Software Failures

Dr. James A. Bednar

jbednar@inf.ed.ac.uk

http://homepages.inf.ed.ac.uk/jbednar

Dr. David Robertson

dr@inf.ed.ac.uk

http://www.inf.ed.ac.uk/ssp/members/dave.htm

How Software Projects Fail

Software appears, by its nature, to be difficult to engineer on a large scale. Nevertheless, there is an insatiable demand for sizeable, well-engineered software.

We continue to be dogged by large numbers of project failures, on small and large projects. Many (most?) of these are due to mistakes in project management.

In this lecture we discuss:

- Examples of project failure on a large scale
- Lessons that can be learned

Scale of the Problem (1)

Quoted from CIO Magazine Dec 1998:

- Recent Standish Group survey indicates “46% of IT projects were over budget and overdue and 28 % failed altogether”
- “only 24 % of IT projects undertaken by Fortune 500 companies in 1998 will be completed successfully”

From 1994 Standish report:

- 91% of projects at large companies failed
- 30% of projects at large companies were eventually cancelled

Scale of the Problem (2)

A source in the US military claims:

- There is one Army project that is 1 billion dollars over budget, and still has no working system.
- 100% of all DoD projects over 1 million lines of code are delayed.
- One third of all projects are cancelled before completion.
- One half of all projects cost twice as much as originally estimated.
- Documentation of a 350 KLOC DoD project costs about four hundred dollars per page.

FBI Virtual Case File (1)

The US Federal Bureau of Investigation has often been criticized for not sharing leads between agents and divisions.

Just before the 2001 terrorist attacks, the FBI hired Science Applications International Corp (SAIC) to develop Virtual Case File software (VCF).

VCF was designed to manage case files electronically, so that any agent with suitable permissions can find relevant information.

Originally scheduled for completion in 2003.

FBI Virtual Case File (2)

After repeated delays, a version was delivered in December 2004, but:

- Was about one tenth of originally promised
- May be scrapped altogether
- Does not approach functionality of existing commercial packages
- Might be used as a prototype, if anything
- Apparently $170 million wasted

FBI Virtual Case File (3)

Apparent causes:

- Changing requirements (after the September 11 attacks)
- Ambitious project, run as an emergency fix
- 14 different managers over project lifetime
- Poor oversight of external contractor
- Not paying attention to new, better commercial products
- Hardware purchased already; waiting on software

License Registration System (1)

In 1990 the Washington State Department of Licensing launched its License Application Mitigation Project (LAMP): $41.8 million over 5 years to automate the state's vehicle registration and license renewal processes.

By 1993 budget was $51 million and system was expected to be obsolete when finished.

In 1997 the plug was pulled, about $40 million having been wasted.

License Registration System (2)

Problems:

- Too big in concept with too few early deliverables
- Split between in-house and contractor development
- The organization didn’t want to hear that the project was a failure

Customer Database System

In 1996 a US consumer group embarked on an 18-month, $1 million project to replace its customer database. The new system was delivered on time but didn't work as promised, handling routine transactions smoothly but tripping over more complex ones.

Within three weeks the database was shut down, transactions were processed by hand and a new team was brought in to rebuild the system. Problems:

- The design team was over-optimistic in agreeing to requirements
- Developers became fixated on deadlines, ignoring errors

Customer Tracking System (1)

In 1996 a San Francisco bank was poised to roll out an application for tracking customer calls. Reports provided by the new system would be going directly to the president of the bank and board of directors. An initial product demo seemed sluggish, but telephone banking division managers were assured by the designers that all was well.

But the the system crashed constantly, could not support multiple users at once and did not meet the bank's security requirements. After three months the project was killed; resulting in a loss of approximately $200,000 in staff time and consulting fees.

Customer Tracking System (2)

Problems:

- The bank failed to check the quality of its contractors
- Complicated reporting structure with no clear chain of command
- Nobody “owned” the software

Payroll system (1)

The night before the launch of a new payroll system in a major US health-care organization, project managers hit problems. During a sample run, the off-the-shelf package began producing cheques for negative amounts, for sums larger than the top executive's annual take-home pay, *etc*.

Payroll was delivered on time for most employees but the incident damaged the relationship between information systems and the payroll and finance departments, and the programming manager resigned in disgrace.

Payroll system (2)

Problems:

- The new system had not been tested under realistic conditions
- Differences between old and new systems had not been explained (so $8.0 per hour was entered as $800 per hour)
- “A lack of clear leadership was a problem from the beginning”

Distribution System (1)

Anticipating growth, a $100 million division of a $740 million manufacturing business earmarks $5 million for a new distribution and customer service system to replace its old one. The project is to take a year and a half to complete. Two years later, the CIO is sacked and a new executive brought in to save the project. Three months later, the system breaks down altogether.

Nine months later, the CIO approached his boss, the CEO to tell him the project is a failure. “It was kind of like telling him a relative had died,” he recalls. “First he denied it, then he went through a grieving process, then he accepted it. It was just so much money for a division that size to wave in the wind.”

Distribution System (2)

Problems:

- Wrong direction from the start
- Inadequate software plan
- Nobody “owned” the software

Critical Failure Factors (1)

Warning signs of a project doomed to failure, or even disaster, from Flowers (1996):

- Organization: hostile culture, poor reporting structures
- Management: over-commitment, political pressures
- Conduct of the project:
  - Initiation phase: technology focused, lure of leading edge, complexity underestimated

Critical Failure Factors (2)

- Conduct of the project (continued):
  - Analysis and design phase: poor consultation, design by committee, technical fix for management problem, poor procurement
  - Development phase: Staff turnover, poor competency, poor communication (e.g. split sites)
  - Implementation phase: receding deadlines, inadequate testing, inadequate user training

Summary

- SW development is inherently a risky process
- Many projects fail for the same reasons
- Unfortunately, hindsight is much clearer than foresight, but
- The risk of failure should be addressed from the very start

Optional reading: Flowers (1996); Glass (1998)

References

Flowers, S. (1996). *Software Failure: Management Failure: Amazing Stories and Cautionary Tales*. Reading, MA: Addison-Wesley.

Glass, R. L. (1998). *Software Runaways: Lessons Learned from Massive Software Project Failures*. Englewood Cliffs, NJ: Prentice-Hall.