

Problem Being Solved

CS846

Topics in Empirical Software
Evolution

No Silver Bullet by Frederick P.
Brooks, Jr.

Presented by Ahmed El Shatshat

- Software development is slow, cumbersome, and suffers from many systemic deficiencies
- There is much hope for a solution that will revolutionize software development
- However, this concept of a silver bullet is mythical, and impedes actual progress in software development practices
- Solutions touted as silver bullets are often revealed to provide some gains, but not at the order of magnitude one would expect from silver

New Idea

- The concept of a silver bullet solution for the pains of software development is infeasible
- This is demonstrated by taking a close look at what causes software development to be so costly
- In this analysis, it is proven empirically that the issues that impede software development are ingrained in the essence of the medium

New Idea

Complexity:

- Software is much more complex than their size would suggest
- Digital systems have a complexity an order of magnitude greater than most other systems

Conformity:

- Software systems must conform with human institutions and systems already in place
- Such institutions and systems often force software to conform to them, increasing difficulties

Changeability:

- Software changes far more frequently than other conventional fields
- This forces software into a complex cultural matrix that it must accommodate based on the context of the application

Invisibility:

- Software is extremely difficult to model, and the constructs we do have can only model a specific facet of software
- Such nebulosity makes it difficult to accurately convey ideas and concepts in an unambiguous way

New Idea

- Solutions such as high-level languages and unified programming environments have provided clear benefits, yet are empirically seen to not be silver
- To have a chance at a silver bullet, the essence of what makes software difficult must be addressed, not simply the technical aspects of the process

Book Chapter: Software analytics and its application in practice

- Identifying essential problems
- Building a usable system early for feedback

Negatives

Antiquated in Nature:

- The paper is nearly 40 years old
- Many ideas presented in the paper are now commonplace in software development

Untested Thoughts of a Single Author:

- Largely a collection of Brooks' musings on software development as he sees it
- Not much in the way of proof that what he's saying is necessarily correct

Positives

Revolutionary Presentation of Software and its Issues:

- Even from the larval stages of software development as we know it today, Brooks makes bare the essence of software development
- Any solution attacking contingent aspects of process or symptoms of that essence will not make the great gains imagined

Timeless in Message:

- Brooks expresses ideas that are still highly relevant today
- Many ideas he posits, despite having been demonstrated to be effective, are not being used as ubiquitously as they perhaps should

A Time Capsule

- It is interesting to see how the software field manifested in the mid 80s
- With the gift of hindsight, one can read the ideas Brooks has in the larval stages of many software tools and practices that today are commonplace, and perhaps reevaluate how they are being manifested today

Future Work

- What has changed over 40 years (since 1986) in terms of the guidelines outlined by Brooks
 - What was he correct about?
 - What did he miss?
 - What has been ignored?
- Addressing the identified essences of software development that are responsible for impeding development
 - Are there ways in which software can be structured in a way that is more resilient to change?
 - Despite the unvisualizability of software, can we develop structural models that are more in-line with the epistemological models software developers employ?
 - What aspects of software development are still victim to arbitrary conformity, and can we abrogate such conformity to streamline the development process without loss of quality?

Rating

5/5

This paper is visionary in what it strives to do, and the impact of the paper speaks for itself in terms of how effectively it achieved its goal

Discussion Points

- Which of the essences of software development outlined in Brooks' paper do you believe are the primary cause for issues and accidents in the software development process today?
- What recommendations by Brooks do you feel have been inadequately implemented by the software industry at large, despite their demonstrated or apparent usefulness?
- Are you a believer in a theoretical silver bullet that we simply haven't discovered, or do you believe that no silver bullet will be found, merely incrementally better solutions?