

# The Mythical Man Month Method (FourM) Software Project Management

Aron Trauring

T++ Technical Skills Training Program

CUNY Institute for Software Design & Development (CISDD)

New York Software Industry Association (NYSIA)

October 7th, 2004

## Fred Brooks - The Mythical Man Month

- PM of IBM's Operating System/360 (OS/360) in the early 1960's.
- Software development a human-centric process, not an engineering discipline.
- Why programming is hard to manage
- Why Projects Fail
- The Mythical Man Month Method (FourM)

## Human Centric

“The quality of the people on the project, and their organization and management, are much more important factors in success than are the tools they use or the technical approaches they take.”

## Complexity Model - I

- Program - two people in a garage
- Programming Product — run by anyone, in many environments with many data sets
  - Written in a generalized fashion
  - Thoroughly tested
  - Documentation/Maintenance
  - 3X

## Complexity Model - II

- Programming System — collection of interaction programs
  - Precisely defined interfaces
  - Stem Integration testing
  - 3X \* Programming Systems Product — 9X

# The Joy of Programming

- Joy of Creating
- Doing something useful
- Solving puzzles
- Constant learning
- Totally malleable medium

# The Tar Pit

- Must work perfectly
- Other people specify the product
- Other people's software
- De-bugging is tedious work
- Non-linear convergence
- Obsolete before its done

# The Mythical Man Month

More software has gone awry for lack of calendar time than for all other causes combined!

- Optimism
- People and Time are interchangeable
  - Sequential constraints - pregnancy
  - Added burden of training
  - Added burden of inter-communication

**Brook's Law:** "Adding people to a late software project makes it later!"

# The Tower Of Babel

- First engineering fiasco
- Communication
- Organization

“Lack of communication leads to disputes, bad feelings and group jealousies.”

## The Second System Effect

“The second is the most dangerous system a man ever designs...The general tendency is to over-design...using all the ideas and frills that were cautiously sidetracked on the first one. The result, as Ovid said, is 'a big pile.'”

- Featuritis

# One Step Forward and One Step Back

“System entropy rises over a lifetime.”

- Bit-Rot

## Under the Rug

“Disaster is due to termites not to tornadoes.”

- Fuzzy milestones
- The other piece is late anyway
- Hierarchical conflict

# Inherent Complexity of Programming Systems Products

“The essence of a software entity is a construct of interlocking concepts: data sets, relationships among data items, algorithms and invocations of functions... ...[T]he hard part of building software [is] the specification, design and testing of this conceptual construct, not the labor of representing it and testing the fidelity of the representation. ...Building software will always be hard.”

- Cognitive
- Social

## There is NO Silver Bullet

“There is no single development, in either technology or management technique, which by itself promises even one order of magnitude improvement in productivity, in reliability, in simplicity... Skepticism is not pessimism, however. Although we see no startling breakthroughs, and indeed believe such to be inconsistent with the nature of software, many encouraging innovations are underway. A disciplined, consistent effort to develop, propagate and exploit them should indeed yield an order of magnitude improvement. There is no royal road, but there is a road. “

## FourM - the Mythical Man Month Method

- Reduce Complexity
- Enhance Communication

## Project Design - Conceptual Integrity

“Conceptual Integrity is *the* most important consideration in system design.”

## Conceptual Integrity: What?

- Unity of Design
- Simple
- Straightforward

## Conceptual Integrity: Why?

- Maximizes ease of use
- Limits complexity

## Conceptual Integrity: How?

“The design must proceed from one mind or a small group of agreeing minds. Having a committee specify and design a system is a recipe for disaster. “

- User Products - User Community + Software Guru
- Tool Products - Software Guru is the user
- Object Orientation (OO) - Decompose the problem
- Object Orientation (OO) - Simplifies mapping of functional to technical

## Project Structure - The Surgical Team

“For each object or component of the system, assign a small surgical team to design and implement it.”

## The Surgical Team: System Team

- Producer - The person who gets things done: builds the teams, divides the work, manages the schedule.
- Director - The system architect, the person who creates the overall design.
- Stakeholders
- Administrative Support

## The Surgical Team: Component Team

- Research shows differences of 10x in programmer capabilities
- Remove the Tower of Babel
- Add a team not a person-month

## The Surgical Component Team: Chief Programmer

- Does the heavy lifting: design, coding, testing, documentation.
- No administrative role

# The Surgical Component Team: Programming Assistant

- CP's apprentice
- Sounding board
- Understudy
- Less arduous programming tasks

## The Surgical Component Team: Program Clerk

- All administrative chores
- Source code revision control
- System Builds
- Management reporting
- Future apprentice

# The Surgical Component Team: Support Functions

- May support multiple teams
- Toolsmith: constructs, updates, and maintains special tools needed by the team.
- Tester: develops systems tests and test data.
- Editor: puts the documentation into good shape.
- Resident hacker: finds the tricks to solve hard problems.
- Administrator: handles money, people, space, and machines, and interfaces with the Producer.

T++ — CISDD — NYSIA

Mythical Man Month Method - SPM

- Administrative Support

# Project Implementation - Iterative and Incremental Development (IID)

- Plan to Throw One^H^H^H-Many Away
- Changing understanding of construct by users
- Changing understanding of construct by developers
- Plan to discard and rebuild
- Deliver working code early and often

## IID: How?

- Tools: IDE, Application Servers, Very High Level Languages
- OO: Reuse of objects and components
- Techniques: Agile Methods

## IID: Prototyping

- Complex Reactive System
- Top Level Business System Specification

# Project Communications — The Documentary Hypothesis

“Amid a wash of paper, a small number of documents become the critical pivots around which every project s management revolves. These are the manager’s chief personal tools. “

## Documentation: 6 Key Documents

- What: Functional Specification — Objectives - needs, goals, objectives and priorities
- What: Technical Specification — Begins as proposal, ends up as documentation
- When: Schedule
- How Much: Budget
- Where: Space allocation
- Who: Personnel Chart - Surgical Teams

## Documentation: Communicate Frequently and in Every Direction

- Between users and developers
- Between managers and workers
- Between team members
- Between teams

# Documentation: Project Workbook

- Portals and Intranets
- Wikis
- Collectors

# Project Organization - Plan the Organization for Change

“There is nothing in this world constant but inconstancy.”

—Jonathan Swift

# Project Organization

- Easier to change software than an enterprise
- Agile Management