

7

Why Did the Tower of Babel Fail

Why Did the Tower of Babel Fail

- A Management Audit of the Babel Project
- Communication in the Large Programming Project
- The Project Workbook
- Organization in the Large Programming Project

A Management Audit

- Mission: Clear
- Manpower: Plenty
- Raw material: Abundant
- Time: No restriction
- Technology: Adequate
 - i.e., the project failed before reaching any limit
- Lacked communication, consequently
- Lacked organization

Communication in a large project

- Large projects necessarily involve many teams
- Many teams inevitably change the speed, functions and sizes of their programs
 - Meanwhile, other teams explicitly or implicitly make assumptions about the inputs to the program and the uses to be made of the outputs
- Sometimes, even a small change can have a disastrous effect on the other teams' programs

Communication channels

- Informal
 - Helps in common interpretation of documents
 - E.g. telephone, email
- Meetings
 - Regular project meetings
 - Each team gives a technical briefing
 - “Smoke out” fears; Refocus on priorities
- Workbook

The project workbook

- A structure for organizing the documents
 - All documents are part of the workbook
 - E.g., see preceding IEEE Std 1058 outline
- Includes objectives, external specifications, interface specifications, technical standards, internal specifications, administrative memoranda, quality-related documents, etc.
- Should be created very early in the project cycle

Advantages of workbook

- Structure
 - Initial design specifications often form the basis for future user documents
 - Having a structure in the early design phase assists later document preparation
- Distribution
 - Gets information to the people who need it
 - Indispensable for a large project spread across several physical and geographical locations

Mechanics of workbook

- Each programmer should see all the material
- Timely updating is critical
 - Replace changed pages
 - Clearly indicate changes, e.g., change bars
 - Maintain a revision history list
- Change summary may be kept in a LIFO fashion and the programmers would read the changes regularly

Modern workbooks

- Network filesystems
- Shared email folders (e.g., IMAP)
- Newsgroups and forums
- Web (publishing); DAV (editing)
- Source control systems, e.g., CVS, Subversion

ARPA network

- “Englebart and his colleagues at the Stanford Research Institute have built such a system and are using it to build and maintain documentation for the ARPA network.”
 - What did the ARPA network become?
 - Request for Comment (RFC) defines your world
 - Compare to Beck, “We craft a solution to today's problem today, trusting we will solve tomorrow's problem tomorrow.”

Internet Engineering Task Force

- Defines all Internet protocol standards via RFC's

0001 “Host Software” S. Crocker. Apr-07-1969

0008 “Functional specifications for the ARPA Network” G. Deloche, May-05-1969

0760 “DoD standard Internet Protocol” 1980

0761 “DoD standard Transmission Control Protocol” 1980

3994 “Indication of Message Composition for Instant Messaging.” H. Schulzrinne, Jan 2005

Workbook access policy

- Parnas' information hiding
 - Shields programmers from details except those they're working on
 - Assumes complete and precise interfaces
- Brooks (initially) disageed
 - Dependence upon perfect interfaces will lead to disaster
 - An effective information system exposes interface errors and leads to their correction

Parnas, Brooks, and open source

- Chapter 19, p. 271
 - “Parnas was right. I was wrong.”
- Mills persuaded Brooks that a public process improves quality through peer pressure
- Parnas' information hiding led to
 - Abstract data types
 - Object-oriented programming
- Eric Raymond cites open source development, “Given enough eyeballs, all bugs are shallow.”

Organization in a Large Project

- Large number of interfaces and teams
 - With n workers, there are potentially
 - $(n^2 - n) / 2$ interfaces for communication
 - 2^n teams to co-ordinate
- Purpose of organization is to reduce the amount of communication and co-ordination, through
 - Division of labor
 - Specialization of function

Tree organization

- Structure delineates authority and responsibility
- “Cannot serve two masters” implies tree structure but...
- Real communication is not restricted to a tree
- A tree's inadequacies give rise to
 - Staff groups, task forces, committees, gossip, etc.

Tree organization essentials

- Mission
- Producer
- Technical director or architect
- Schedule
- Division of labor
- Interface definitions among parts

The producer

- Acquires resources
- Assembles team, divides work, and establishes schedule
- Communicates upwards and sideways
- Defines communication and reporting inside team
- Ensures that the schedule is met
- Shifts resources and structure according to need

The technical director

- Defines the external specifications, conceives the design, identifies subparts, and outlines the internal structure
- Ensures conceptual integrity
 - I.e., limits the system complexity
- Handles individual technical problems at the design level
 - Proposes solutions and/or
 - Changes the system design

Producer and Technical Director, as the same person

- Effective and efficient for smaller teams
- Not suitable for larger projects
- Discouraged by difficulty of finding both management and technical skills in one person
- Can increase project risk by putting the controller on the technical critical path
- Larger projects have full-time jobs for each of the Producer and Director, so sharing is not efficient

Producer as Boss and Technical Director as Right Hand

- Difficulty in establishing technical director's authority to make technical decisions
 - Impacts his time as he is involved management chain-of-command
- Producer should *strongly* support director's authority
- Producer should have high respect for director's technical skills

Producer as Boss and Technical Director as Right Hand (cont.)

- Producer and director should discuss and clarify the technical issues so that they can have a common stand
- Producer can subtly express the authority of technical director by using symbols of status
 - E.g. office size, furnishing etc.
- In most cases, the technical talent of the technical director remains under-utilized

Technical Director as Boss and Producer as Right Hand

- Technical director retains the *ultimate* authority
- Mundane administrative activities are delegated to the producer
- Technical director focuses on the technical issues
- Technical director may keep track of the activities being delegated to the producer
- Brooks suggests that this arrangement is suitable for small teams and producer as boss is more suitable for larger projects

Open source development structure

- According to Andreas Brand, sociologist
- KDE is a typical open source project
- Core developers surrounded by peripheral
- Peripheral technical documentation, translation, and release coordination
- The most important participants form an inner circle and work on sub-projects
- From <http://programming.newsforge.com/article.pl?sid=05/01/25/1859253>

Open source development (cont.)

- Decisions are made differently in every project
- Linux has a benevolent dictatorship
- Debian a "democratic" voting system of members, comparable to the Roman senate
- KDE has no formal decision structure, but a special mailing list (kde-core-devel)
 - For development decisions
 - The predecessor should appoint a successor
 - Kind of cooptation