Smooth Product Launches

cutting through the noise

WHITEPAPER

Tony Pashigian

9/1/11

Smooth Product Launches

cutting through the noise

Program Managers have to orchestrate the resources of the enterprise to deliver on contractual obligations. Through schools, online training or organizations that train the discipline, anybody can learn the mechanics of being a Program Manager. However, the "secret sauce" makes success more likely. It is a combination of the experience succeeding and failing, people skills and business acumen. This paper provides a roadmap for Program Management success.

Some Assumptions for this Paper

This paper assumes that the product will be some combination of purchased and manufactured (vertically integrated) physical components. A similar approach can be taken for IT projects, but some of the references and examples do not directly translate to that type of industry.

Also, I use the term title "Program Manager" as interchangeable with "Project Manager" and "Portfolio Manager". While the technical differences exist in formal PMI, PMP, and PMO definitions, I believe the required skill sets and boundaries of responsibility are the same.

The comments in this paper apply to either a de-railed program turn-around or a new program kickoff. I have been assigned to more programs that were in distress than I was assigned that was a new award of contract. So, I have had turn-around success by regrouping, reengaging and resetting.

> Regroup: a series of one-on-one and full team discovery meetings to learn the true status of each team member's contribution to the program execution. This is not the time for a performance review.

> Reengage: make sure you have buy-in on the program deliverables and expectations. Does everybody see the same lighthouses and destination?
>
> Reset: the old plan was the old plan and it wasn't working. So, set the new program schedule and new individual expectations for the team members.

For the purposes of this paper, *launch* failure is anything that doesn't meet expectations getting the product to your customer's receiving dock. It does not include *products* that were doomed to fail (not sell) because there was no market, like purple ketchup, the DeLorean, New Coke or the Ford Edsel. So, this paper considers product launch failures to be a company's inability to get the product to the customer, as planned.

The Problem

Companies in every industry struggle with new product launches. The process of converting the voice of the customer to criteria and then the product development cycle of concepts, solution, development, verification and implementation is well defined in academic material on Program Management. Also, benchmark examples exist for each of the functional disciplines like Quality, Design, Engineering, Manufacturing, and Purchasing. Yet, somehow, all of that rarely comes together for a smooth launch.

Programs, with products of any complexity at all, provide many opportunities for Cross-functional teams to fail. There are huge volumes of content in PMBOK and PMI publications to provide the academic roadmap to Program Managers. Individual companies have Program Management Offices (PMOs) to provide specialized guidance based on the particular products and organizational structure of the enterprise. Yet, cross-functional teams grasp defeat from the jaws of victory time and again.

The small percentage of successful new product launches use most of the same tools as those that fail. The difference is in the "secret sauce" of Program Management. It has to do with filling in the gaps between the pieces of systems and academia in a way that embodies leadership and people skills. This paper defines the secret sauce.

Product Launch Failure Has Many Faces

Before you think this subject doesn't apply to you because you launched on time, we should cover a couple of different symptoms that your program team fell short of success.

There are a few different types of failure to launch successfully. The easiest to identify is when a launch is late. If a program launch is late, then the customer is waiting for your company before they can sell their own product. This affects their business plan, their customers, their revenue and, most likely, your relationship with the customer.

If all of the components for the launch arrive on-time to be assembled to meet the customer due date but there is no way for the assembly plant to assemble them in such a way as to meet customer quality expectations, it was not a successful launch.

Sometimes you are able to launch on time, but you have to add extra steps in your manufacturing process (product containments), which inflates cost, or you can just barely get the product delivered on-time with the addition of expensive premium freight somewhere in your value stream.

So, in summary, any substantive compromise on cost, timing or quality is an unsuccessful new product launch because it prevents the customer from meeting their goals to be better/faster/cheaper.

While some products manage to launch on time with the expected quality level and within budget, "the typical failure rate of new product launches can be anywhere in the 85% to 95% range"[1] according to Forbes.

The General Program Management Approach – The Basic Steps

1) Learn the Scope

The Program Manager's first assignment is to learn the expected outcome of the project. A source that should be reliable for that is the Statement of Work (SOW), assuming that the Statement of Work accurately reflects the voice of the customer regarding project outcome (it's a huge assumption, which could be the subject of another paper) and any safety standards or regulations.

You need the big picture. Many programs (projects, if you prefer) fail because the Program Manager, and hence the whole team, focus on (and actively manage) only a portion of the program instead of the full project. It's easy to take the bait and engage in the details of the current program activity. After all, your team members are uncovering all of the issues and opportunities based on their current work.

Leave the tasks of the functional groups to the functional groups. Your role is to help them acquire any program parameters they require as input and to clearly define the output the upstream functional groups require. They will try to pull you into their troubles and you should be willing to completely engage with them once you're sure you've tended to the big picture.

So, learn what the customer believes the scope is and then compare it to your management's perception. Close any gaps that exist, and then ensure that your team completely understands the Statement of Work (SOW) in terms of cost, timing and quality.

2) Planning at a Macro Level

Identify, at a high level, what tasks will have to be completed to meet stakeholder expectations. Your team should be involved in this process to ensure that their input is considered and that you have their buy-in on the eventual plan. A cross-functional team brainstorming session is an effective way to accomplish this task.

The goal is to identify all of the different tasks, an estimate of their durations and having them logically sequenced based on dependencies.   To establish confidence in the task duration estimates, have the functional group that will be performing each step prepare a reasonably detailed plan for their lower level tasks. Their responses will give you an indication if they've thought through what is required of them and if they have been either optimistic or overly conservative in their estimates. I have found that creating a detailed plan for, and attempting to dictate it to, a functional group is not received well and meets with resistance.

Program Management excellence is not demonstrated by the size and complexity of a program schedule, but rather the successful execution of the project. I go through more of the actual steps that I recommend in creating a timeline in the next section, "The Secret Sauce".

3) Cross-Functional Team Follow-up

Determine a cadence for regular follow up with the team. Do not assume that no news is good news. Your simultaneous development team, or cross-functional team, meetings should be focused on the important factors in the program *at that* *point in time*. Some programs can afford the luxury of a weekly meeting while others, either due to size or complexity, require a 25 or 30 minute daily "stand up" meeting.

The agenda should be:

a) Did we do what we said we were going to do yesterday?
b) If not, how will we recover?
c) What needs to be done today?
d) What are the deliverables within the next 90 days? And,
e) A re-cap of the important milestones for this whole project.

The idea is to make sure that the current tasks are being executed as forecasted while you help the team keep the horizon in mind. It's important for the cross functional team to anticipate, and plan for, upcoming group-to-group handoffs. An example is when Engineering is scheduled to have drawings of parts available that Purchasing has to go buy in their supply chain. In that example, Purchasing can be made aware of the impending requirement weeks in advance. They can contact their suppliers and tell them what to expect. They might even invite them in for a preliminary look at the drawings. If the team were to wait until the drawings were complete, valuable weeks would be lost while Purchasing got their process started. It's like a relay race. Runner 2 has to get started running up to speed so runner 1 doesn't crash into them during the baton hand-off.

4) Disregard the noise

There are two types of noise that your program team will experience: internal and external.

External noise will be provided from sources like management, customers and suppliers. They will constantly be trying to influence what you and your team consider important and urgent. It will be up to you to constantly assess inbound noise for it's impact to the critical path of the program and your team's ability to successfully, and smoothly, launch the program on time.

Internal noise will come from your team members. There will be personality issues, differing work habits and different ideas about work/life balance that will all threaten the stability of your program team. Your job is to define the culture of the team. There is more on culture in the next section, "The Secret Sauce". It helps to point out to the team that there are plenty of external sources for pointing out every little issue with your project and that your fellow team members need to "have each other's back's" to survive.

The successful Program Manager will cut through the noise and help the team, in the fast paced program environment, stay focused on what is required for successful execution.

The Secret Sauce – The Ingredients for a Smooth New Product Launch

People and Team Culture

The combination of your team members and the prevailing culture will serve as the infrastructure for your project. How a Program Manager deals with people is the difference between success and failure as a team leader. On the last program that I managed, the key to our success was in the "mission first, people always" team culture that I instilled in the team. The "people always" wasn't something people were used to. For example, instead of starting with a list of dates that people had to travel to different locations to get the job done, I started by asking for their birthdays, spouses and kids birthdays, anniversaries, already planned vacations, and so on. Then I suggested that the rest of the team help to protect those dates as "no travel" dates for each team member. They started out very disorganized and cynical, but they wanted leadership. Once they got leadership and respect from me, and the rest of the team, everybody's performance went up as individuals. An executive made the comment "Tony's people would throw themselves on a grenade for him" in my performance review for that program.

As for the "mission first" part, everybody has to buy into the non-negotiable obligation to meet customer expectations and contractual obligations. The Program Manager owes the team clear scope definition, a credible plan, sufficient resources and all of the required tools and skill-sets to deliver on the commitments. Not having those things can instantly create "non-productive" team members because they can't envision success. Your job is to provide the vision.

Performance issues are very real and they have to be dealt with. If you have the scope, plan, resources and skills and team members simply aren't doing their job, you must be permitted to elevate that to their functional managers or your senior management. Program Management is a series of hugs and kicks. Without some basic expectation for performance, the kicks just turn into nagging without follow up, which is bad for the morale of the whole team. If you are running the program well and you have reasonable management, I would suggest that you, constructively, elevate performance issues to be dealt with outside of the team.

When you are assigned to the team, the individual team members likely consider themselves just that: a group of individuals. That individual perspective will guide their behavior. Your job, with respect to culture, is to emphasize the excellence of the team in terms of conduct, standards and output. I tell every team that I have led that I want to be perceived by outsiders as the Navy Seals of the organization. They always find it amusing at the start of the program. However, after we start the basic blocking and tackling and start to get organized, they start to feel less afraid of failure and even a little proud as they start to "get it".

Be sure to celebrate the successes and learn from the failures. On my teams, nobody "gets killed" for a mistake. We simply correct the mistake and move on. It not only instills a find-it-early-fix-it-early culture, but you also build up the person that made the mistake when they are feeling vulnerable.

Final comment on people and culture: Everybody's style is different, but I recommend you make it a fun environment for your team. Relentless accountability, while required, can wear on a team.

Program Milestones

When a program is scheduled to launch 24 months into the future, it is very easy to let up on near term milestones. For example, Engineers have to define the product before Purchasing can buy it. A design release date emerges from the planning basic steps, earlier in this paper. Engineers are often in the perpetual pursuit of excellence and are willing to miss their design release commitment by a week to keep the improvements evolving. As Program Manager, you need to instill the discipline that the design release date is Engineering's launch and hold them accountable to hitting the date with a product of a quality level that is satisfactory to meet stakeholder expectations. If every group in the cross-functional team takes it upon themselves to consume an extra week of time, will your project still launch on time? Tracking to mid-program milestones is an effective way of conducting performance analysis on your project and provides substantive metrics against which your team's performance can be assessed.

Program Management Jargon

I've read Kerzner books, PMBOK and studied PMI content and am impressed by all of the subject matter to be learned about the discipline of Program Management. I've also known some people that knew an intimidating amount of Program Management jargon that, no doubt, came from serious studying about Program Management. My experience, however, has been, that these people can be a source of power in a discussion about the topic of Program Management but they lack the practical skills to relate to a cross-functional team and launch a program, successfully.

"Program Managers neither create nor destroy information. They merely orchestrate the cross functional resources to deliver on the contractual obligations of the enterprise."

These people can create forms that nobody can understand and create timelines that are so complicated that nobody can follow. Then I've seen them stare down their noses at the people that are really doing the work as if to say, "What's not to understand? I'm from Corporate, and I'm here to help. You're not as smart as I thought you were."

For my whole career in Program Management, I have said, "Program Managers neither create nor destroy information. They merely orchestrate the cross functional resources to deliver on the contractual obligations of the enterprise."

The Program Manager's only marketable output is management of the program and providing leadership. There is no product we can put a price tag on and sell. We have to guide the team, anticipate risk, mitigate risk, balance resources, etc. To dilute the functional team member's time with a bunch of self-serving program management jargon does nothing to launch a program.

Cross-functional team members just want to be led. They want somebody to keep the end goal in front of them and have meaningful touch-points for assessing project status and direction. They do not want to learn how many fancy Program Management concepts you can articulate.

Program management "expertise" must exist. It has its place in the Program Management Office (PMO). The PMO is the place where the Program Managers go to sharpen their saw and make sure they have the right tools in the toolbox. However, I believe that when a Program Manager is deployed by the PMO, they need to spend less time talking about the tools and more time demonstrating what the tools can produce.

If the carpenter you hired to build your deck spent all of his time trying to impress you with his special saw instead of using it to build your deck, would you think he was helpful? The tool is of no value other than the product that it produces.

Credibility

You are constantly being watched by your team, as a Program Manager. They are looking to see if you have the courage to make a decision, if you have character and if you know how to prioritize. They want to know if you are logical and if your actions match your words.

Cross-functional team members want to be led. As Engineers, Designers, members of Purchasing, Operations staff, Sales, and Quality Engineers, they are prepared to bring excellence and sweat to the cross functional team, but they want to know if you know how to orchestrate the team to deliver on the obligations of the enterprise.

The inexperienced Program Managers that I have had working for me in the past try to establish their credibility with functional group knowledge, politics or frequent lessons on PMBOK or other program management tools. The team quickly learns that there are no program management tools that can fix any problem they are experiencing. All the team hears is: "Ready, FIRE!, aim". What the team needs is the Program Manager's most important tool: leadership.

When the team see's the Program Manager accept blame for the team and not single out an individual, the Program Manager earns credibility (of course, someone that just can't get the job done needs to be dealt with, but I'm talking about the "nobody gets killed for a mistake" approach to leadership).

Also, when the Program Manager protects the team from unreasonable customers, they earn credibility. When the customer insists on a solution but the Program Manager explains that the team must analyze and decide before it can implement, the Program Manager gains credibility. When the

Program Manager leads with passion and uncompromising integrity and character, the team will trust their leader based on credibility.

Programs are not long enough to lose credibility and regain it. So, protect this valued resource at all times.

Ready, aim, FIRE!

Timelines

Some Program Managers create complex timelines with hundreds of tasks and dependencies that are so complicated that the team can't support it because they don't understand it. It also takes considerable energy to maintain or change these complex schedules. It has been my experience that these Program Managers have a difficult time trying to figure out why the team can't follow a schedule that is so thorough and logical, in their opinion. They spend considerable energy trying to teach the team the timeline instead of executing the plan.

I prefer to take a sum-of-the-parts approach. I ask that each functional group prepare a schedule for their contribution to the program. Then I use larger blocks of time to represent their schedule and logically sequence each of the individual blocks. At different times in the project, a more detailed timeline might be required to coordinate many groups in the cross-functional teams coming together with concurrent tasks or hand-offs. I recommend generating them when they are needed and using them as a tool to lead the team.

In terms of creating the schedule, these are the steps that I go through:

a. Assess for "can" versus "will" timing

   When asked how long a particular task will take, program team members often respond with a duration within which they "can" get it done. That just means it's physically possible to get it done in that period of time. However, when you ask when "will" they get it done, they consider the other projects their working on, equipment availability and capacity and other potential inhibitors to them hitting the "can" timing. You must know when the team *will* get their task done rather than when they *can* get it done.

b. Logical sequencing (dependencies)

   This is pretty self-explanatory. You can't bake a cake until you mix the ingredients and pre-heat the oven. Make sure your program schedule comprehends what must be completed before another task can begin. This is the time to apply lean thinking to eliminate waste in the process that your team will use to deliver on program expectations.

This first look at the logically sequenced tasks, that have been assessed for “can” and “will” should, be made without considering customer milestones. You now have a plan that is logically sequenced and your team can deliver without compression or expediting. This is important because compression and expediting are extraordinary and cost either time, money or quality. As you will see, below, the team might decide that extraordinary actions would be tactically or strategically prudent. However, looking at the schedule without extraordinary measures helps the team, stakeholders, management and the customer bear the cost and risk of extraordinary measures with “their eyes wide open”.

A common mistake is to start with the customer milestones and then cram all of the required tasks between the milestones without regard for whether it is even reasonable to assume that the team can do it or resources are even available. This is a very effective way to lose credibility with your team by *you* making promises that *your team* cannot keep.

c. Ability to meet customer milestones

After a project plan that has emerged from the schedule building process, you have to overlay customer deliverables to assess whether extraordinary measures must be taken to meet timing. This might include things like creating prototype tools and parts, paying for lead-time compression, premium freight or containment.

If your initial plan starts with customer milestones and cross-functional team tasks won’t fit with standard timing, then the Program Manager has a risk to mitigate. Simply insisting that a functional team compress their timing means that your team has just assumed the risk of compression. That may be the tactical path your company chooses to take, for strategic reasons, but they should go into it with eyes wide open. Management might have to authorize additional resources. Sales might attempt to charge the customer more for extraordinary measures. The novice Program Manager will simply absorb the disconnect and lose credibility with his/her team.

d. Cross-functional team member buy-in

It is imperative that your team buys in to the plan. They need to be willing to get behind the logical sequencing, the durations, any concessions that had to be made, any extraordinary requirements and any risks that are known. If your team is starting with a "well I never thought this plan would work, anyway" attitude, you are starting an uphill battle and your team is targeting failure.

Courage

As Program Manager, you are going to be asked to make decisions with partial data. If it's possible to get the data and keep the schedule, then you should wait to decide. However, where the data simply isn't available, you will have to assess the risk and decide on a direction for your team. Protect the team from analysis paralysis. As an example, if the team knows there is a prototype order in 16 weeks but nobody knows how many parts will be required, use whatever you know and make up a number and state it as though it is non-negotiable. If you know the customer is going to need 5 to 10 parts and you'd like a couple for your team to assess, then weigh the cost, consider the budget and decide. The team wants to be led.

Prioritization also takes logic and courage. When I run a program I think of it as "Tony Incorporated" and the buck stops with me. There will be tradeoffs and not everybody is likely to be happy during the whole program. It's good to talk these things over with team members or stakeholders, if appropriate or if they are available. However, they are likely dealing with the same missing data or dilemma that you are and the team needs direction. Ultimately, and usually quickly, you have to set priorities. You can make corrections as you go, but analysis paralysis will derail team morale quickly.

Problem Solving

It is essential for Program Managers to be problem solvers, by nature and by training. Kepner-Tregoe, 5-Why and Shainin "Red X" are all very useful methodologies. Team morale and confidence in their leader will fall rapidly in an environment that relies on trial-and-err problem solving.

Email

Program Managers that sit behind their computer screens and attempt to manage a team by email are destined to fail. First, it is too impersonal for your team to ever grant you leadership and credibility. Make no mistake, leadership cannot be taken – it must be given by those that you mean to lead. Second, it gets overwhelming to manage the thousands of emails that will be compiled through the execution of a Program.

Program Managers must lead their teams face-to-face. Having said that, email clearly has its place. For broad and general announcements, meeting minutes, following up to document an agreement, email is a superb tool. When it replaces organic interaction of carbon-based life forms (two people talking) is when the team starts to fall apart.

The Tools of the Trade

I believe in simplifying the project environment for the team and for my own sanity. I reduce the tools that I update regularly to the essential minimum to run a program.

1) Program Timeline - Already discussed at length, above.

2) Parts Matrix or Bill of Materials – The devil is in the details. Assemblies are rarely late. Components that make up the assemblies are late. You need to have the details about each component in front of the team. What parts are required? What are the made of? Is the material readily available? Is it new technology? Have potential suppliers been vetted and approved for use by your company? What is the lead-time for tooling? What is the lead-time for component build? What does the supply chain look like? What is the PO number? This tool grows in importance as the complex functionality of your end product or your bill of materials component count increases.

3) Open Issues List – This is different than a task list. This is a list of items that, if left unattended, will cause the program to miss a cost, timing or quality expectation of the team.

4) Task List – A simple “to-do list”.

5) Statement of Work (SOW) – On a project of any complexity, there are constant scope creep assaults. They can be so subtle that they amount to elegance creep, but can still affect cost, timing and quality. I find that I am required to refer to the SOW frequently.

6) Meeting Minutes – If it wasn’t written down or recorded, it wasn’t said.

Closing

The academics of Program Management are relatively simple. The soft skills are the secret sauce for good Program Managers. Realizing that the rest of the program team has probably worked with a Program Manager in the past that has given them reason to consider us a “necessary evil” is important. They knew all of the tools but didn’t understand people, didn’t think logically, were easily thrown off the scent of the critical path, didn’t have the courage to decide or the ability to prioritize, didn’t simplify and considered team culture to be an afterthought.

I’ve arrived at this approach based on course correction after a couple of examples of doing it the hard way and falling short of expectations. I have, repeatedly, used the combination of the basics and the “secrets” contained in this paper to lead program teams to success. In fact, the same approach has worked in department and company “turn-arounds”. There is, absolutely, no magic in this paper. However, it ties several ideas to Program Management that are not immediately obvious to successfully manage a cross-functional team.

References

1) Forbes on MSNBC.com, http://www.msnbc.msn.com/id/40535834/ns/business-forbes_com/t/most-memorable-product-launches/#.TpCe--tqLrQ