you used or know any third-party components before and what were the benefits and challenges?

How to better enable component sharing?

Off-shoring

How can off-shoring help?

What are problems according to the article?

What do you think of off-shoring?

What do you think are problems/challenges and benefits?

Why do you think it could work or might create more burden?

Quantitative Measurements of Software Quality

How can you quantitatively measure software quality?

Which aspects of software quality can be measured?

Do you know any tools to measure aspects of software quality and which ones?

What do you think are advantages and what are disadvantages of these approaches?

Do you already use any measures for your own software development?

How to measure a developer's productivity?

How to measure productivity?

What does the author talk about for measuring productivity?

What do you think might be a good approach and/or why do you think it (might) work or (might) not work?

How would you measure your productivity?

What are concerns with respect to measuring a developer's productivity?

No Silver Bullet and Chronic Crisis

- Do you think there is a silver bullet for the crisis?
- Why or why not?
- Is component reuse a silver bullet for all?

Response Papers

- Discuss BOTH articles (not just one)
- Do not just summarize or explain the article structure
- Use good/proper English sentences! The message as well as how it is conveyed is important!