

Hatching a Phoenix: reclaiming success from software failure

Darrel Ince, Marian Petre
Department of Computing
Open University
Walton Hall, Milton Keynes
MK7 6AA
d.c.ince@open.ac.uk
January 30, 2010

The Challenge

“A minute's success pays the failure of years.” (**Robert Browning**)

We pose a challenge to the applied computing community: to reclaim a spectacular failure. There is something of a history of failure of computing projects initiated by the UK government, for example, see the front page of *The Independent* of 19th January 2010: “Labour's computer blunders cost \$26bn”. The challenge is to rectify one of these expensive blunders, by building an effective open source system with a fraction of the resources expended on the original project and in a fraction of the time that it took for the original project to be cancelled.

A possible system that we could redevelop would be the C-NOMIS system, which was intended to track the progress of an offender through the justice and probation system. It was cancelled after a \$600m spend of which approximately \$150m cannot be accounted for. It is conceptually simple: it is a chronicling system that keeps track of an offender through different locations and agencies. However, it is socially complex, involving a number of stakeholders with disparate perspectives.

Hence, a significant part of the challenge lies in finding efficient means to identify requirements and specify success – or at least sufficient success for the system to be useful and usable. To some extent, this involves separating the politics from the engineering, identifying effective ways to collect information about needs without mobilising political manoeuvring among stakeholders.

One approach would be to develop a generic version of a system which could be instantiated to something specific, rather than developing a specific system. For example, there is very little conceptual difference between C-NOMIS, the cancelled Integrated Children's System, ContactPoint, and the patient record component of the NPfIT programme.

Advantages

There are a number of advantages that should appeal to the computing community:

- It addresses the zeitgeist. It's an example similar to the *NetFlix* challenge which asked teams around the world to develop software that used data mining and other technologies to develop a recommender system – a challenge that was successful.
- It is focussed and practical, and the criteria for success can be articulated.

- It would appeal to the software engineering community, which seems under-represented in grand challenges.
- It would be a publicity coup to achieve the challenge.
- It would point out to government that computing academia is ready and able to enter a debate about e-government computing in a constructive way.
- It would demonstrate the potential of open source development for public systems.
- It is a macrocosm of software engineering challenges, such as security, privacy, requirements conflicts,

How to do it

There are a number of ways to proceed, for example:

- One is to capture the imagination and commitment of an existing community, which could take on the challenge.
- Another is to adopt an open source model. Gather a team of academics to meet over a short time and develop an architecture. Publish the architecture and invite contributions of code. University departments could challenge students, say, as part of a second-level practical course, to implement portions of the system.

There is some evidence that such challenges work, albeit in a local context. One of us (Ince) has just re-engineered the failed Integrated Children's System replacing heavy-duty MIS technologies with a word processor, spreadsheet and some regular expression-based Java code. Students in our department have just completed writing a book on computer art that uses the Java-based *Processing* system of the 69 students who took part 65 delivered material of the highest standard. There is also the success of a number of open-source projects, for example, those sponsored by the Apache Foundation.