Crashing the Schedule in DCS Validation Projects

Schedule-crashing is a high-risk strategy which can lead to greater costs and diminished quality – unless a solid infrastructure is put in place to mitigate the impact of changing the project plan.

By Mark Cupryk and Doina Morusca at Invensys Validation Technologies, and Dean Takahata at Amgen

A ’schedule crash’ represents an effort to reduce the overall duration of a schedule by either adding resources (human or otherwise) or increasing work hours (overtime, weekend working). Crashing is generally done as a trade-off between shorter task duration and higher task costs. It must be determined whether the total cost savings realised from reducing the project duration are enough to justify the higher costs associated with reducing the individual tasks. If the cost savings on a delay penalty are higher than the incremental cost of reducing the project duration, then the crashing is justified.

In a pharmaceutical environment, enhancing project performance during a schedule crash is no straightforward task. The approach is high-risk, as it usually leads to greater costs and diminished quality – unless a solid infrastructure is put in place to mitigate the impact of changing the project plan. Project teams can create a viable infrastructure by using a combination of tried-and-tested project management practices, as well as Lean Six Sigma techniques.

In this article, we examine how to apply project management practices and Lean Six Sigma successfully to a schedule crash in a pharmaceutical automation environment.

CASE STUDY

The subject of this case study is a Distributed Control System (DCS) validation project that formed part of the automation of a new biotech pilot plant. In this project, schedule-crashing delivered a complex, compliant DCS to the plant below cost and without compromising quality.

The project assumptions and baseline were aligned with the client’s corporate policies regarding validation. The validation effort benefited from leveraging a controlled software library based on S88 standards, resulting in the elimination of a considerable amount of unnecessary testing. Once tested, the library could be used multiple times and by multiple sites.

The software was developed using a modular approach, allowing the re-use of the same code multiple times. This approach helped build a testing strategy that minimised redundancy and reduced risk. The project spanned approximately 1.3 years and required a team of approximately 15 people, with an increase to about 30 during the ‘crashing period.’

Project Management Institute methodologies, such as Earned Value reporting, were used. The project involved a high number of deliverables (approximately 7,000 documents) with complex workflows (for remote and neighbouring locations). It involved multiple team
were identified and becoming leaner Opportunities for each activity reviewed and assessed before crashing the schedule. Duration and effort were reviewed and assessed for each activity. Opportunities for becoming leaner were identified and later implemented.

Figure 1: This process map was used to discuss improvement areas for existing activities before crashing the schedule. Duration and effort were reviewed and assessed for each activity. Opportunities for becoming leaner were identified and later implemented.

Figure 2: The key categories were measured using a Pareto chart to determine the most significant issues, providing immediate focus areas for corrective action by the project team (note: actual numbers of issues were factored).

The initial estimate of work was 11 months for 17 people; however, due to the fact that the facility was a new pilot plant, the project team encountered a series of changes that expanded the scope of the project and created re-work. An updated schedule forecasted that the additional work would require an additional six months at the existing planned pace.

THE CHALLENGE

The company was thus faced with two courses of action: either maintain current project delivery speed (hence delaying product launch), or crash the schedule in an effort to meet the set launch date. Given that the launch involved multiple products with high value to the business, the decision was made to crash the project schedule by doubling the team within two to three weeks.

Crashing a schedule comes with a series of problems related to both cost and quality. The dangers of exponential cost over-runs during a schedule crash are very real due to the challenges of managing a larger team, and the potential for unforeseen inefficiencies. Quality may also be affected by the rapid ramp-up of the project team’s size with new members whose training or knowledge-set may not be sufficiently aligned with the project’s specific dynamics.

As part of the transition planning, additional supervision requirements were identified to ensure a smooth transition to the higher productivity model. The new work model involved striking the proper balance between over-time and bringing in new talent, so as not to affect the team with attrition or burn-out. Finally, ideas to maintain morale were discussed and planned. The level of out-sourcing support was evaluated so as to help manage the rapid increase required for the project to be successful.

During the project, the team continuously analysed the process by using Lean Six Sigma tools. They kept focusing on critical path activities to ensure that all efforts were kept on schedule (the crashed schedule), while at the same time preparing for the next critical path activity to ensure success at each step. A process flow diagram was devised with key input and output variables, and value- and non-value-added activities were analysed (see Figure 1).

A cause-and-effect (fishbone) diagram was prepared to break out the root-cause categories of issues that had an impact on the process. Based on these results, the next drill-down exercise was to analyse why the scripts did not match the design specifications, repeating the combination of cause-and-effect and Pareto tools to further isolate high-impact root causes. The following root causes were identified and analysed using a Pareto chart (see Figure 2).

- Not match design – issues occurring when the software script (that is, the documented test case) did not match the design specification. This resulted in corrections to either test script documents or the design specification.
- Test case outline – issues occurring when the test script could not be executed as documented.
Innovations in Pharmaceutical Technology

Figure 3: A snapshot of the activities required to approve test scripts for execution, highlighting the potential for lean improvements in both value- and non-value-added activities.

- Insufficient information – issues occurring when the design specification did not have sufficient information to write the test script.
- Coverage – issues occurring when the structure of the test script did not cover all potential testing scenarios from the design specification.
- Not used template – issues occurring due to formatting errors such as footer/header, page break and numbering.
- Not used guideline – issues occurring due to not respecting guideline information such as phases or interlock strategy.
- File name – issues occurring specifically with respect to improper file naming of scripts or design specifications.
- Design issues – issues occurring in design specification such as missing information or not matching functional requirements.

The team determined that one of the main causes was that the scripts were written according to an older version of the design and had not been updated. A tighter control prior to review was implemented, and the impact of these issues was significantly reduced.

Issues with the least impact on the process were removed from the reviewing activities, further reducing the time needed to review the scripts. (Fortunately, negligible-impact issues were few in number, which made their removal that much faster.) Some of the key process flows were also further mapped according to Lean principles focusing on cycle times, value-added and non-value-added activities (see Figure 3).

As expected, crashing the schedule had a huge impact on project management activities. Due to its high level of risk, crashing should only be attempted when the team is convinced that it is fast-tracking the project, and has placed all activities in overlapping or parallel states to reduce the duration as much as possible.

A counter-measures matrix was built to ensure that all potential risks associated with crashing were mitigated. Some of the key standard risks identified were lack of control over resources, unclear project processes, diminished quality in deliverables and unclear training guidelines for the ramped-up team.

DEALING WITH ISSUES IDENTIFIED

The current Work Breakdown Structure (WBS) was reviewed to identify the impact of crashing on each activity. The detailed project plan was revised for each task in light of the accelerated project time-line. Root-cause analysis guided continuous improvement, and Lean techniques ensured improved workflow, removing delays and non-value-added steps (see Figure 4).

Daily reporting replaced weekly reporting for further control on Earned Value (EV) enabling better monitoring and control of project progress. Key metrics were visually communicated to the entire team, who became focused on surpassing the daily goals. EV provides a tangible measure of the actual progress of work in relation to the planned value for the work. Too often, project teams measure their progress against budgeted hours without...
any consideration for the deliverables. As a result, the project team is not focused on completing the actual tasks in the required time; instead, it gets sidetracked by preparing other deliverables that were not in the original scope, or by enhancing the level of quality beyond the planned scope (scope creep).

The required granularity and frequency are important for EV management. A simple, visual graph of EV reported monthly (see Figure 5), combined with detailed daily deliverable progress in the form of a table, enabled the team to truly focus on completing critical daily tasks. The team understood each task that needed to be completed at the test script level and executed it accordingly. A ‘Red Flag’ alarm system was implemented to rapidly assess the situation and remove road-blocks to ensure success.

As part of the project crash, a revised detailed communication plan was presented to project stakeholders – the procurement, quality, engineering, automation, and validation teams – delivering real-time progress of all critical-path activities.

LESSONS LEARNT

Applying the principles of prototyping the project, the project team sat down and analysed the data after 3-4 process cells delivered. A series of lessons were drawn and action plans were developed and implemented:

- Fix all problems prior to execution in all cloned process cells, so as not to propagate any issues unnecessarily. The project benefited from replication of similar process cells that enabled accelerated coding and testing. In order to further streamline the process, it was critical that the first prototype was tested completely and that all generated corrective actions were considered prior to the replication process.

![Figure 5: The Earned Value (EV) graph tracks project progress after crashing the schedule. The initial budget was maintained – a key attribute in defining project success. (Note: actual hours were factored.)](image-url)
Make sure the Steam In Place (SIP) and Clean In Place (CIP) charts are updated prior to testing. SIP and CIP are processes that impact the majority of process cells. Due to their complexity, SIP and CIP are often subject to multiple changes (daily!). Communication of the modifications is critical to minimise re-work. Prior to executing the scripts, the team performed an informal check to ensure that the latest versions of the charts were used.

No additional technical review is needed for all cloned test scripts as the work is redundant. Since an extensive assessment of the prototype was performed, including all associated documentation, the quality of the test scripts was assured and the team decided that the risk of removing the technical review was minimal.

Remove extensive comments in test summaries not required by the quality team. Quality decided that all comments put in the comments section of the test scripts needed to be transferred to the test summary. A tremendous amount of effort was required to transfer, review, and track the comments. The team decided to minimise the use of comments that provided little benefit to the particular test. Such a reduction enabled savings in time and higher quality deliverables. The test summary was less likely to contain errors and was focused on the value-added activities.

Post-review of test scripts should be done just in time after the execution of the test cases so as to correct any items immediately and not lose momentum. Initially, the test scripts were accumulated, and momentum to complete was somewhat diminished as the Quality team were only able to review them 3-4 weeks later. The project team decided that, once all test scripts were executed, a review and approval session was held with Quality, which then augmented their capacity to accommodate the accelerated review process. All key stakeholders were summoned to a 'review and approve party.' All issues were immediately identified and rectified with high priority. In addition, any new quality requirements were added to the checklists of the team to ensure that other deliverables were appropriately prepared and executed.

All documents must be prepared prior to execution to ensure focus on execution rather than documents or peripherals. The team realised that by having all the tools and items ready prior to the start ensured easy tracking of all test scripts and issues. Having to prepare binders or print documents during the testing sometimes caused minor distractions impacting schedule and quality.

**CONCLUSION**

Crashing should be considered only after fast-tracking (overlapping all tasks as much as possible) has been optimised. Crashing a project requires rigorous movement towards effective control of the project through instructions, statistical monitoring of performance (control of work processes) and – when possible – mistake-proofing (see Figure 6).

It should be recognised that crashing has become a way of life for validation projects – and project teams are embracing accelerated timelines as a normal part of their projects. Solid project management understanding and know-how ensure flexibility to adjust the course of the project efficiently and effectively. The team must also be ready to cope with a high degree of stress due to heightened expectations. Hence, the leadership must motivate and communicate much more regularly. A positive work environment is paramount for the health of the talent engaged in the overall project goal.

Due to the pharmaceutical industry’s continuous focus on cost, businesses are constantly trying new ways to derive maximum value from each dollar invested in projects. Solid project management techniques combined with Lean Six Sigma provide the required framework to align metrics with execution for success – whether crashing or simply executing a project. This scientific methodology translates into less effort and greater control over the project during execution.

The authors can be contacted at mark.cupryk@ips.invensys.com, doina.morusc@ips.invensys.com, and deant@amgen.com

---

**Figure 6:** Diminishing the effort for validation activities includes moving towards tools for more effective deployment of information and communication. The project was managed with the vision of moving towards mistake-proofing and statistical control when possible.