AVOIDING AND MANAGING PITFALLS OF CONTROL SYSTEM PROJECTS

A. Jesenko*, G. Pajor, Cosylab d.d., Ljubljana, Slovenia

Abstract

Cosylab is the leading provider of software solutions for the world's most complex, precise, and advanced systems. We also provide software products and services to the largest medical device manufacturers and cancer centres worldwide. In our 20-year history, Cosylab has worked on hundreds of multi-year projects worldwide.

Software projects run the highest risk of cost and schedule overruns. On average, large IT projects run 45 percent over budget and 7 percent over time.

There are some common pitfalls when defining and executing projects in the Control Systems field. This article presents some concepts on how to address them, such as: having a clear definition of a project's scope, budget, and timeline; doing project risk management; having a clear division of responsibilities; and having a project sponsor from the management.

INTRODUCTION

Control system development is an engineering discipline like many the others, often with a quite complex life cycle. Jumping right into prototyping and coding can bring some initial results quickly but is very prone to a myriad of risks and undesired results, such as an unclear scope, a lack of acceptance criteria and vaguely defined responsibilities. This can consequently manifest in never-ending development, an unnecessary multiplication of effort and non-existent or sub-par documentation.

Control systems are an engineering discipline like all the others, but with an even more complicated cycle:

1. Write specifications
2. Architecutre
3. Design
4. Prototyping – probably the only fun part
5. Test procedures
6. Implementation (coding) – the only software part
7. Documentation
8. Testing
9. Debuging
10. Acceptance

Each control system project has the following entities

- Consumer: the entity that is ordering or is otherwise interested in getting parts of (or the whole) control system.
- Supplier: the entity that is supplying parts of (or the whole) control system to the consumer.

A control group can have either of these roles; it can be the consumer ordering the control system project from another entity and it can also be the supplier supplying the control system to their own or another institution. One part of the control group can also assume one role, and another part of the same group can assume the other role. In all these cases the basic concepts described in this article remain the same.

PROJECT PLANNING

Quite intuitively, project planning is a crucial process for any project. The better you define the project and how it will be closed in the beginning, the less unfinished scope will remain in the end. This can result in a lower total cost of ownership. Remember that there will be some work after the project is complete and this work should also be planned for. In any case, don't count on defining the project scope while the project is ongoing (see also "What about Agile?").

The basic elements that define a project are:

1. the scope (what will be done, and who will do it),
2. the budget (how much will it cost),
3. the risks (what can go wrong).

The first steps in creating a project plan are to define the scope of the project, and to find out the time and budget constraints. After that, the scope is broken down into a "work breakdown structure" (WBS), and the risk management process is started.

Project Scope

The project's scope is usually documented into a statement of work (SoW). This is an important document with details about:

- What was agreed would be done, and by whom.
- What will not be done in the scope of the project.
- What are the deliverables.
- How acceptance of deliverables will be done.
- What assumptions were made.

Project Timeline and Budget

A rule of thumb to estimate the budget for an accelerator control system is to take 10 - 15 % of the total budget. To start planning a more detailed timeline and budget, work should first be broken down into a work breakdown structure (WBS). An effort estimation should then be done for each task in the WBS. This is a good way to gain knowledge about how much effort from people of each profile you will need for the project. Note that the effort estimations will only be accurate if they are done by persons experienced doing them.

Once we have a WBS done, we need to know when we will be ready to start each of the tasks. Now it's useful to create a critical path. By looking at a critical path, we can tell prerequisites and a timeline for each of the tasks and the project in general.

Then we can set some milestones for the project. It's a good idea to plan monthly milestones, in which the team

* anze.jesenko@cosylab.com

delivers or shows something. This way it's easier for both the team and the manager to objectively see how well the project is doing, whether there are some issues that must be resolved …

Project Risk Management

While project risk management is widely recognized as a beneficial activity, it's also widely forgotten about during actual project management. Even when it's being done, many times a risk assessment is done at the beginning of the project and then later forgotten about during the project. For a project to be a success, it is necessary to monitor the risks and act on the spot. To be able to do this, the risk manager should document measures and procedures for each identified risk:

- A measure that reduces the chance of the risk materializing.
- A procedure to perform if the risk materializes, a "missed approach procedure" [1] in aviation terms. This should be done so that you are not required to think in panic.

What About Agile?

Agile software development is probably a good idea, especially for internal control groups. But it is worth understanding that with large control systems projects, there are many things that must be agreed in the beginning of the project, or even before that. Otherwise, some of the participants of the project will not be able to work and deliver in time, and this will cause delays. But if critical specifications and interfaces are agreed up front, then doing the software development the Agile way is ok.

In short, for internal control groups Agile is a good idea and we recommend it, but not as a replacement for the definition of the project.

PROJECT EXECUTION

Rather than giving lengthy instructions on how to execute a project, we are focusing in this article on a few ways how a project manager can avoid the most common pitfalls of control system projects.

Change Control

Uncontrolled change is probably the most common reason of failure of control system projects. When we talk about change, we talk about change in any dimension of the project definition (scope, timeline, or budget). While some changes in scope are necessary in most projects, it is worth understanding that an increase in either scope or timeline results in an increased cost of the project. To protect the success of the project, any change must be accepted in a controlled way.

The Agile methodology inherently handles change by planning work in smaller packages. When not using an Agile methodology, a change control procedure must be established [2].

A summary of a basic change control procedure is:

1. A change is identified and reported by anyone involved with the project. The importance of the change is estimated.
2. Impact assessment is done. What is the impact on the budget and timeline? Are there any risks involved?
3. Review and approval of the change is done by a decision-making body. Changes with a high impact and low importance are usually rejected, and changes with a low impact and high importance are usually approved.
4. The project specifications are formally updated.
5. The change is implemented.

Single Source of Truth

It is important that a single source of truth is established for each project. This is some location (like a cloud share) where anyone involved in the project can access and add their latest versions of specifications, meeting notes, source code …

It does not matter much what exactly the implementation of this system is (like a Git revisioning system for source code, a document management system for documentation). But it does matter that every participant uses it and that it is clear to everyone that the truth is only what is stored in the system.

Regular Communication

Regular communication is important to build and keep good relationships, to be sure that everyone understands the project the same way, to provide a place where everyone can report issues and voice concerns …

Once the project has started regular communication paths have to be established. All stakeholders must be considered (team, management, outsourcers …), and the type (sync meetings, mails …) and frequency of communication must be agreed. During sync meetings, it is a good practice to note down meeting minutes and action items and store them in the single source of truth. Action items should be a regular part of the agenda. If an action item is not done on time, it should be discussed what the reasons for this. Participants of the meeting can offer insight and help with any issues. Issues that cannot be resolved can be escalated.

Escalation Path

Sometimes an issue cannot be resolved within the project team. To keep relationships between team members intact, it's a good idea to have sponsors from management assigned to each project. A "project steering committee" should be established already when planning the project, and a meeting schedule should be agreed. The steering committee should be composed of the project manager(s) and persons with decision power, e.g., the people responsible for the project resources like personnel and budget. During a steering committee meeting the project manager presents the status of the project and the escalation requests. The committee provides the project manager decisions on the course of action for each escalation.

PROJECT CLOSURE

To start closing the project, first revisit the SoW document to check that everything that was planned for was done. Acceptance testing of deliverables should be done whenever possible to check that the goals of the project were achieved. The consumer (entity that is ordering the project) must agree with both the testing procedure and the results. Once tests pass without dispute, the deliverables are accepted.

But project closure isn't just about the completion of acceptance tests. Consider and plan also for how the transfer of knowledge will be done, who will maintain and upgrade the solution, and who will provide support to users and the maintenance team. Take some time to think about the lessons learned and document and discuss them with the team· Look at the initial effort and timeline estimates and compare them to what happened.

And don't forget to buy the team some pizza and beer.

CONCLUSION

Due to control system's specific role, it interfaces almost all other systems. This consequently means that it relies on information and deliverables coming from these systems and for the most part, delays on these prerequisites also cause a delay of the control system project. This is further emphasized by the fact, that the control system is usually one of last things to be finished and can lead to erroneous conclusion that the control system is responsible for the overall delay. To avoid such an unfavourable situation, good planning and adequate communication with all relevant stakeholders is vital – on top of well managed project execution of course.

REFERENCES

[1] Missed approach, https://skybrary.aero/articles/missed-approach

[2] Change control, https://en.wikipedia.org/wiki/Change_control