CS 428
THE MYTHICAL MAN-MONTH Chapters 4, 7, 11, 14

Winter 2019

Bruce F. Webster

Ch 4: Aristocracy, democracy, and system design

- Brooks: conceptual integrity is *the* most important consideration in system design (I agree)

- Simplicity, straightforwardness, unity of design are necessary

- The design must proceed from one mind or a very small number of agreeing resonant minds

- The conceptual integrity of a system determines its ease of use

- A consistent architecture enhances the creative style of implementers

- A well-thought-out architecture increases the robustness and adaptability of the resulting software system

- Your observations/experience?

Ch 7: Why Did the Tower of Babel Fail?

- What they did have:
  - A clear mission
  - Manpower
  - Materials
  - Time
  - Technology
- What they lacked?
  - Communication
  - And, as a consequence, organization
- Your observations/experience?

Ch 7: continued

- Project workbook: replaced today by online organization (e.g., your project wiki/repo)

- Communication challenge: with n workers on a project, there are $(n^2-n)/2$ possible interfaces and 2^n possible sets of workers

- Solution: Division of labor / specialization of function

- Key: project manager and chief architect need to be different people
  - Their priorities conflict
  - Chief architect will tend to be overly optimistic

Ch 11: Plan to throw one away

- As with "second system effect", Brooks feels his comments here are superseded by use of iterative/incremental software development

- Still, far too often, "pilot" or "prototype" systems are forced to evolve into production systems

- Only after your first cut do you often see how you should have done it in the first place

- What has been your observation/experience?

Ch 11: Continued

- Plan the organization for change
  - Still a very real issue: lack of technical advancement track in most organizations
  - Instead, developers are pushed into management if they want to be promoted
- Two steps forward and one step back
  - Most 'maintenance' work involved adding new features
  - Introduces software entropy (or, if you prefer, software rot)
  - Production systems that are modified become less stable/reliable over time
  - "Less effort is spent on fixing original design flaws; more is spent on fixing flaws introduced by earlier fixes"
- Your observations/experience?

Chapter 14: Hatching a Catastrophe

- "How does a project get to be a year late? One day at a time."
- Milestones must be concrete, specific, measurable events
  - The myth of the "Oh, we're about XX% done" statement
  - 90/90 rule: 90% of the project takes the first 90% of the schedule; the remaining 10% of the project takes the other 90% of the schedule.
- The "three weeks before deadline" rule:
  - "*Underestimates* [of project schedule] do not change significantly during the activity until about three weeks before the scheduled completion."
- Need for a critical-path schedule (e.g., PERT) to show the critical path
- Observations?

CH 14: Continued

- Not being willing to pass bad news uphill
  - Webster: The Thermocline of Truth (2008)
- Not knowing the news is bad
  - Webster: Lies, Damned Lines, and Metrics (parts I through III) (2008)
  - Project progress metrics need to be objective, repeatable, and informative
  - Weinberg's Law of Metrics: That which gets measured gets fudged.
  - The Metric Law of Least Resistance: "The more human effort required to calculate a metric, the less often (and less accurately) it will be calculated, until it is abandoned or ignored altogether."
- Thoughts and observations?

Assignments for next class (02/04)

- By midnight on Saturday (02/01)
  - Create and post on team wiki org chart **and** roles/responsibilities document
  - Create and post on team wiki first team status report, with billable hours
    - NOTE: make sure that you post this & subsequent reports so that all can be viewed
  - Watch one podcast (#2)
- By start of next class period (02/04):
  - Read *The Mythical Man-Month*, chapters 16-19
  - Webster #2 (online)