“I paid for cost management software. Why are they still using spreadsheets?”
Recognizing Mistakes that Break Cost Tool Implementations

Abstract - Organizations are rapidly adopting cost control software, eager to take advantage of the automation, standardization, data analytics, and benchmarking capabilities such tools offer to save users’ time. The extent a project controller benefits from cost software, however, directly relates to the decisions made during software implementation—and bad decisions can have massive impacts later. Many organizations believe that adhering to solid project controls theory or recreating existing procedures will be enough to deliver but find their cost engineers reluctant to abandon spreadsheets. A successful software implementation ultimately depends on end-user and project management buy-in. It must recognize the complexities of the cost engineer’s day-job, their reporting demands, and offer meaningful improvements or end-users will resist change. This paper seeks to elucidate frequent mistakes organizations make when adopting cost software, along with case studies drawn from the author’s experience using and implementing such tools.

Author - Jason Kimbrell is a Project Controls Specialist at Omega 365, where he develops and implements the Omega Pims Cost tool for clients around the world. Prior, he spent nearly a decade as a cost engineer in Shell’s capital projects portfolio, working on projects in upstream, downstream, and LNG, as well as serving on the company’s cost software pilot team as a super user.

Introduction

The past decade has seen remarkable advancements in cost management software, which offer tangible benefits to the cost engineer that spreadsheets cannot match. Organizations have moved to adopt these technologies, hoping to increase productivity by automating data-entry and non-value-added tasks while redeploying cost engineering resources towards analytical pursuits. Yet cost engineers have not embraced digital tools the way that planners and schedulers have embraced Primavera. Despite significant money and time, organizations complain that their cost engineers continue to clutch the broken-formulas and manual data entry of spreadsheets like castaways clinging to life rafts watching the *Titanic* sink. If cost software has come so far, why are cost engineers still reluctant to change?

Decisions made during cost software implementation have an outsized impact on tool adoption rates. Organizations must consider the realities of the cost engineer’s demands and the stakeholders they report to. Frequently, organizations outfit implementation teams with little input from future end-users, leading to tools that, at best, may be theoretically proficient, and at worst, designed for a different discipline all together. Cost practitioners and their stakeholders must be involved at every step, from deciding which software to go with, to determining the tasks to automate, the level of detail to standardize, and the kinds of reports the software should produce.

What are digital cost management tools and why use them?

While cost management software may sometimes look and feel like a spreadsheet or make extensive use of data from an enterprise resource planning tool (SAP, Oracle, JDE, etc.), it is distinct in that it is designed for the express purpose of cost control [1, p.3]. For example, reputable cost management software features built-in project controls functions, such as earned value analysis, estimate forecasting, automated time phasing of cost based on schedule dates, the creation of cost breakdown structures and alternative reporting structures, and reporting.

Cost tools standardize processes, meaning every cost engineer performs the same tasks in the same way. This allows organizations to easily onboard resources to fill gaps, as resources only need know the scope and not a new workbook to contribute. Additionally, standardization allows for data sets that are ripe for data analytics. Visualizations that were previously difficult to achieve because of sample differences no longer have those differences. The same can be said for benchmarking, where users can tag data like Primavera P6 users tag schedule activities.

Finally, cost software offers more security than spreadsheets. Reputable systems have user access rights that limit whether users can edit or view data. Software also performs more efficiently than spreadsheets, where a user might edit a formula regularly during the control process, thus introducing the chance for errors.

The result is that the focus of the cost engineer's job moves away from inputting data or managing formulas in cells and shifts towards value-add activities like analysis.

What does a successful implementation look like?

To deploy cost software, organizations work through four phases, as described in AACEI's recommended practice RP94R-18 [1, p.2].

Phase 1 – Readiness Assessment - Organizations form an implementation team. They analyze their current processes in detail and develop a list of requirements communicated to software vendors, who will in turn provide demos. At the end of this phase, the organization will select a software package to implement.

Phase 2 – Installation and Configuration – The software developer conducts workshops with the organization to fine-tune requirements and identify any alterations to "out of the box" functionality to meet the specifications. The developer then works with IT systems to connect to interfaces such as SAP or P6. At the end of the phase, the software will be fit for a pilot team to test the tool.

Phase 3 – Pilot Project – Developers train a project team to test the tool and serve as superusers for the initial roll-out in the next phase. Simultaneously, the organization updates procedures and creates a roll-out plan. At the end of this phase, the organization should decide if the software is ready for a wide audience, and if not, determine what should change or if the project should go forward.

Phase 4 – Implementation and Rollout – Project teams begin using the tool in earnest after end-user training. Reports and features that may have been considered "nice-to-have" are rolled out and a governance is put in place to handle changes to the software in the future.

Success will differ for each organization. A good tool will replicate the existing data outputs while making the inputs much easier to work with. It might tighten the organization's internal processes or bolster the organization's ability to handle multiple projects with limited resources.

The best metric for a successful cost software implementation, however, is the number of active users in the system on daily basis. If the tool does what it is designed for, active user counts should increase proportional to the number of projects in the tool.

Where does software go wrong and what can we do about it?

Mistake #1 – The wrong people are in the room

If cost control software lives and dies by its ability offer meaningful change to the end user, the end user must be heavily involved in its development. Stakeholders outside of project controls should also be included, even if they do not benefit directly, or the tool will phase similar push back.

For a successful roll-out, the organization's software team should include:

- *Project Controls Practitioners*: Practitioners have a trench perspective that allows them to give input on what will work and what will not. They must buy-in to any changes to existing practices. Organizations should include representatives from multiple business streams to prevent one team from dictating how all teams work.
- *Project Controls Leadership*: Leadership should drive software to balance the organization's goals and procedures while solving the issues that plague end-users. They must endorse the tool, or end-users will view it as optional.
- *Project Management:* Project Management must understand the value of a tool that they will never use and will likely change the look and feel of deliverables they have relied upon to make decisions. If stakeholders do not buy-in, end users will continue to produce the old reports rendering the implementation pointless.
- *Finance*: Second to project management, cost engineering interfaces with finance so much that any changes to processes and deliverables need finance buy-in. Finance should not, however, drive the development of the tool.
- *IT*: Cost tools are by nature information technology, and project controls managers are not typically IT experts. IT helps integrate the tool in the greater software framework of the organization.

Mistake #2 – Not Enough Self Reflection

Project controls tools reflect what the organization brings to them. They are not silver bullets for poor project controls and can exacerbate problems if allowed. Even in the best circumstances, they will create a significant change in the way an organization controls cost.

Successful implementations place an emphasis on the "Readiness Assessment" phase. Like a *kaizen* in lean methodology, they seek to understand the current state of cost engineering without trying to fix the issues. The end-users are in the best position to make that decision. They know what works, what fails, and where skill or practice gaps exist. Stakeholders, too, provide insight on why they ask for particular reports, and may even be convinced to do things a better way. Teams should pay attention to pain points, but should also determine whether software is needed to fix them or just a simple process change.

Case Study: Monthly Report Template

An organization sought to reduce the volume and intensity of project management reporting by adopting a cost tool. The implementation team, comprised mostly of project controls leadership, had the software developer recreate a new monthly report template that leadership had not yet discussed with project teams. Cost engineers began using the system and found that the tool did not offer the ability to group data the way the template required, which had been the reason the template failed to take off as a spreadsheet. To meet organizational policies, they had to export all data from the cost tool to Excel, reassemble it, and re-import it, causing additional working hours, but only for those that used the cost tool. Tool adoption lagged as a result.

Mistake #3: Picking the wrong software

Additional time spend in the "Readiness Assessment" phase also benefits the software selection process. A cursory glance at the vendors sponsoring project controls networking events provides a view to the breadth of options available to organizations seeking a cost tool. This diversity does not reflect the failure to bring to market the perfect tool, but instead the subjectivity of cost engineering. There is more than one way to calculate a forecast at completion, and indeed, a 2019 study found that two popular tools used different methods to different results [2, p.15]. Each tool has strengths and weaknesses that appeal to different industry players. For example, a tool designed for an owner company will de-emphasize quantity tracking and focus on enterprise-wide reports.

Organizations should cast a wide net when evaluating vendors. It may be useful to score each vendor based on requirements, or by what each system does well [3, p. 10]. Some useful metrics may be ease of implementation vs. functionality, or what kind of support the vendor provides when things go wrong. Though it may be tempting to go with a tool with a recognizable name, lesser-known developers may be more agile or adaptable for organizations that want to take a more active hand in crafting a tool just for them [3, p.5].

Mistake #4: Too much Automation

Automation is a doubled edged sword that both leads successful implementations and crashes haphazard ones. Perceived time-savings come at the cost of flexibility and mandatory standardization.

Highly repeatable tasks with clearly defined rules are good candidates for automation. For example, users may always calculate the estimate at completion for a scope of work using the cost performance index applied to the remaining budget to complete. Cost software might introduce a "forecast method" for users to have the tool perform this calculation on the user's behalf.

Tasks that involve expert judgement, even if repeatable, are not good candidates for automation. That said, complexity does not necessarily translate to manual effort. Involving end-users early in an implementation can help guide organizations towards making the right automation decisions.

Case Study: Project Structure Based on SAP Code

An organization sought to automate the creation of a project reporting structure based on its SAP coding, as cost engineers were responsible for maintaining the SAP structure and the organization did not want them to "create the project twice." The software developer tied the creation of WBS and Work Package elements, as well as the initial project estimate, to the finance system's structure and implemented a push-button solution for creating the project. However, cost engineers tracked work at a lower level than SAP in their Excel workbooks. When the end-users began using the system, there was not enough detail to manage reporting needs within the tool. Since project structure was automated, lower levels were not possible and users avoided using the tool as a result.

Mistake #5: Too much customization

Organizations customize software solutions with good intentions, but frequently these customizations push users away rather than draw them in. Often, organizations seek to recreate existing reports and templates without trying the functionality of the tool they adopt. Other times, they try to satisfy one team but fail to consider the experiences of other teams, especially those working in locations apart from the implementation team, leading to poor adoption rates in those areas.

Customizing tools have two main draw backs. The first is systemic: changing the way the tool works in one way may have unintended consequences for other aspects of the tool. The second is isolation: each customization locks the tool into an organization-specific instance that makes it more difficult to apply regular bug and software fixes that may not play nicely with the new customization.

The reality is that no implementation will be entirely out-of-the-box. Successful implementations customize as a last resort, but this does not mean that they do not customize. A complex ERP interface, or a highly customized report, are good examples. These types of changes may help lighten users' workload or achieve buy-in with stakeholders.

During the pilot phase of the implementation, have the team utilize the tool as close to off-the-shelf as possible. Let the pilot team give feedback on what should change and what could stay the same. This not only allows the organization to see if the software really needs a customization, but it also gives the organization fluency with the tool to better describe what they want it to do to the developer.

Conclusion

Cost management software offers organizations the chance to increase productivity by de-emphasizing the daily grind of spreadsheets and repositioning the cost engineer as a trusted advisor to project leadership. Successful implementations involve the end-user from the beginning of the process and rely on their expertise and trenches perspective in guiding software selection and configuration.

Bibliography

1. AACE International, "Cost Control Software Requirements," Recommended Practice 94R-18, AACE International, Morgantown, WV, Latest Revision
2. AbdelRazik, Mostafa, 2019, "How to Calculate Estimate at Completion at a Project Level," EVM-3251, AACE International Technical Paper 2019
3. Caddell, Christopher, 2017, "Framework for Evaluating Project Controls Software Systems", IT.2403, AACE International Technical Paper 2017.

Acknowledgements

This paper would not have been possible without the support of Omega365's array of subject matter experts and gracious support. This work represents the work of the author and is not representative of Omega365's views unless otherwise stated.

Additional Information Requested

Full Name: Jason Michael Kimbrell

Email: Jason.kimbrell@omega365.com

Phone: 1-713-870-0685

Company: Omega 365

Primary Author: Jason Kimbrell

Email: Jason.kimbrell@omega365.com

Phone: 1-713-870-0685

Company: Omega 365

Secondary Author: N/A