Why Software Projects Fail

Dr. Randall Jensen
Les Dupaix
Software Technology Support Center
Hill AFB, UT
Project success rate

CHAOS 2004 Software Project Survey

- Success: 29%
- Failed: 18%
- Challenged: 53%
CHAOS Success rate definitions

- **Success**: Produced acceptable results delivered close to on-time and on-budget
- **Challenged**: Delivered software product significantly over budget and/or schedule
- **Failed**: Failed to deliver any usable result within budget or schedule constraints
Software problems are not new

- Unreliable
- Late delivery
- Modification costs prohibitive
- Impossible to maintain
- Inadequate performance
- Product exceeds budget costs

1968 CS Conference, Munich, Germany
Reasons for project failure

- ADVERTISED
  - Immature technology
  - Inadequate requirements
  - Insufficient developer experience

- COVERT
  - Poor estimating and planning
  - Hope (Pandora’s paradox)
Repeattable things

- Development environment
- Productivity
- Minimum development time
  - Effective product size
  - Complexity
  - Paul Masson rule

Let’s look at some proof
BIG productivity drivers

- Analyst capability
  - Management style
  - Motivation
  - Problem solving skills
  - Use of team methods
  - Working environment
- Application domain experience
- Automated tool support
- Programmer capability
- Use of modern practices
Capability shift (environment)

Basic Technology Constant

5500 6200 6500 7500

1980 2005

8635 (1990)
1996/2003 Productivity for Avionics and Unmanned Space

The graph illustrates the productivity progression of software development from 1996 to 2003. The data is based on 52 Mil-Spec Avionics & Unmanned Space Software Data Records for 1996 and 51 similar data records for 2003. The productivity has significantly increased, and the 50th percentile for both years shows a significant improvement.

Aerospace Corp – Long et al. 2004

1996 graph based on 52 Mil-Spec Avionics & Unmanned Space Software Data Records
2003 graph based on 51 Mil-Spec Avionics & Unmanned Space Software Data Records
1996/2003 Productivity Distributions for Ground and Mobile

...progressed at LESS THAN this rate in lines/developer-months

1996 graph based on 112 Military Ground & Military Mobile Software Data Records
2003 graph based on 135 Military Ground & Military Mobile Software Data Records

Aerospace, Long et al, 2004
Common technology claim

If you use (*new technology*),

Productivity will improve by an *order of magnitude*

And

Defects will reduce to *zero*
There is always HOPE

Structured Analysis
Structured Design
Ada
OOD
Process Maturity
PWB
Structured Programming
3rd Generation Languages


There is always HOPE
Where would you focus effort?

Source: G. Weinberg, Quality Software Management, Vol. 3
Relative payoff

Source: G. Weinberg, Quality Software Management, Vol. 3
Repeatable things

- Development environment
- Productivity
- Minimum development time
  - Effective product size
  - Complexity
  - Paul Masson rule

Let’s look at more proof
Historic project data

Source: Long, L. G. et al, 2004
Historic space project limits

Size (kloc) vs. Time (years)

Successful Completions

Source: Long, L. G. et al, 2004

BE AMERICA’S BEST
Three development variables

- Cost
- Schedule
- Scope

Developer can control any two
“More software projects have gone awry for lack of calendar time than for all other causes combined…”

F. P. Brooks, Jr., Mythical Man Month
Common estimate dilemma

THE ORIGINAL SCHEDULE LOOKED LIKE THIS...

ONE MONTH FOR A MANAGEMENT DECISION AND ONE YEAR TO DO THE PROJECT.

THE REVISED SCHEDULE IS THIS...

ONE YEAR OF INDECISION FOLLOWED BY INTENSE PRESSURE TO DO THE IMPOSSIBLE BEFORE THE DEADLINE.

NOW IF YOU'LL EACH TAKE A PAIR OF 3-D GLASSES...

YOU CAN SEE THE LAYERS OF MANAGEMENT INCOMPETENCE PRACTICALLY JUMP OUT AT YOU.

NOW SCRATCH ONE OF THESE SCENTED CARDS TO SNIFF THE UNMISTAKABLE ODOR OF DOOM.

I DON'T SMELL ANYTHING. IS MINE BROKEN?

© 2002 United Feature Syndicate, Inc.
Constraint analysis

- Success and failure determined by
- Expected cost and schedule determined by
- Project plan determined by
- Cost and schedule ESTIMATES determined by
- Managers and/or Estimators
Key estimate factors

Management

Development

Personnel

Process

Environment

Technology

Size

Productivity

Cost

Schedule

Start Here
Elements for successful estimates

- Basic understanding of the requirements
- Ability to accurately size the deliverable product
- Assessment of the deliverable complexity
- Profile of the organization’s delivery capability
Without well thought out estimates, there is no firm basis for:

- Defining costs and schedules
- Making tradeoffs
- Associating development costs with the benefits
- Conforming to a budget or schedule

**THE PROJECT IS ALREADY OUT OF CONTROL!**
Project success rate

CHAOS 2004 Software Project Survey

- Success: 29%
- Failed: 18%
- Challenged: 53%
Success definitions: 2nd look

- **Success**
  - Produced acceptable results delivered close to **on-time** and on-budget

- **Challenged**
  - Delivered software product significantly **over budget** and/or schedule

- **Failed**
  - Failed to deliver any usable result **within budget or schedule constraints**
Conclusions

- Software project failures are primarily due to failed expectations
- Technology, requirements and experience are largely accounted for in resource estimates
- Failure ranking:
  1. Poor estimating and planning
  2. Hope (Not a management strategy)
  3. Immature technology
  4. Inadequate requirements
  5. Insufficient developer capability (Ebonians)
New estimating models?

- Lack of confidence in existing tools
  - Optimistic estimates
  - Unacceptable estimates
    - Skill
    - Experience
    - Integrity
- Aging tools
  - Quality data
  - Culture is constant
  - New models require validation
- Silver bullet
- New estimators?
Historic note:

We learn from experience that we don’t learn from experience.

D. H. Lawrence
The End, or is it The Beginning?
(History does repeat itself)