

Lecture 2

No Silver Bullet

About Background Survey

- Total 18 people responded. We know who did not fill out the survey. To redeem a half of your grade, fill out the survey by today.

About Background Survey

- <file:///Users/miryung/Documents/UT%20Course/EE382V-Spring2009/backgroundsurveyresult>

About Reading Assignments

- How to access the papers
- <http://users.ece.utexas.edu/~miryung/teaching/EE382V-Spring/readinglist.html>
- Due 11 PM instead of 8 PM
- **Write your name and UT EID in every review inside the review content**
- **No file attachment**
- Review length? Less than a page please.

About Class Presentations

- 52 slots - 22 * 2 = 8
- but currently 20 slots are available
- As I circulate the sign up sheet, check your presentation dates & papers

Today's Presenter

- No Silver Bullet: Jason Vanfickell (advocate) and Enos Jones (skeptic)
- On the criteria... by Parnas: David Pugh (advocate)

Today's Class Participation Points

- For the next five minutes, please fill out your peer feedback forms for Jason and Enos.

No Silver Bullet

- Essence and Accidents of Software Engineering
- What problem is F. Brooks addressing in the paper?

Why is software engineering difficult?

- Immaturity of SE
- Uniqueness of each software
- Requirements are always changing
- Organization, institutional knowledge is shifting
- Miscommunication, expectations are difficult
- Difficulty of anticipating evolution

Which of these are essential difficulties of building software?

- Immaturity of SE
- Uniqueness of each software - inherent
- Requirements are always changing - inherent
- Organization, institutional knowledge is shifting -accidental
- Miscommunication -A, expectations are difficult -I
- Difficulty of anticipating evolution inherent

**Which of these are essential
difficulties of building software?**

Which of these are essential difficulties of building software?

(I) Complexity

- no repeated regularities
- data sets, relationships among data items, algorithms, invocation of functions
- complexity leads to team development, in turn communication difficulties among team members

Which of these are essential difficulties of building software?

(2) Conformity

- People believe that they can do anything with software. Thus, software must conform to real world interfaces.

Which of these are essential difficulties of building software?

(3) Changeability

- Because software can be changed more easily, people try to change them often.
- Why does software changes so frequently?
-

Which of these are essential difficulties of building software?

- (4) Invisibility
 - Abstraction of software is difficult. No blueprint, models, etc.
 - Several graphs may represent the flow of control, the flow of data, patterns of dependency, time sequence, name-space relationships.
 - Inherently unvisualizable...

Attack on accidental difficulties

- High-level languages
 - Give me an example of high-level language support and its benefits:
 -

Attack on accidental difficulties

- Time sharing
 - Ability to compile and execute programs interactively greatly enhance productivity.
 - Does the same argument still hold?

Attack on accidental difficulties

- Unified programming environment
 - Toolbenches in which a new tool can be applied to any programs that used the standard format
 - What are some examples of a unified programming environment?

Hope for the Silver

- Ada
- Object-oriented programming
 - Hierarchical types
- Expert systems
 - Inference rule engines to assist programmers in software engineering tasks: e.g., interface rules, advise on testing strategies, remember bug-type frequencies, and offer optimization hints
 - MSR research
- Automatic Programming (Program generators)

Hope for the Silver

- Graphical Programming
- Program Verification
 - Need for complete and consistent specification
- Environment and tools
 - Language-specific smart editors
 - Integrated database systems to keep track of myriad details

Attacks on conceptual essence

- Buy components and reuse them.
- End user programming
- Incremental development and rapid prototyping
- Needs for educating *great* designers

Recap - Silver Bullet

- Take-away message:
 - As a researcher, we must distinguish inherent difficulties from accidental difficulties of software engineering.
 - To me, accidental difficulties are opportunities for advanced software engineering analysis and tools.
 - What is your take-away message?

Model literature survey papers & Tool Evaluation reports

- I uploaded links to literature survey papers written by graduate students as a part of literature survey project at the University of Washington.
- Same for tool evaluation reports

Before Going Home

- Find your project partners in the class today
- Take your graded reviews on Wednesday.
- Don't forget that your project option selection is due tomorrow at 8PM