Behind Great Infrastructure is Excel: Experiences from a Large Engineering Project

Mark Handel and Steven Poltrock
Boeing Research and Technology
M/C 7L-70, PO Box 3707
Seattle, WA 98124

mark.j.handel@boeing.com, spoltrock@gmail.com

ABSTRACT
We describe our work in developing specialized software applications to support a large collaborative engineering program. We find that despite many of the applications being bespoke efforts, designed to the requirements of the users, virtually all major applications have an unofficial spreadsheet or database backing up the official application. This spreadsheet is invariably where the actual work of the application occurs, with the official process and system being used primarily for mandated record keeping and auditing purposes. We discuss the implications of this finding for scientific collaboration and long-term record keeping.

General Terms
Human Factors.

Keywords
Software Infrastructure, Excel, scientific collaboration

1. INTRODUCTION
1.1 Engineering versus Scientific Collaboration
Our research setting is a large collaborative engineering program focused on developing a complex manufactured product. Although this is an engineering, not a scientific, program, we argue that there are significant similarities to scientific collaboration and infrastructure that make our work relevant and useful.

In terms of the size and structure, this program is significantly larger than all but the largest scientific collaborations. The program's overall costs run in the many billions of dollars, and it has been underway for nearly ten years. We expect the program to continue, at a minimum, for at least another thirty years. This means that records and data being generated today will need to be maintained for most of this century. The work is divided between the lead integrator, about six partners who have both a significant financial and an engineering stake in the project, and about 100 primary suppliers, responsible for smaller elements of the final product. The partners and suppliers are in various locations around the world, including Australia, Japan, US, and many European locations. Final assembly of the product will take place in multiple US locations.

Although the general outlines of the configuration of the final product was known early in the program, there were significant unknowns that had to be resolved during the engineering phase. These included uncertainties about the materials used, design of the product, manufacturing techniques and processes, and supply chain issues. This situation is similar to many scientific collaborations where the outcome, and in some cases even the general direction of research, is not necessarily known at the beginning. Some of the problems encountered have been both substantial and late-breaking, requiring urgent and significant coordination between multiple partners.

1.2 IT Infrastructure
Like many recent large scale scientific collaborations [1, 3], this program also has a large IT infrastructure developed to support collaboration among all participants. The authors have supported several IT architecture projects, and it is this experience that forms the basis of our position paper. The IT architecture is fairly complex. At the core is a suite of three integrated engineering applications, each highly customized for this program. Tightly integrated with the core suite of three applications, is a larger collection of about 30 secondary applications. These applications are widely used and mission-critical, but do not necessarily share the same visibility as the core suite. These applications have also been extensively customized or developed especially for this program. Finally, there is a larger set of approximately 500 applications that have specialized, domain-specific uses. Although the development and deployment of these applications has been difficult and challenging, the overall IT infrastructure has not been seen within the company or industry to be a limiting factor in the program's schedule or success.

These software applications are acquired, developed, modified, and deployed in a context with demanding requirements. This program's final product is subject to strict regulatory requirements and exacting quality control (ISO9000-like) requirements. This necessitates the software's use being mediated by formal, documented processes. These processes ensure consistency between different partners and suppliers, compliance with regulatory requirements, and documentation of engineering decisions. This overarching process infrastructure may be one of the biggest differences from scientific infrastructure. There is a extensive library of documents detailing how work is done. The library has a treed hierarchy of documents, 6 levels deep, detailing everything from overall requirements and goals down to step-by-step work instructions. In addition, there is a full time training organization to support learning how to use the tools and processes. The human infrastructure needed to create, maintain, support, and update the contents of these applications is substantial [2].

We helped to develop and deploy both the processes (including the documentation and training) and tools for eight applications used by this program. Within the program, an application is generally viewed as the combination of both software and processes. There can be an application with no specialized software, or two applications can share the same underlying software package.

2. SOMEWHERE, THERE'S EXCEL

In our development process we strove for an iterative approach, with multiple releases of process and software. Each release added new functionality and addressed problems encountered in previous releases.

As we worked with end-users, we found that virtually all of them had started an unofficial, shadow application, usually based on an Excel spreadsheet or an Access database, to track the progress of each process instance. These shadow applications were not part of the process documentation, nor were they supported by the IT organization. (In many cases, the IT organization didn't even know the shadow applications existed.) Table 1 lists all eight applications and indicates the type of shadow application. Only one application did not have an Excel or Access file backing it up, *Neuron*. *Neuron* is a special case, and will be discussed later.

An Excel file was used to shadow the *Luxury* application, and it was sufficiently important to the end users that they developed an Access database to take its place. We saw a similar progression of maturing the unofficial tools in other processes as well: both *Embargo* and *Fluffy* moved from an Access database to a fully supported SQL Server based application. Also, in the case of *Moscow*, the bespoke application was eventually abandoned in favor of a combination of process control and rapidly matured unofficial tools.

Table 1: Shadow Excel / Access tools behind Processes

Process[1]	Excel	Access	Work Item (W) or Data (D)
Luxury	X	X	W
Moscow	X	-	W
Whiskey	-	X	W, D
Fallen	X	-	W, D
Fluffy	-	X	D
Embargo	-	X	D
Ticket	X	X	D
Neuron	-	-	W, D

Some of these shadow applications were used to track the work items managed by the official applications. In many cases the official applications were essentially workflow applications, and the shadow applications provided a way to track the status of a workflow document. In Table 1, where these shadow applications were used to track work items, the last column shows "W." However, it was not always the case that these shadow applications tracked just the work item. Half the applications were used to record the underlying data or to manipulate the data in a meaningful fashion. These are marked with a "D" in the last column.

Why were these shadow applications developed? Invariably, the applications did not conform to the way work was done on a day-to-day basis. Situations such as gaps in functionality, additional processing required on the underlying data, or differences in work processes all drove the development of the shadow application. For example, the manager of the *Luxury* process wanted to know the reason for any process delays. *Luxury* could readily provide a list of all process instances and report any delays, but not the reasons for the delays. The Excel and Access files contained explanations such as key employees on vacation or dependencies on other processes that were unavailable to *Luxury*. Sometimes the tasks tracked by *Luxury* were informally decomposed into subtasks, assigned to a team to resolve, and then recorded in *Luxury* when all the detailed work had been completed. *Luxury* had no provisions for this kind of task decomposition.

A particularly interesting case was found while visiting a supplier in Japan. One employee of this supplier was responsible for updating the official system (*Fallen*) and generating a local Excel spreadsheet for tracking the work within this supplier. The first few columns of the spreadsheet matched exactly the output of *Fallen*. The next columns were exact copies, only translated into Japanese. Other workers used the Japanese columns, while the English columns were used to synchronize with the source system.

The case of *Neuron* deserves special attention. *Neuron* was a late addition to the IT infrastructure, and rapidly became one of the most important application / processes. *Neuron's* purpose was to provide dumps of relevant data from the core and secondary applications into Excel spreadsheets. This was a late realization that Excel does provide significant value in the engineering design application chain. However, *Neuron* is a one-way dump of data. There is no provision to load data back into the source systems, nor to track what happens to the data after it is exported.

3. DISCUSSION

In our explorations of the use of show applications we frequently heard the infrastructure applications described in terms such as "brittle," "constraining," and "lumbering." The perception was that the official software was only used to maintain the required long-term records, and that the informal applications were used to get work done.

When traceability of data and actions are important, the near-universal use of informal applications has important repercussions. There is a significant amount of tribal knowledge encoded in the informal applications. Once the data is transferred into the official tools, this knowledge is lost or greatly distilled. For scientific collaborations, this means that much of the details of the manipulation and analysis would be lost. Later re-analysis of the data would be impacted by the loss of this information.

[1] : The names of the processes are code names, not the actual process name

We work in a highly regulated environment. Following the documented process is an important step in ensuring regulatory compliance. The concern is that these unofficial tools are not documented and can lead to process breakdown or an audit finding. As scientific infrastructure becomes more mature, there is a greater expectation of repeatability across different groups doing similar work. Adoption of informal applications could pose a problem: one group may have subtle differences in data gathering, analysis, or work processes that make data sharing more difficult.

The program we describe here is one of many of its kind within the company. There are often attempts to move applications developed within one program to another, in an effort to reduce the overall number of applications and promote commonality across the company. These attempts are often hampered because only the official part of the application is moved. The unofficial applications stay within the originating program, greatly impacting the success of the new deployment. Similar situations could be easily predicted to occur as scientific applications are moved to new collaborations, without a full understanding of the unofficial support applications.

These shadow applications pose a clear risk to an organization determined to follow standard practices. There are no simple solutions to this situation. One approach could be to lock down the infrastructure, requiring everyone to use the approved software and actively discouraging non-standard and informal tools. Of course, this requires a degree of control over workstations that would be unusual and difficult in a research setting, and would cause overall productivity to plummet. (If the informal tools didn't make the work easier, they would be quickly dropped by the users.) Alternately, the informal tools can be adopted into the approved suite of software. However, this assumes that the users are savvy enough to generate the informal tools and that overall management, and in particular IT management, is willing to accept the responsibility for further development and maintenance of the applications.

4. ACKNOWLEDGMENTS

The authors thank members of the process and tool development organization within the program. In particular, BB, LS, SO, and TM have been invaluable in helping the first author understand much of the detail of process development, the regulatory and quality framework, and how the program's work actually occurs.

5. REFERENCES

[1] Enabling Grid for E-sciencE (EGEE), http://www.cern.ch/egee

[2] Lee, C., Dourish, P., Mark, G. "The Human Infrastructure of Cyberinfrastructure." *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work.* p. 483-492.

[3] Olson, G, et. al. "The Upper Atmospheric Research Collaboratory" in *interactions*. Vol. 5, No. 3, p 48-55