Learning From our Mistakes – Prospects for a Discipline of Software Forensics

Paul Bailes$^{1,2}$

with

Christine Cornish$^{1,2}$, Toby Myers$^{1,2}$, Lou Rago$^2$, Nick Tate$^3$ and Mal Thatcher$^4$

$^1$School of ITEE, The University of Queensland, QLD 4072 Australia
$^2$BCI Technology, Level 2, 167 Eagle St, Brisbane, QLD 4000 Australia
$^3$RDSI Project, The University of Queensland, QLD 4072 Australia
$^4$Mater Health Services, Raymond Tce, South Brisbane, QLD 4101 Australia
The Message

If aeronautical engineering can be developed and matured "from scratch" within a century, there is no excuse for accepting the excuse of immaturity for scandalous outcomes of software development and procurement.
Overview

- Software Development Catastrophes
- Chaos not Order
- Aeronautical Engineering Lessons
- Towards SEFA
- Challenges
- Conclusions
Software Development Catastrophes

- USAF ERP - US$1B(illion)
  - [http://www.cio.com/article/721628/Air_Force_scraps_massive_ERP_project_after_racki ng_up_1_billion_in_costs](http://www.cio.com/article/721628/Air_Force_scraps_massive_ERP_project_after_racking_up_1_billion_in_costs)
- UK welfare - UK £300M(illion)
- Queensland (Aus) Health payroll AU$1.2535B(illion)
- UK NHS - UK £12B(illion)
- Obamacare
- Charette’s (2005) Hall of Shame – 9-figure losses unexceptional
- More recently ...
Chaos not Order

- Requirements: nat.lang. vs logic vs graphics
- Specification: model-based vs abstract; logic vs graphical; executable vs non.
- Design: TP vs OO vs ...
- Implementation: C++ vs Java vs Scala vs ...
- V&V: formal methods vs testing
- Overall: waterfall vs V vs agile
V vs agile

V
• Requirements
  – System Design
    • Architecture Design
      – Module Design
        » Coding
      – Unit Test
    • Integration Test
  – System Test
• Acceptance Test

Agile
• requirements refined from iterative coding
Aeronautical Engineering Failures

• very well-documented
  – http://en.wikipedia.org/wiki/Aviation_accidents_and_incidents

• ICAO Treaty (annex 13, since 1951) mandates investigative standards including national investigative bodies e.g.
  – http://www(aaib.gov.uk/home/index.cfm
Stress distribution at 56.9 kPa cabin pressure and 1.3 g inertia loading.
Where the crew of TE901 thought they were flying — along the computer track used by the previous sightseeing flights.

Where TE901 was actually flying — following the changed computer track. It looped down to 1500ft through a gap in the clouds.
Aeronautical Engineering Lessons

• Evidence-based
  – fatigue
  – whirl mode → flutter
  – engine attachments

• Beyond technical
  – maintenance procedures
  – integrity of navigation parameters
  – ICAO treaty

• List the findings and causes established in the investigation. The list of causes should include both the immediate and the deeper systemic causes.
Towards SEFA

• Knowledge
  – basic SE
  – legals
  – research (general and specific)

• Modes
  – preventive
  – corrective
  – investigative

• Structure
  – private “Software Forensics Institute”
  – AAIB/ATSB/NTSB
Challenges

• apprehend narrative record of specific projects
• assess against specific SE processes & techniques
• assess artefacts also
• what is the philosophical basis for making inferences
  – how do you really know if A caused B
Conclusions

SEFA raises hopes for the following:

• an evidence-based, more specific understanding of the different circumstances under which different software processes and tools are more or less appropriate;
• similarly for other variations from canonical process(es);
• meta-level tools and techniques to enable the above;
• more specific directions in software engineering education and training;
• incidentally, because software systems dominate aeronautical engineering, a formally-established “Software Forensics Institute” would discharge implicit ICAO obligations in software dimension of air accident investigations.
Some specific tech. issues

• “black boxes” for software developers
• integrated standards as bases e.g.
  – OGC Gateway
  – ISO standards (12207 & 15288)
• a rubric when “agile” methods are appropriate (or not)!
• meta-level considerations e.g.
  – recording of forensic investigators’ implicit assumptions
  = “black boxes” for software forensic investigators
Aims:

• to advance the theory and practice of software engineering through the analysis of software development projects by distinguishing between the characteristics of successful versus failed or failing projects;
• simultaneously, to develop tools and techniques to facilitate these analyses;
• equally, to foster the development of social institutions and practices (both voluntarily and by regulation/legislation, as appropriate) that will encourage the adoption and application of the above;
• thereby engendering improvements in the timeliness, cost and effectiveness of significant software procurement exercises;
• and thus, to achieve the economic and social benefits resulting from all the foregoing.