

Fall 2018
Bruce F. Webster

CS 428
THE MYTHICAL
MAN-MONTH
Chapters 16-19

Ch 16: No Silver bullet – Essence and Accident in Software Engineering (1986)

- ◇ Probably one of the single most important essays ever written about IT
 - ◇ Core argument: “Building software will always be hard. There is inherently no silver bullet [to slay the monsters of software development].”
- ◇ Four inescapable essential difficulties in software development
 - ◇ Complexity: increases non-linearly with program size, both technically and managerially
 - ◇ Conformity: code must “work with” its ever-more-complex environment
 - ◇ Changeability: constant pressure to improve or fix existing systems
 - ◇ Invisibility: software is extremely hard to inspect and examine (vs., say, a building)

Ch 16: No Silver bullet (cont.)

- ◇ Things that do help
 - ◇ Buy vs. build
 - ◇ Buy and adapt (or adapt to) an existing solution that someone else had built and maintains
 - ◇ Requirements refinement and rapid prototyping
 - ◇ “...it is really impossible for clients, even those working with software engineers, to specify completely, precisely, and correctly the exact requirements of a modern software product before having built and tried some versions of the product they are specifying.”
 - ◇ Incremental development
 - ◇ “A large, complex system that works is inevitably found to have evolved from a small, simple system that works.” – John Gall, Infomatics
 - ◇ Great designers
 - ◇ “The very best designers produce structures that are faster, smaller, simpler, cleaner, and produced with less effort. . . . Those software systems that have excited passionate fans are the products of one or a few designing minds, great designers.”
- ◇ Analysis and observations?

Ch 17: “No Silver Bullet” Refired

- ◇ “I can’t help noticing that the nostrums published so vigorously in 1986 and 1987 have not had the dramatic effects claimed.”
- ◇ Brad Cox in 1990: “The reusable, interchangeable component approach [is] an attack on the conceptual essence of the problem.” This led to the ‘reuse’ push of the 1990s, which failed utterly.
- ◇ David Harel in 1992 offers “The Vanilla Framework”. Ever heard of it?
- ◇ Object-oriented development: also another brass slug (hence my book “Pitfalls of Object-Oriented Development” [1995])
- ◇ Brooks says his analysis stands; 30 years later, I agree with him.
- ◇ Analysis and observations?

Ch 18: Propositions of “The Mythical Man-Month”: True or False?

- ◇ Hint: this chapter is a great cheat-sheet for the open-book midterm

Ch 19: “The Mythical Man-Month” after 20 years

- ◇ Why has “The Mythical Man-Month” persisted? Again, me before Congress in 1998:
 - ◇ "Fred Brooks explored many of the root causes [of IT project failure] over twenty [now forty] years ago in *The Mythical Man-Month*, a classic book that could be regarded as the Bible of information technology because it is
 - ◇ universally known,
 - ◇ often quoted,
 - ◇ occasionally read,
 - ◇ and rarely heeded."

Ch 19: “The Mythical Man-Month” after 20 years (Cont.)

- ◇ Brooks sees his central argument not about scheduling or staffing, but rather about conceptual integrity and the need for a chief architect
- ◇ Second-system effect: define the set of users:
 - ◇ Who they are
 - ◇ What they need
 - ◇ What they think they need
 - ◇ What they want
 - ◇ “It is far better to be explicit and wrong than to be vague.” [Why?]
- ◇ Triumph of the WIMP interface, which Brooks sees as eventually being replaced by voice (I disagree)

Ch 19: “The Mythical Man-Month” after 20 years (Cont.)

- ◆ “Build one to throw away” – as we discussed, Brooks abandoned this in favor of iterative development – but most ‘waterfall’ is iterative these days as well
- ◆ Brooks acknowledges his fault in rejecting information hiding and now sees it as essential
- ◆ The mythical man-month: Boehm shows that “hardly any projects succeed in less than $\frac{3}{4}$ of the calculated optimal schedule, regardless of the number of people applied.”
- ◆ Brooks Law: yes, there are cases where adding people can help but “I stand by the bald statement as the best zeroeth-order approximation of the truth, a rule of thumb to warn managers against blindly making the instinctive fix to a late project.”

Ch 19: “The Mythical Man-Month” after 20 years (Cont.)

- ◆ People are everything (well, almost everything)
 - ◆ Cites Peopeware by DeMarco & Lister (your next book to read)
 - ◆ Boehm’s studies: “the quality of the team is by far the largest factor in its success, indeed four times more potent than the next largest factor.”
- ◆ The power of giving up power
 - ◆ Effective software management means building teams and letting them succeed
- ◆ The biggest surprises?
 - ◆ Millions [really billions] of computers
 - ◆ Massive amounts of shrinkwrap software (and now apps)
 - ◆ Note: he talk about 4G languages like Hypercard, which again have failed to pan out

Assignments for next class (10/8)

- ◇ By midnight on Saturday (10/6)
 - ◇ Create and post (on wiki) your team's initial requirements document draft
 - ◇ Submit your team's status report
 - ◇ Watch one podcast
- ◇ By start of next class period (2/5):
 - ◇ Peopleware (DeMarco & Lister), chapters 1-6
 - ◇ Webster #3 (online)