



Metrics for Successful Automation Project Management

Most automation engineers spend their days controlling manufacturing processes, but spend little or no time controlling their project schedule and budget. This is not surprising as they have little or no involvement in budget reviews or project management activities, other than the schedule, for much of the duration of the project. However, when the project is suddenly over budget, automation engineers have more involvement in finances than they want, and most of that involvement is not good. They are likely to be dragged through the dirt. But if the project has been successful and the control system does what it is supposed to do, Automation Engineers get little positive feedback or reward for this achievement. Such is the plight of the engineer.

As the economic situation is getting tougher, most companies are requiring their engineering staff to do more, take on more responsibility and spend even more hours just to keep things moving. As a result, automation engineers will be tasked with much more project control. This article will review some recommended practices, which if they are implemented early in the project, will allow the automation engineer to be as successful from a project management perspective as he is on the technical side.

Like control systems in automation, a project control system must be implemented to control costs, ensure quality and manage schedules. Successful controls require measurements. In the PM world those measurements (inputs/calculated values) are called metrics and the greater the frequency of measurement, the more accurate the control. All measurements need to be compared to an expectation (set point). If any deviation is noticed then action is required (modify the output).

Capers Jones indicated that only “**10%** were deemed successful in that they achieved their schedule, cost, and quality objectives.” (Jones, October 2004) This means that 90% of all automation projects (or the automation component of a larger project) are not adequately monitoring their progress against an expected goal and do not have adequate tools and processes in place to do so. A set of measurements should be chosen early on depending on the key business drivers of the company, and processes and tools put in place to capture the data and provide metrics against these measurements and targets.

Metrics are indicators which are representative of a project’s ability to meet its intended goals. If carefully selected, they can also predict the results of not taking any actions. We will break metrics down into the following 4 subject areas:

- Schedule
- Cost
- Mixed
- Quality



Schedule Metrics

Schedule is the main subject of discussion with management as automation is usually the last to be completed and could introduce an overall project delay. In most cases by the time the automation work can be initiated, the project is already late and management is already scrutinizing the schedule. Under those conditions the automation engineer needs to think constantly about the schedule. Cost is not forgotten, but as automation is usually less than 15% of the overall project cost, automation is rarely the cause for major cost variance.

Schedule performance can be measured relatively easily and can be really simple, or extremely complex. The simplest method is just to use the visual tools provided by your scheduling software package.

Figure 1 below uses Microsoft Project®. In the Gantt tracking mode, the software can graphically show progress against an actual task (below). A dotted line indicates the current date. This allows the automation engineer (or automation project manager) to quickly determine if the project is behind, ahead of, or on schedule.

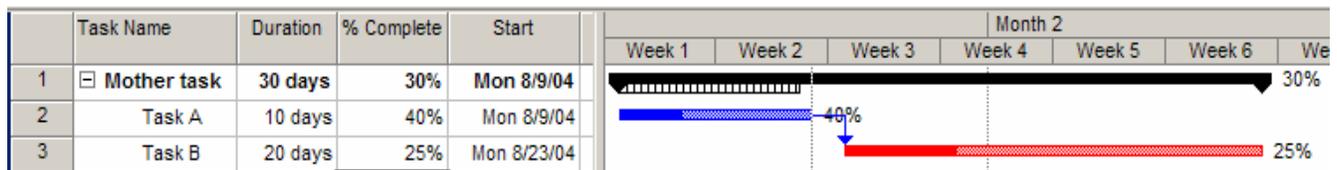


Figure 1—Example of a Project Schedule

It is important to understand that this intuitive type of performance measurement has major limitations. In this example, Task A is 1 week late, but Task B is ahead of schedule. The automation engineer has no idea of the overall status of the project, productivity issues, identification of root cause, nor any leading indication of future performance. So while intuitive analysis has a place in project metrics, it should be used in conjunction with other indicators.

The next step is to look at each activity and use a more qualitative approach. Most Project Managers will ask their staff for an estimated percentage completion (e.g. 50%). This method often leads to overstating progress, because you are relying on the resource executing the tasks to tell you where they are, and human nature is to not give bad news. It seems that 80% of the task effort occurs after 90% of the progress has been reported using this method. This method can only work if the activities are broken into small enough components so any estimate error will not affect their rollups.

When possible, a quantitative approach would be more accurate. This approach is based on countable items. For example, a progress measure may be based on the quantity of drawings completed versus the expected total. These methods are extensively used in the construction world where you can easily count the quantity of items installed against the estimated total. Unfortunately this is not always applicable. The best example is the creation of a document. You do not know how many pages a user requirement will be until you are finished. In this case a hybrid approach is recommended by defining some rules early in the project.



In our document creation example, the rule could be that a document is 50% completed when the first draft is issued for review, 75% when the comments are back, 90% when the documentation is issued for approval and 100% once it is approved.

All sub-activity progress is added up based on the weight of each activity, and overall progress is calculated and compared to the expectation. A very simple but very effective method to look at schedule progress is a progress chart, as seen below in Figure 2.

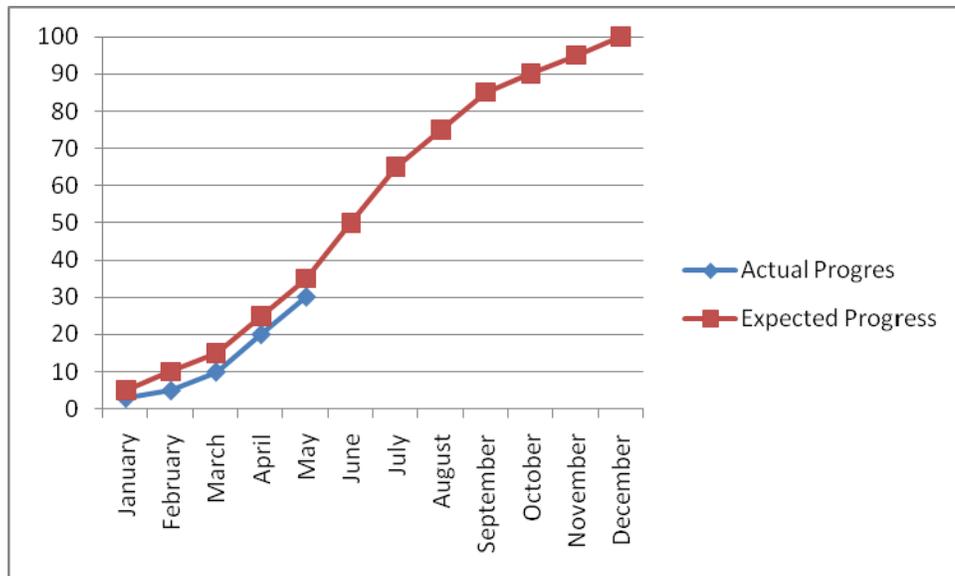


Figure 2 - Example of a Progress Chart

Reporting the actual progress against the expected curve even allows you to measure the delay. In our example, it appears we have a ½ month delay. This sort of metric is among the techniques found in Trending Management. It is a very popular automation engineering concept and we are using it every day to control process variables. By itself, this metric just give us one side of the story. Indeed we are 1 month delayed, but is that important? If we have a reasonable explanation (e.g. project has started 1 month late, or the overall project is delayed by a month due to the delay of the delivery of the process description, etc.) the delay might be acceptable. Furthermore, we need to find out how costs are trending to give us a better understanding of the entire situation. If we are late, but have been under-spending then the easy solution is to increase our project speed or, as it is commonly referred to, crash the schedule. However if we have been spending the expected budget and are late, this means we have an efficiency problem. Before expanding too much on this let's talk about cost.



Cost Metrics

Every project starts with an estimate. Once negotiated with management, this estimate becomes your budget. As the project evolves, additional information is discovered and further estimates are produced. This is an extremely important process and we cannot emphasize enough the need for this re-estimation or re-budgeting process at each phase of the project. In any case, for the purpose of this article, we will call the revised budget the "actual budget." Another standard activity is to provide management with an expected cash flow. From a financial perspective this is an important activity, but it also can be used as your cost expectation. (See purple line in Figure 4.)

As the project progresses, you monitor the actual spending which becomes your actual cost. This cost divided by your actual budget is the % spent, which can be compared to the expectation to provide an excellent guide to where you stand cost-wise. However, with respect to progress, cost comparison alone is limited. A trend of the cost spending is a useful tool. Actual cost monitoring has several constraints. The major limitation is the availability of the data in real time. Indeed, most project accounting systems have major delays, usually due to the billing/payment system. It might take more than a month before an invoice gets into the accounting system. The best way to resolve this problem is to use a parallel real-time approximated accounting system. Invoices are accounted for as spent as soon as the work has been performed. Another inadequacy with corporate accounting systems is that they do not have the granularity required to adequately monitor project progress. If you have contractors on your team, their invoices will typically get rolled up as one or two line items in a system such as SAP. This high level summary provides no insight into the detail that is required to correctly manage projects.

Mixed Metrics

As we have noted previously, focusing on only one metric (cost or schedule) does not tell you the complete story. It might be acceptable to be under spent because you have not performed the work expected. You might be overspent, but you are way ahead of schedule. Neither of those scenarios is necessarily bad.

Just as having both eyes open gives you depth perception, combining both spending and progress together gives the automation engineering better perception. Simple methods to compare your results in terms of schedule and cost together can be easily implemented and will be much more informative. In Figure 3, we compare the progress vs. spent for each automation sub-discipline.



WBS Element	Activity Description	% Complete	% (Actual/Budget)	Difference + = worse than expected
	Project X	30%	25%	-5%
1	Project Management	32%	33%	1%
2	Instrumentation and Control	40%	35%	-5%
3	Hardware Engineering	42%	32%	-10%
4	Software Engineering	25%	30%	5%
5	Construction	5%	7%	-3%
6	Commissioning	1%	2%	1%
7	Training	8%	5%	-3%

Figure 3 - % Progress vs. % Spent

The example above indicates that the project is performing well overall, but the software engineering team is in trouble. The major problem with this method is that the work is following an S-curve and it takes a lot of money to do very little at the beginning and the end of the project.

The Earned Value method provides a more structured approach for this sort of comparison. It compares the work that has been performed to the expectation, and the cost spent to achieve this work. The metrics (or indices) used in this method are the Schedule Performance Index (SPI) and the Cost Performance Index (CPI).

SPI in its simplest form is defined as the quantity of hours “earned” as a ratio to the expected (or planned hours) at the time of the calculation. Earned hours are calculated by multiplying the % progress by the total effort (in hours) for the task. For example, if you have performed 60% of the work on a task of 10 hours, you earned 6 hrs. If the expectation were to spend 5 hours to get you there, then the index is $= 6/5 = 1.2$ (a value > 1 is good and < 1 is bad).

CPI is similar, but focused on cost. It is defined as the ratio of the earned cost (% progress x budget for the task) to the actual cost. In our previous example, if the budget for the task for \$100, then you have earned \$60. If you have spent \$70, then the ratio is $60/70 = 0.86$.



Graphic tools are often used to present several dimensions at the same time. In Figure 4, expected cost, actual cost and schedule are shown together.

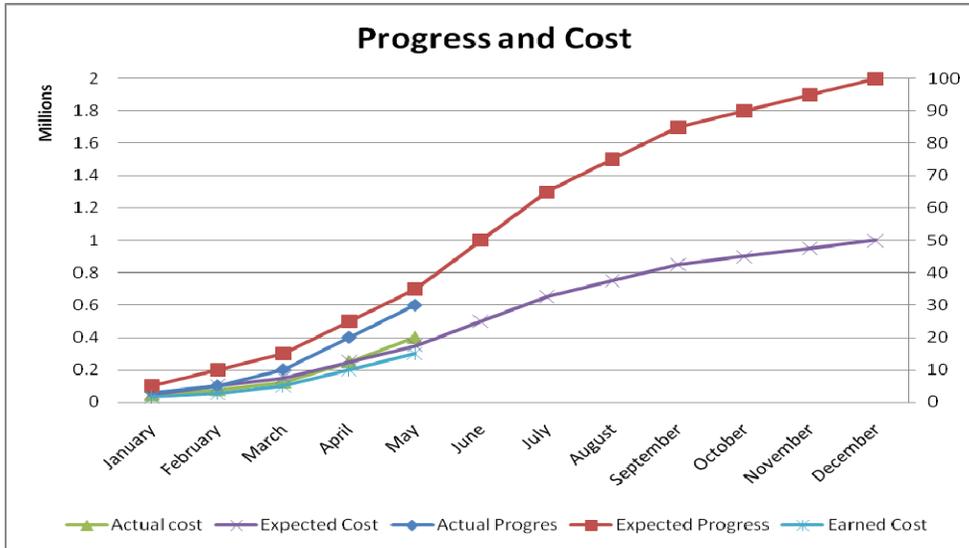


Figure 4 - Multiple Metrics Chart

Quality

None of the metrics previously discussed take into account an extremely important area for automation engineers: quality. What good will the project be, even when it is on budget and on schedule, if at the end of the day, the control systems do not work as required?

Measurements such as quantity of defects re-work in hours, retesting in hours, number of scope changes, etc., represent the quality of the work performed. By themselves they are not really interesting; they need to be put in a more measurable context such as a ratio:

- Quantity of defects per 1000 lines of code (or module configured)
- Re-work vs. work as a %
- Retesting vs. testing as a %
- Cost associated with scope changes vs. total cost per type of changes.

Conclusion

Containing costs and managing budgets will become an absolute necessity in the years to come and automation engineers will need to develop and implement methods to achieve these objectives. Using appropriate metrics is the only effective way and can be relative simple if well thought out from the beginning of the project. We believe the methods presented here will help the automation engineer on his journey.