Experiences with implementing multi-user database platforms to increase quality and efficiency of management and communications on large and medium size infrastructure projects

Mike Turner, Kevin Gutteridge, Doug Thiele, Cullen Welbourn August 24, 2015

Abstract
Based on extensive personal experience the authors have found that most medium and large scale infrastructure projects fail to take advantage of the efficiencies available from modern data management technology. We describe here an approach that has proven highly successful at bringing together people, data and business rules in a common information environment in the chaotic, fast-paced world of railway infrastructure projects. We believe it is extendable to many other project applications.

The Context
We are a group of four technical managers and engineers who have been working on railway and mining infrastructure projects around the world for over 25 years each. We have observed the problems described here to be common and chronic. The solutions we describe were evolved over the last 15 years in positions from all sides of the projects (owner, owner's engineer, operator, integrating contractor, subsystem supplier). The projects have included some of the largest such jobs in the world. Typical durations and values are 5 to 10 years and USD 10M to 2B. Fields and disciplines involved include civil works (tunneling, viaduct and station buildings), trackwork, rolling stock, power, communications, signaling and control centre integration. We believe the solutions will apply equally well to any type of infrastructure project.

The Problem
The following list of problems will be familiar to managers and engineers on any economic enterprise. We have observed that larger projects usually have the added complication that the stakeholders and parent organizations treat them as "one-off" situations. This is sometimes because the relevant contract requires a set of custom execution rules, and sometimes because the relevant party is a consortium or project team assembled from multiple parent organizations and acts as a new entity belonging to none of the parents. Either way the life cycle management does not use proven or established rules and tools and each project has to invent its own set from scratch, every time. When compared to organizations that invest in and apply a standardized approach to all their projects, the resulting depth and quality of information management is usually substandard, with corresponding negative and "hidden" effects on quality, efficiency and project risk.

In a typical example, one of us started work as an engineering manager for the general contractor for a transit line (contract value USD 500M) to find that they intended to track all correspondence, documents and drawings using a stand-alone spreadsheet resident on a clerk's desktop computer. The information was not secure and it was not available to the large number of staff who needed it on a daily basis. Additionally there were no plans or budget for improving the situation. When we pointed out that the capacity (then) of a spreadsheet would not hold the expected number of items (200k) the first response was that it was a future problem and would be solved later. This approach, had it been allowed to continue, would have deprived several hundred staff of quick and easy access to information that was essential for efficient performance of their duties. Each separate group would have responded in a different manner to obtain, distribute and maintain the subset of the information that it needed. The result would have led to creation, throughout the organization, of multiple sets of indexes and data repositories (electronic and paper) each of which would have been uncoordinated with the others and at best minimally complete for the purposes intended.

The following is a partial list, expressed from the viewpoint of the user of the information, of some of the more frustrating problems we have encountered:

- It is very hard to make sure that everyone who needs it can see a particular document.
- It is very hard to establish and follow a correspondence chain.

- I am not sure that all the inbound letters have been correctly dealt with.
- I am not sure that all outbound letters have been fully reviewed by all parties who need to do so.
- It is very hard to get a complete and up-to-date picture of the lifecycle status of a particular document or correspondence subject.
- I am not sure that everyone is using the correct revision of a document.
- I am not sure who has seen or received this particular document or letter.
- I am not sure that all the relevant contract requirements have been identified and responded to.
- I am not sure that requirements are tracked adequately through the design, implementation and testing phases.
- I am not sure that we are ready to [energize, ship, test, …] against a particular milestone.
- I wish I could search the correspondence and document sets the way I can search on Google.
- I need to track and report the status of issue X at weekly intervals based on [documents issued, letters received, widgets installed, …]

The project management lexicon has general terms for the functions concerned, including:

- Document Control
- Correspondence Management
- Requirements Management / Verification and Validation (V&V)
- Configuration Management
- Progress Tracking and Reporting.

In the typical project organization these are treated as separate, almost entirely clerical, functions and responsibility for them is placed under different parts of the organization. We believe this is an artefact left over from the days of paper-based management when information flow was required to be tightly bound to the organization diagram.

A New Approach

We have found that we can make dramatic improvements in workflow, job satisfaction and, we believe, efficiency by providing all users with access to all these standard management functions via a multi-user database platform and then allowing the project team to invent and evolve their own customizations, variants and even new tools on the same platform. The benefits are twofold.

First, a common implementation of the standard functions eliminates duplication of effort and resources within the organization and improves access to information for all users. This in itself is often sufficient justification for the approach.

Second, and most importantly, we have observed unexpected and significant additional benefits arising because the various data sets are available to each other for cross-referencing and co-use. For example, the strength and coverage of a requirements management process is easily improved if it can be cross-referenced with the project's master list of documents.

The following are points we have found to be key to successful solutions:

- It is a great advantage to start with a platform where the basic functions have been proven on other jobs.
- Select a database technology that can be distributed to everyone. It is alright to have a few levels of user ability (eg: read-only, normal user, super user, admin) but avoid products where the per-seat

cost of giving access to everyone is prohibitive. The advantage of the approach are realized only if it is available to the entire team. Technologies that use internet browsers as user interfaces are attractive for this.

- Select a database technology that allows very fast development and rework. The "production" database needs to be capable of being modified in almost real time. The software industry sometimes uses the terms rapid application development (RAD) or "agile" for this.
- Make the platform extremely responsive to user's needs. The more people who use it and the more things it is used for, the more it adds value to the enterprise. If something doesn't work or isn't being used, then fix it or abandon it. Avoid the classic developer / user dichotomy by making the developers act as users and the users act as developers. There is no time or resources available for requirements and functions to be formalized and exchanged between the groups. We have had good success with a hybrid model where a staff member dedicated to toolmaking activities works for a senior manager whose mandate includes several of the functions listed above. The dedicated toolmaker is promoted from within the existing project staff based on aptitude, and grows into the position. We find that even large projects can be supported with a very small number of dedicated toolmakers (typically 1 or 2) when guided by an experienced manager and particularly when a proven platform from another project is used as a baseline.
- Data integrity is not a primary concern. We have found that, provided the platform has frequent and reliable backups, the data itself does not need to be treated with a high level of integrity. This is not a point of sale, or inventory or banking application where a single data error loses money instantly. It is more like a library card catalog - it has to be highly available but errors are usually self-correcting over time and easily tolerated.
- Select a database technology that gives very fine control of users and their authorities. Ideally this should be context-dependent and operate at the level of individual fields and records.
- Be prepared to "connect" the platform to other data sources in order to add value. We have successfully incorporated into our platforms data from commercial packages including Maximo, Documentum, Aconex, EDMS, Doors, Comply Serve and Primavera.
- Select a database technology and internet connections that allow remote users. Modern project teams work from locations around the world and it is important to be able to give everyone access to the platform. We have used Citrix (and Citrix-like) applications, as well as custom PHP interfaces and direct-to-web page generators. The trickiest part of this is usually negotiating with the remote users' IT departments for firewall access.

Appendix 1 presents a list of functions available on a typical platform. It shows the basic functions considered essential to make the platform worthwhile as well as a number of other additional functions that are surprisingly easy to implement given the existence of the platform and the basic functions. Figure 1 shows a simplified map of the main data sets and their relationships on a typical platform.

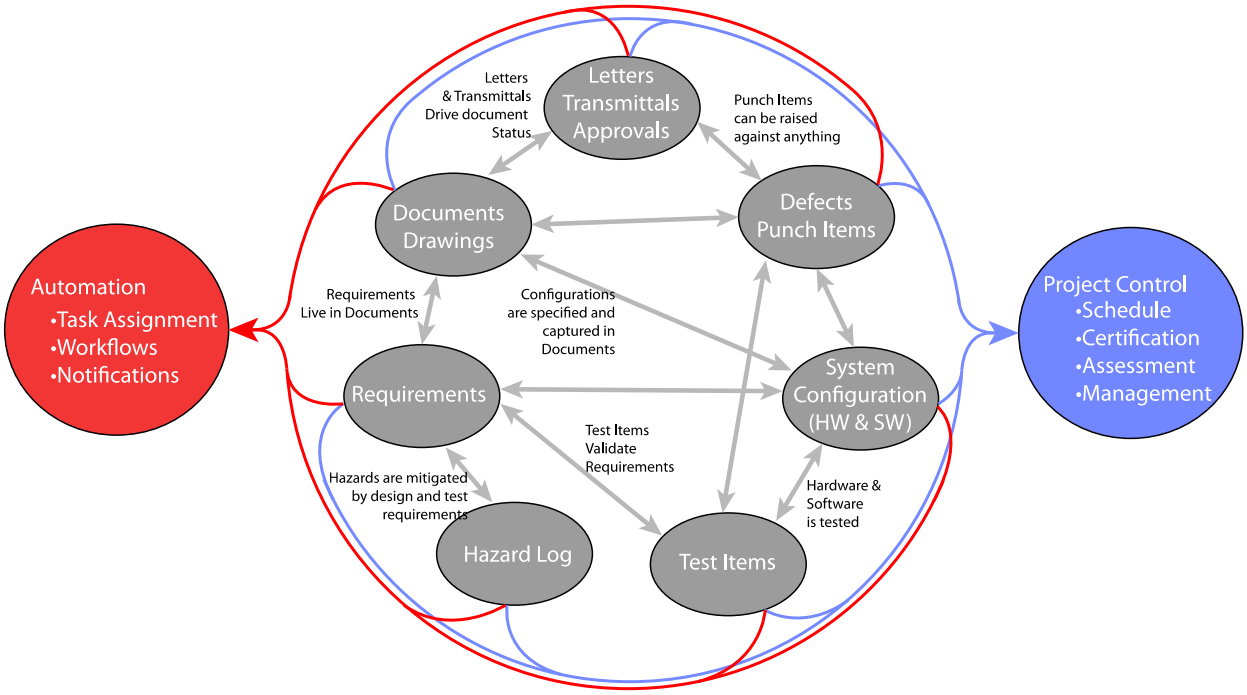

Figure 1 Main Data Sets and Relationships on a Typical Project Platform

As we developed it, the aspect of this approach that surprised us the most was the ease with which additional and highly effective functions could be developed once the basic data sets were available to each other and to the users via the common platform. We present three examples below, but there are many others.

Example 1: How do you know you're finished? Every project is faced with the problem of deciding when a particular milestone has been achieved. In some cases this assessment is needed for commercial purposes and in other cases there is a safety aspect. For this purpose most projects use some sort of certificates - a document claiming / stating that a particular milestone or objective has been achieved and signed by the parties responsible for delivering and checking the result. For example, prior to energizing an electrical substation, the engineering and construction groups might prepare and sign a certificate that would be accepted by the testing group (who's safety depends on correct design and construction). Or, prior to handing over a railway line to its owner, the general contractor's engineering, construction, testing and quality groups might prepare and sign a certificate of readiness for acceptance by the owner. On our projects, each certificate is supported by an underlying hierachy of PDLs (Prerequisite Data List). Each PDL is a list of evidence (documents, correspondence, test results, etc.) that supports a particular sub-objective of the certificate's objective. The common database gives an "OK" or "Not OK" conclusion for each item on each PDL and then sums the results upward to give a final overall conclusion for the target certificate. The PDL hierarchy and the entries on each PDL are constructed early in the life of the work and the database makes overall status available to all users. Starting from months prior to a target milestone, the stakeholders can be provided visibility of all the prerequisites to be considered and the responsible managers can use the platform to focus their attention on the "Not OK" items. The net result, in our experience, is a very smooth, controlled and highly visible advancement to full completion that reduces risk, both safety and commercial, for all parties.

Example 2: Who does what? We have used the common platform approach to present each user with an integrated list of the basic data items (documents, letters, etc.) that require their action. The outstanding items are presented to each user at logon. Hyperlinks allow the user to move quickly to each item and often to execute the action as well. Typical examples include:

- approving or commenting on a draft of an outbound letter
- reviewing a design document and registering comments against it
- assigning actions to be taken on an inbound letter

Example 3: Effective, real-time requirements management. Forensic analysis of major cost overruns on large infrastructure projects frequently trace the root causes to poor requirements management, particularly where software is involved. As a result, most modern projects mandate some form of requirements management (RM) in their procurement chains, and often force the contractors to use a particular commercial RM package. In these cases it is very rare in our experience to see a contractor's RM implementation that adds any appreciable value for any party. The main reasons for this are:

- the high per-seat cost of the specified RM package inherently limits the number and expertise of people involved in the process
- most contractors do RM off-line so that completing the RM is not a criteria for releasing the design. This limits the efficacy of the RM feedback since it is performed after the fact.
- the common commercial RM packages are very easy to use badly and extremely difficult to use properly
- the common commercial RM packages are difficult to integrate with other design management and document control systems

We have found that most of these problems can be eliminated by adding RM modules to a project-level integrated database that includes document control, design management and other related functions. We have even used the common commercial RM packages as sub-modules in the scheme. The result has made it easy to turn RM into a real-time step that adds value at the correct stage in the design process and to make the traceability and cross-checking inherent in a good RM approach more easily visible to the stakeholders that need to see it.

Conclusion

Using databases for project management is by no means a new idea. All enterprises have implemented it to one extent or another using custom tools within their own organizations. There are commercial software platforms that claim to provide many of the individual functions but we know of none that do them all or well. Our interest in this started because all of the (then) available platforms for a document control system were insufficiently customizable for our purposes and none had a low enough per-seat cost. We constructed our own platform, began to add basic functions and then applied the experience to additional projects. We have presented here some guidelines for the properties of a successful platform, the most important of which is that it be agile and easy for the users themselves to adapt and maintain. Our novel observation, and the main point of this paper, is that the resulting synergy allows the group to build, very efficiently, a set of effective management tools that were not possible with individual, unconnected, implementations of the basic functions.

Appendix 1
The following presents a list of some of the key functions from a typical platform

Document Control Functions

- List all documents and their revisions
- Show the status of each (eg: Pending Issue, Issued, Superseded, Cancelled) according to the particular life cycle rules of the project
- Show dates and addressee for distribution and circulation of each
- Show the status of each with respect to various project business rules such as “review by Owner” or “next action by”
- Allow users to search and sort the list
- Allow users to download (as text or spreadsheet files) or generate reports (as .PDF files) for selected subsets of the list
- Allow authorized users to download or view an electronic copy of the document
- Allow authorized users to generate a transmittal to send the item to another party on the project’s list
- Allow authorized users to enter new documents or revisions

Correspondence Management Functions

- List all correspondence, inbound and outbound
- Allow users to search and sort the list
- Identify the required action(s) and actionees for each
- Allow authorized users to download or view an electronic copy of the correspondence
- Allow authorized users to create and circulate for review a draft of a new outgoing letter
- Show the development status of draft outgoing letters.

Change Control

- List all change requests and their target items (eg: a document or a configured item)
- Show the status of each within the (project-defined) lifecycle of a change
- Allow users to search and sort the list
- Allow users to download subsets of the list as text or spreadsheet files

Configured Items

- List each configured item and show its place in the project’s hierarchy
- Show the status of the item within the (project-defined) lifecycle

Requirements Management

- Identify each requirement and its “level” within the (project-defined) hierarchy
- Identify the parent(s) of each requirement
- Identify the children of each requirement
- Identify the ownership of the requirement and responsibility for its implementation. In practical terms this means identifying the document(s) where the requirement is defined
- Provide evidence that each child requirement satisfies its parent in the manner intended and that each parent requirement is satisfied in its entirety by its children
- Provide full traceability of the requirements chain in easily visible chains or trees
- Provide records of the assessors and checkers who defined and assessed each requirement and its relationships

Test Management

- Identify test items and their intended execution (one test may be repeated many times in case of multiple components)
- identify the test procedure (document) where each test item is defined

- identify the test report (document) where the results of the test item's execution will be / is reported
- identify the status of the test item's execution (eg: pass, fail, incomplete, etc.)

Hazard Log Management

- Identify individual hazards, including ranking and ownership and overall status according to the applicable business rules for the project.
- Identify mitigations for each hazard including mapping to the relevant design requirements, document sections or test items
- Produce hazard log reports

Defect and Deficiency Tracking (Punch List, DRACAS)

- Identify defects or deficiencies and track their status against a pre-defined lifecycle
- Record the ranking and categorization of each deficiency
- Map deficiencies to owners
- Track special subsets of deficiencies according to particular rules (eg: DRACAS items)
- Produce reports

Aggregate Status Management

- Allow aggregation of selected groups of data items and desired target status into prerequisite data lists (PDL) intended to measure progress toward and achievement of particular objectives
- Allow users to define a hierarchy of PDLs with appropriate status aggregation so the top level PDL can provide final OK / Not OK status against complex objectives
- Identify deviations from the target objectives for individual data items
- Allow management to approve and control deviations
- Generate reports providing visibility of status against objectives