Why is software cost estimation **SO** difficult? (in particular for MOD projects)

Dave Thombs BSc PhD

Slimbridge Software Services Ltd
Beresford House
Churchend
Slimbridge
Gloucester GL2 7BL
Tel. 01453 890353
e-mail dave.thombs@ntlworld.com

V1.0

Disclaimer

The views expressed in this presentation are mine and do not necessarily represent the views of the Ministry of Defence.

D A Thombs

- Why is software cost estimation **SO** difficult?
- Because software is **SO** difficult!

Low (alleged) probability, high impact software acquisition risk

$7bn development cost + uninsured $500m loss for rocket and payload

Contents

- Background.
- Sources of acquisition risk.
- Risk reduction.
- Outstanding problems.

Contents

- Background.
- Sources of acquisition risk.
- Risk reduction.
- Outstanding problems.

Denver International Airport

Denver International Airport

The Denver New International Airport Baggage System

- Contractor - BAe Automated Systems
- Major System Software Problems
  - Loss of Destination Information
  - Peak Time Glitch - Sends Outbound Baggage to Inbound Carousel
  - Failure to generate reports

Denver International Airport

IMPACT OF DELAYED BAGGAGE SYSTEM

- Cost overrun £16.8M
- United Airlines additional spend £34.3M
- Other Airlines additional spend £21.8M
- Install old manual system £31.8M

Denver International Airport

IMPACT OF DELAYED BAGGAGE SYSTEM

Operational delays	£225M
Airport lost income	£23.1M
Loan fees	£5M
Ticket prices increase	£12.50
Total impact cost	£358M

How to miss Venus

- During the American space program, an unmanned vehicle was sent to look as the planet Venus. Part way on its journey a course correction proved necessary. A computer at mission control executed the following statement:-

  DO 3 I = 1.3 (A perfectly valid Fortran Statement)

- However, this was intended to be a DO (Iteration) statement whereas it is in fact an assignment statement. The difference is that (in the DO statement) the full stop should be a comma.

  DO 3 I = 1 , 3

- The space probe turned on the wrong course and was never been seen again.

- NOTE! This program had been compiled and tested successfully.

Microsoft Windows XP

- Microsoft released Windows XP on Oct. 25, 2001. (45 M SLOC)
- Same day posted 18 megabytes of patches on its Web site.
- Two patches fixed important security holes. One worked; the other didn't.
- Microsoft advised users to back up critical files before installing the patches.
- However no way to restore backup files if things went wrong.

Background

- Comments on software dependent projects:
- "Achieving sufficient reliability in systems which are becoming increasingly integrated into the central activities of modern society."
- "The difficulties of meeting schedules and specifications on large software projects,"
- "The education of software engineers."
- "It is of the utmost importance that all those responsible for large projects involving computers should take care to avoid making demands on software that go far beyond the present state of technology unless the very considerable risks involved can be tolerated."
- "Define a subset of the system which is small enough to bring to an operational state within 12 months, then build on that subsystem."
- "If the military software problem is real, it is not perceived as urgent by most high military officers and DoD civilian officials. Our Task Force does not undertake to prove that it is urgent; we do tell how to attack it if one wants to."
- "We know why projects fail, we know how to prevent their failure - so why do they still fail?"
- "In no other discipline is the gulf between best practice and typical practice so wide"

Background

- Comments on software dependent projects:

NATO Conference, Garmisch, Oct 1968

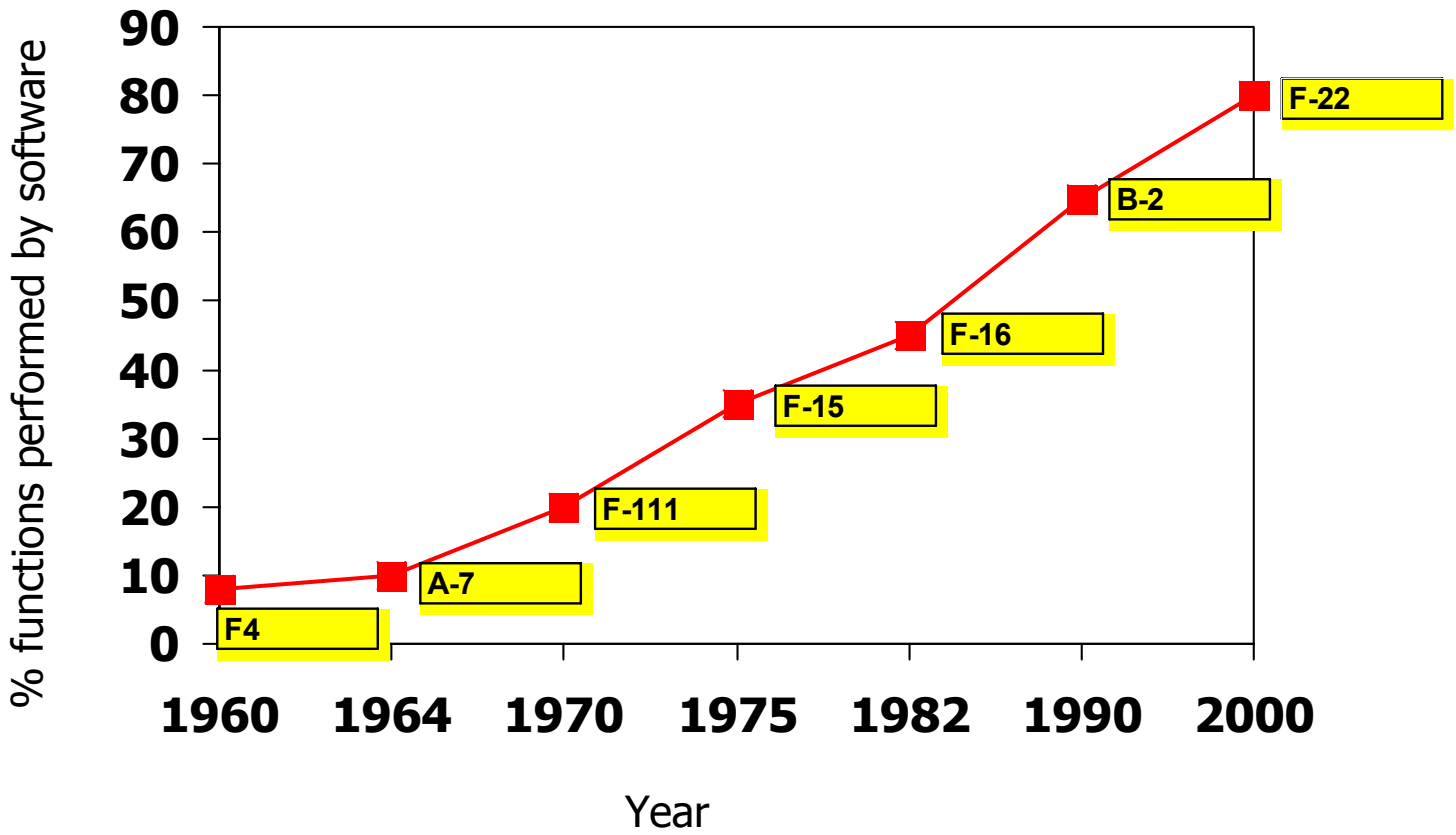
Defense Science Board Task Force on Defense Software, November 2000

The challenges of complex IT Projects, RAE/BCS, April 2004

USA private and public sector

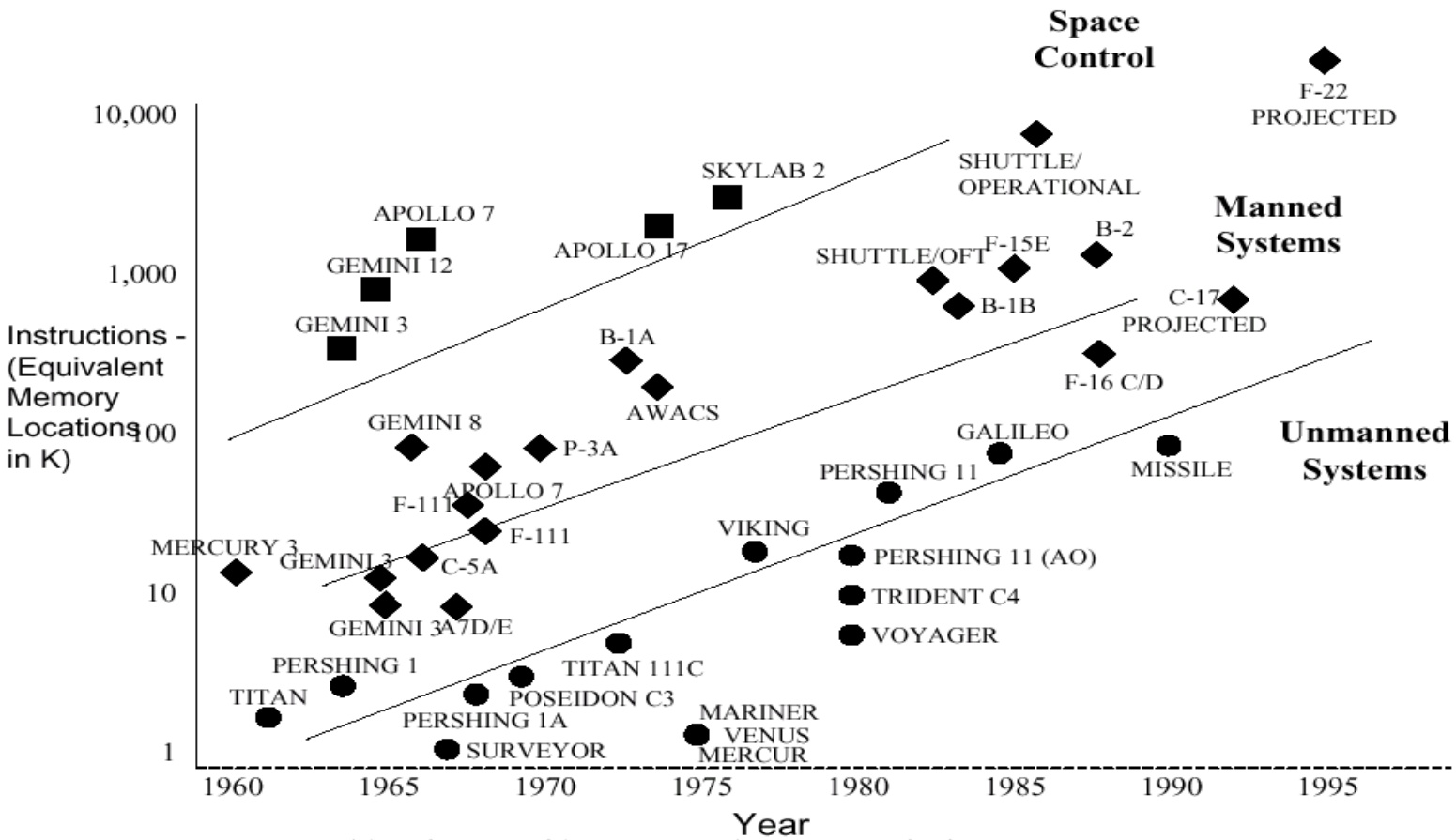
- 16% of all IT projects complete on time and on budget.
- 31% are cancelled before completion.
- The remaining 53% are late and over budget by 89%.
- Of the IT projects that are completed, the final product contains only 61% of the originally specified features. *CHAOS Study, Standish Group, 1999*
- But latest data indicates improvements may be happening. The Challenges of Complex IT Projects, RAE/BCS, April 2004

US aircraft software fit



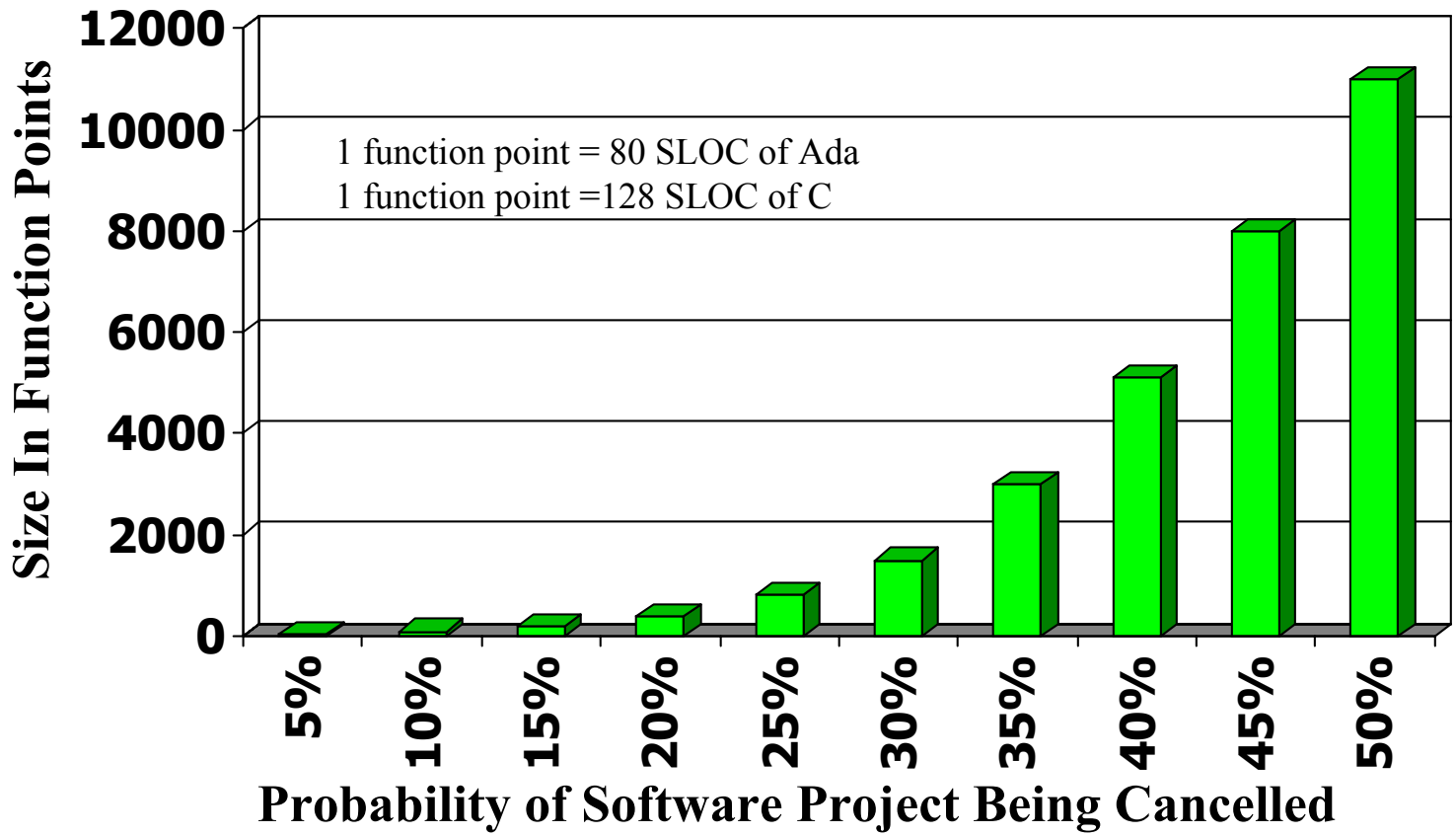
DoD Defense Science Board Task Force on Defense Software, November 2000

Software size growth



Guidelines for Successful Acquisition and Management of Software-Intensive Systems, Air Force Software Technology Support Center, June 1996

The Problem - Size matters



Capers Jones, Becoming Best In Class, Software Productivity Research, 1995 briefing

Risk and uncertainty in DPA

2001 NAO MOD Major Projects Report

"ASRAAM - Missile hardware and software technical difficulties (+18 months).

Attack Helicopter - Static Code Analysis of software (+£8m).

C130J - The in-service date (delivery of the 12 th aircraft), was achieved in June 2000 - some 23 months late. These delays arose due to difficulties experienced in the Contractor's development programme, largely hardware and software integration problems. Late delivery of sub-contracted avionic equipments and difficulties with their integration which caused delay to start of the contractor's flight test programme. Further difficulties were experienced during the flight test programme and included hardware/software integration problems, (23 months).

High velocity missile system – Project Software problems encountered in integrating ADAD into SP HVM caused seven months delay.

Merlin Mk 1 - Safety Critical Software Analysis (+£12m).

S&T Update – Software engineering problems on sonar system (+£9m). Sonar system development has been delayed due to software engineering problems (+12 months).

Tornado GR1 MLU - Integration of a Computer Loading System (CLS) to provide a mission data and software loading capability, whilst retaining the option to revert to the Cockpit Voice Recorder (CVR)."

Past Programmes

Software Intensive Projects Improvement Programme (1997)

Area	Number of recommendations	Number implemented
Policy	5	0
Process	3	1
Augmentation Team	4	0
Capability	2	0
Programme Management	2	0
Project Management	2	0
Competences	4	0
Longer term	3	0
Total	**25**	**1**

It could be worse - since 1985, 6 major DoD studies, 134 recommendations, 3 implemented

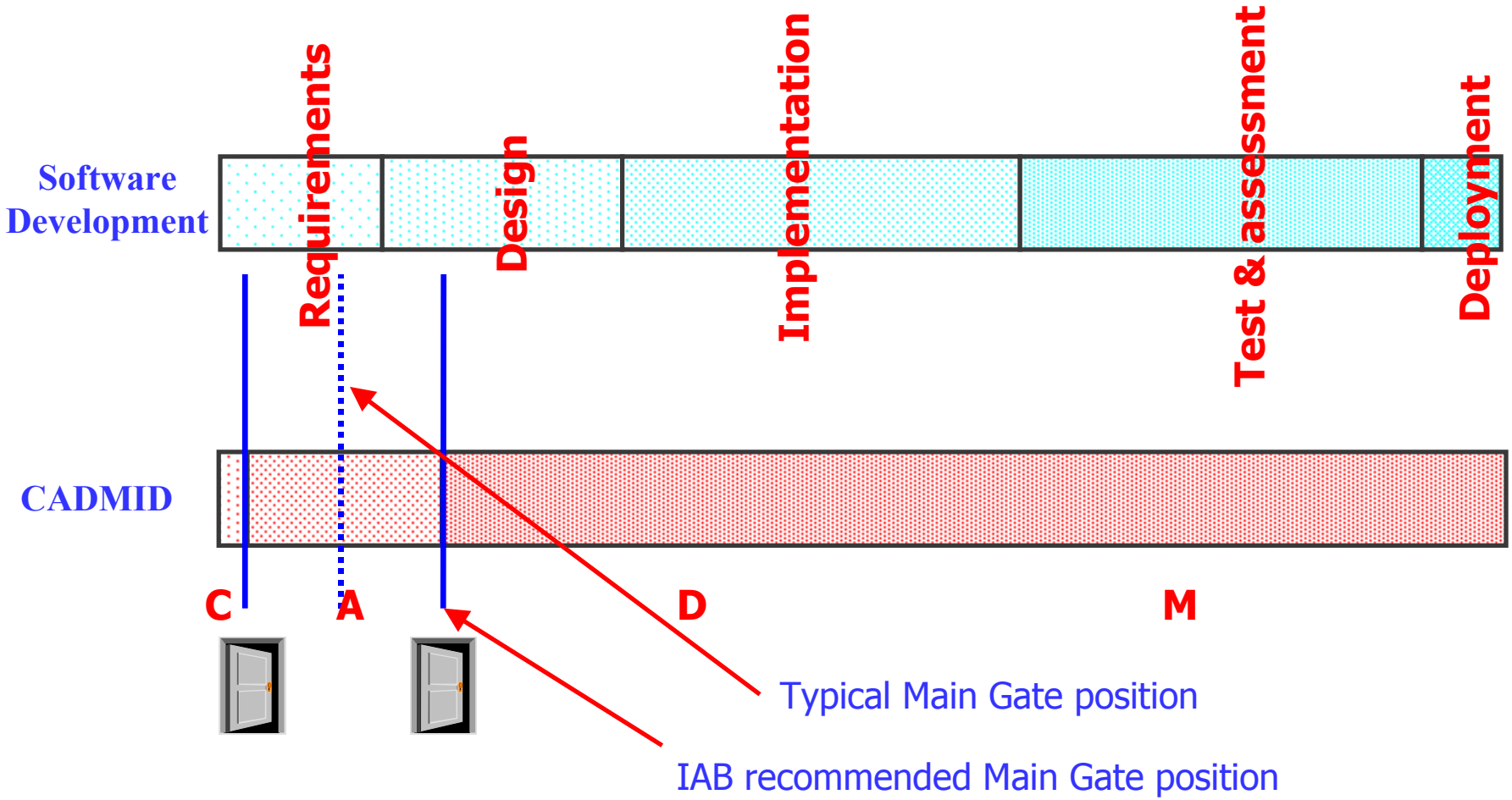
Which implies that:

- Craziness is doing the same thing and expecting a different result. Tom Demarco
- With a few minor exceptions, **all** DPA projects will use software to deliver an increasing proportion of the functionality.
- Comments such as "I am acquiring a system and not software" are ignoring the greatest risk to the project.
- Unexpected software acquisition risks should keep you awake a night.

Contents

- Background.
- Sources of acquisition risk.
- Risk reduction.
- Outstanding problems.

Sources of risk - typical Phase Cost profiles



Is Software different?

- Software does not wear out.
- Only design faults.
- Discontinuous behavior:
  - DO 10 J=1,3
  - DO 10 J=1.3
- No manufacturing phase.
- Novelty, difficulty and complexity.
- Difficult to measure products (eg dependability).

Software reliability targets

- A320 flight control system - 10^{-9} failures per hour.
- Sizewell B PMS - 10^{-4} probability of failure on demand.
- THORP chemical plant - 10^{-4} probability of failure on demand.
- Air traffic control - <3 seconds down time per year.
- Surgical robots - undefined!!
- Automobiles - no published figures.
- Railway signaling and control - 10^{-12} failures per hour.

- Current limit which is statistically valid is 10^{-4} failures per hour.
- **Anything better is based on judgment.**

Barry Boehm's Top 10 Risks

- Personnel Shortfalls.
- Unrealistic Schedules and Budgets.
- Developing the wrong software functions.
- Developing the wrong user interface.
- Gold-plating.
- Continuing stream of requirements changes.
- Shortfalls in externally-performed tasks.
- Shortfalls in externally-furnished components.
- Real-time performance shortfalls.
- Straining computer science capabilities.

Boehm, Barry W., "Software Risk Management: Principles and Practices ", IEEE Software 8(1), pp. 32-41, January 1991.

- Acquisition management and staffing
- Requirements/architecture feasibility
- Achievable software schedules
- Supplier integration
- Adaptation to rapid change
- Quality factor achievability and tradeoffs
- Product integration and electronic upgrade
- Software COTS and reuse feasibility
- External interoperability
- Technology readiness

Software-Intensive Systems of Systems - CrossTalk, May 2004

Effects of acquisition risk and uncertainty

- Software acquisition risk and uncertainty can impact:
  - Cost
  - Schedule
  - Functional and non functional attributes
- Each is not independent.
- Complex relationship - difficult to quantify.

How bad can it get?

- Use Cocomo II Early Design Model.
- Assume software development of 33,000 SLOC.
- Vary scale factors (5 parameters).
- Vary effort multipliers (7 parameters).
- Best, most likely and worst case scenario.

Cost £m			Schedule months		
Min	Most likely	Max	Min	Most likely	Max
0.057	1.0	99	6.6	17.6	96

The "silver bullet"

- Software development of 33,000 SLOC.

Status	Cost £m	Schedule (months)	Maximum staff
Average	1.0	17.6	7.8
Best facilities	0.73	15.9	6.3
Minor improvements in all areas	0.34	11.9	4.0

Contents

- Background.
- Sources of acquisition risk.
- Risk reduction.
- Outstanding problems.

Risk Reduction

- Ensure adequate weighting for software at tender assessment.
- Ensure that the use of software development best practice is mandated in the contract.
- Best practice is fairly well understood but evolving.
- Ensure required skills are in place at the contractor **and** in the DPA.
- Expert review must be integrated in to the acquisition process.
- SISAIG is important - US,UK and Aus have the same problems.
- Plan for the unexpected!!

Contents

- Background.
- Sources of acquisition risk.
- Risk reduction.
- Outstanding problems.

Problem areas

- Software skills shortage and best use of existing skills.
- Optimum use of COTS components.
- Software measurement.
- System of systems, in particular security issues.
- Interoperability.
- Requirements capture.
- Obsolescence.
- Software engineering.

Systems of Systems

- NEC a system of systems
  - Not designed together
  - Very large
  - Very complex architecture
  - Beyond our experience
  - Cost (IG & MG)?

But what about support?

- Acquisition costs (and risks) are a small part of the WLC.
- Software support risks and costs are much more difficult to predict.

Any Questions

Dave Thombs
Slimbridge Software Services Ltd
Beresford House
Churchend
Slimbridge
Gloucester GL2 7BL
Tel. 01453 890353
e-mail: dave.thombs@ntlworld.com