Fred Brooks - Recap

- Difficulties are essential or accidental
- Since the accidental is (much) less than 9/10 of difficulties, must tackle the essence of software development to gain 10x improvement
- No technique will appear in next 10 years that gives 10x improvement
- Hard to say this has happened even in the last 20 years; perhaps object-oriented programming

David Parnas

- Seminal paper introducing information hiding: "On the Criteria to be Used in Decomposing Systems into Modules", *Communications of the ACM*, December 1972
- Advocates software engineering as an engineering discipline
- Currently, Professor of Software Engineering, University of Limerick, Ireland

David Parnas (2)

- No silver bullet, because:
  - People have a tendency to look for easy answers to hard questions, and designing software is hard
  - New technologies are often over-hyped (painted silver) in order to get people's attention
  - People want to seek better tools than actually learn the trade: "the poor workman blames his tools"

David Parnas (3)

- Much of the advances in software are due to hardware advances
- People ignore the "lead bullets" that are disciplined, hardworking, and need training to use

Linda Northrop

- Previously, Professor of Computer Science at Air Force Academy and SUNY; software developer at IBM and Eastman Kodak; private consultant
- Currently, Director of the Product Line Systems Program at CMU’s Software Engineering Institute
...software engineering involves more than programming...the hardest thing about building software is figuring out what to say, not how to say it.”

Focusing on essential complexity yields "breathtaking" results

Noted that Kristen Nygaard (co-inventor of OO in the form of Simula) said that we had “lost the essence, we were all excited about languages and doodads and frills” that had nothing to do with OO

Still need great designers and to cultivate an interdisciplinary perspective to understand the needs of users

Computational linguist

Pioneer of participatory design

Currently, Cisco Systems project manager

Progress has been made in that more people can create useful applications (e.g. websites) without being programmers

Architect of IBM Visual Age Smalltalk and Eclipse IDE

Founder and past CEO of Object Technology International Inc. (now IBM OTI Labs)

Currently, Founder and Chairman of Bedarra Research Labs

State of the art is a gratuitous disaster due to fast pace of change in APIs and frameworks creating immature software

Universities more concerned with certification than competence

Lessons from engineering and manufacturing for large projects

Breakthrough: recognition of the importance of testing

Real success in niches like airline reservations and hedge funds and highly specialized domain-specific and specialized language applications

Skills required are still “very, very high”
Ricardo Lopez

- Currently, principal engineer and manager, Qualcomm
- Main source of silver bullets is fear of failing
- But there is a silver bullet
  - Embrace complexity rather than avoid it for fear of failure
  - Share your excellence with your peers and become the silver bullet

Martin Fowler

- Pioneer of object-oriented technology, refactoring, patterns, agile methodologies, domain modeling, the UML, and XP
- Currently, Chief Scientist at ThoughtWorks, an IT consulting firm with main office in Chicago
- Werewolf in disguise

Martin Fowler / Werewolf (2)

- OO is a very dangerous idea, but nobody actually does it (especially outside OOPSLA)
- New accidents are waiting to happen like multicore concurrency systems
- Buying rather than building also is dangerous, but so far bad libraries are easy to build

Martin Fowler / Werewolf (3)

- Great designers could cause problems but no one is willing to make that case to the general public
- Waterfalls only give the illusion of control
- Desire for silver bullets creates more work

Question (1)

- Would a team of the greatest developers from all organizations become a silver bullet or a recipe for disaster?
  - Work with really good people to become good
  - No one tries to help people work together; ignore books like Peopleware
  - Good mixtures are encountered accidentally, cannot reproduce them repeatedly
  - Leadership skills matter

Question (2)

- How do you measure productivity?
  - What would an experiment to test for increase in productivity look like?
  - LOCs is not a good measure, perhaps completed user stories with weights
  - Which is better? 2 days to produce 2000 lines or 3 days to produce 500 lines, if both do the job correctly
Question (3)

- When will there be a proper development environment that allows developers to focus on the essence and not the accidents?
  - Barriers of communication between developers and others, especially business people
  - Perhaps the goal should be to get rid of cataclysmic code, avoid bad code, and accept pretty good code
  - Should not consider people who write complex code geniuses
  - Stop shipping incomplete frameworks

Conclusions

- Fowler/Werewolf: Even though progress has been made, humans always are overly optimistic about their capabilities, and problems appear unexpectedly
- Lopez: We will continue to make progress against increasing complexity; the werewolves will get bigger and uglier, but this one will pass
- Thomas: “A goal is important to have, a vision is important to have, not achieving doesn’t really matter if you make the journey”

Conclusions (2)

- Namioka: The search for silver bullets has created software systems that are used by people every day
- Northrop: Lesson learned: “focus on the essence, not the accidents”
- Parnas: We should not be looking for easy answers to hard questions

Conclusions (3)

- Brooks: “I know of no field of engineering where people do less study of each other’s work, where they do less study of precedents, where they do less study of classical models. I think that that is a very dangerous thing.”