Project Management: Tips For Helping You Adopt A Process

By Randy McGowan

The Rational Unified Process, Enterprise Unified Process, Agile Development Methodologies, Unified Modeling Language. They come in many names, complexities and sizes, but following one will help ensure success on your next project.

This article is not a detailed overview of a formal process. Instead, it provides an overview of the most critical components common to each, as well as some tips on successfully deploying them. Although many process descriptions do an excellent job of breaking down the various components of the process they rarely cover areas like how this affects your team, how many processes to use, or offer practical advice on issues encountered in the real world when trying to deploy one.

It can be very helpful as a beginners' introduction to process and can help you more easily grasp some of the concepts you will be introduced to. For the more experienced process guru, it should have some helpful tips on smoothing over some of the rough edges we all deal with from time to time.

The information here is based on experiences and lessons learned in 15 years of developing and managing over 100 complex project releases.

Following these fundamentals will improve your chances of success in any process you adopt and provide a solid foundation for maturing it.

What's a Process and Why Do I Need One?

Regardless of what business we are in, software, website design or retail clothing, we all have a process we follow to complete a given project. Sometimes it works and sometimes it does not, often with costly results. When we talk about adopting a process, we are talking about a more formal process. A process is essentially an integrated set of roles, methods and techniques to in part and help achieve the following:

- Minimise risk.
- More accurately estimate your project schedule and budget.
- Detect problems early (upstream) instead of later (downstream) when they are much more expensive to fix, if they can be fixed at all.
- Better communication among team members regarding project scope, requirements and status.
- More accurately track the progress of the project and detect slippage early.
- Accomplish the project's goals as efficiently and cost-effectively as possible.

Formal processes are often created and refined over years of trial and error to attempt to create an ideal "recipe" for having an optimal chance at successfully completing any project. While they were developed for and commonly used in Software Development, Aerospace and Engineering, most of the core concepts are not specific to these or any industry, and anyone can benefit from using them.

**How much process is enough?** It is critical to the success of any process to understand how much you initially need to bite off. The risk of trying to do too much too soon with a process can be as risky as not doing anything at all, especially if you are a more agile company trying to make the transition to being more process oriented. Overloading your team with a new set of responsibilities and methods they are not accustomed to, or ready for can easily derail you. However, if you do not start changing you will continue to have the same problems. Here are some tips for finding the right balance.

- **Risk factor.** What is the project's risk factor? Obviously making software for an artificial heart is much riskier than deploying the third generation of a website and the process, initially, should match the risk. The former would need extensive, redundant and exhaustive QA checks and balances where the latter can be easily adjusted on the fly after deployment with no loss of life. Be realistic about what your risks are, how expensive they will be to address downstream, and use this as a basis for deciding how much is required. No one knows your environment, project and team better than you, so use some common sense in deciding what feels right.

- **How much can your team handle and what does it need most?** The impact on the team is often overlooked. Any process is only as good as what your team can manage and regardless of the ultimate benefits, initially it will cause additional effort in training and new tasks your team is not accustomed to managing. To be successful, you must achieve buy-in and commitment to the process from everyone. If you do not your team will simply go through the motions and roll their collective eyes in project meetings. To overcome this, find their pain points in how they work now and start with the areas of the process that directly address these.

- **Start small.** Start with a few areas that you feel are critical, again including pain points, so your team sees immediate benefits. It will be easier to add more process layers later when they see it as a benefit and not simply extra layers of bureaucracy. Gaining buy in is critical, and if you start small, your team will have a chance to get their collective heads around this as well as see the benefits, making more maturation downstream easier.
Team and Environment

One of the most commonly overlooked elements of employing or maturing a process is the team itself. Each team has a different dynamic and will respond very differently to various aspects of what you are trying to do. Too often, out of frustration with problems new process is forced on a team. This does not mean your team should dictate your process, but as mentioned above, your team's buy into what you are doing is essential for your success. I have never seen a process successfully steamrolled over a team. So tread carefully, get your team involved in discussions about what you are doing and why, it will pay dividends.

• **Roles and responsibilities.** Any process will have roles defined for individuals, and it is critical that each person clearly understands the role they will be playing and feel they are comfortable in that role. Spend some time here and ask people if they are comfortable in their role, ask questions and listen! Once your team is set, make sure they are empowered to do what they need to do and make sure everyone on the team is aware of who has a gun and a badge. If your developers refuse to tell your project manager the information they need, you will have a problem. If the project manager reacts by dropping soft milestones into your project plan, you have a problem you will not even know about until it is too late. So make sure roles are clearly defined for everyone and that everyone knows who has the power on the team.

• **Full disclosure.** Enough cannot be said about this. The purpose of any process is to address problems as early (cheaply) as possible, and this can only be done with visibility at every stage accurately to assess the status of the project. Developer egos, team infighting, and defensive posturing all create an environment where no process can be effective. It is critical that team members are willing to admit mistakes, call out problems and do so in a way that does not create a hostile environment. To do this, you must bring the parties together and openly discuss this issue. Address the fact that issues are brought up for the overall good of the project and organisation. Reward those who find fault in themselves and point out mistakes. Often the tension can be cleared by starting with admitting your mistakes first, others will follow, so lead by example and you will see that you can create an open environment where people feel free to view mistakes and even criticise constructively.

• **Visibility.** Similar to the above, visibility is all about people feeling comfortable disclosing information to the group. Developers will want to sit on the code until the last minute because they know it is not ready; designers hate people seeing unfinished work. So understand why your developer or designer may be twitching as their early work is paraded in front of a group and tread lightly at first with criticism until they become more comfortable with this. Phrases like; "This is great but how about..." are invaluable, use them! The fundamental goal of any good process is
catching issues as early in the process as possible. So you must discuss this with your team and make sure everyone understands that this can only be done with full visibility on all aspects of the project.

The Proper Tools

You cannot control what you cannot measure. So make sure you have the proper tools in place for both managing the process and being able to track and communicate about your project.

- **Managing the process.** There are many excellent tools available for managing requirements, QA, and development. As with the process itself, make a call on how much you need before you dive in and start buying. Shore up critical parts of the process. Requirements management often gets the most attention, but requirements can be easily managed in a Microsoft Word document while QA is often overlooked. A solid database that allows QA to track features from implementation to completion and any bugs that result will be invaluable for QA and development and the project as a whole.

- **Tracking the project.** It is essential that your project manager be armed with a tool that can be used to show the progress of the project and its various comments. Look for tools that allow the right level of detail (high-level for management) and more detailed for individual departments and the project manager themselves. For example, Microsoft Project is an excellent tool for managing very strict rules driven projects. However many projects are exception driven, making strict project management tools difficult to use in a fluid changing environment. A great alternative is scheduling calendar software like The Calendar Planner, which provides the ability to manage various levels of detail in an easy to use calendar format. Allowing project status to be easily communicated among the team.

Scope

Next to the team environment, this is probably the most critical aspect of any project. You must focus on clearly defining scope at the earliest stages of development. The biggest mistake is usually trying to do too much on too short a schedule or budget or defining the scope and then not adhering to it. This frustrates developers and ultimately throws the project into chaos.

- **Be realistic.** Everyone wants everything right now, especially Sales and Marketing. Ask tough questions early of these departments about what features your customers MUST have versus what they WANT to have. Sometimes you will find marketers have made promises to a single client and were trying to save face when in the grand scheme the feature is not as important to the company’s goals. Focus on what you truly must do, let them know they cannot have everything and force them to choose. Make
sure the company’s goals are represented at all times in requirements. This is where the Vision document below comes in.

- **Vision document.** These may have various names for different methodologies, but a vision document is essentially a high-level overview of your project. Think of it as a mission statement for the project itself. This is where the company can clearly define what the goals are for the project. Who the stakeholders are and what the high-level requirements are. This will be your primary document for setting the scope of the project. Keep it HIGH-LEVEL, details can come elsewhere. Make sure the requirements map to the goals and that as you move forward the work being done remains true to these goals and requirements. That can change but only when the stakeholders agree and sign off on the changes.

- **Do not increase the scope without adjusting your schedule and budget!** Seems simple enough, but probably the single most common mistake. People always try to add "small" things that involve "minimal effort." These add up, and the impact is often not addressed, which ultimately leads to a failure in schedule and budget. The Change Board is your main defence against this, see Change Board below.

## Requirements

Requirements in any project are tricky, and many excellent books are dedicated to this subject alone. It is true that of the projects that fail most issues can be traced back to requirements. Strange how this continues to be the case when requirements are the easiest and cheapest way to find and fix problems.

- **A problem will never be cheaper than it is right now.** When you review your requirements, really review them, do not just scan them. Think about and try to visualise what they are saying. You will often find problems are apparent at the surface. Take the time to do thoughtful reviews and continue to refine the requirements until everyone feels they are correct. Compare the expense of re-writing a sentence in a word processor to re-writing hundreds of lines of code, re-testing and re-deploying, and you'll see these are the last chances you have at a cheap fix.

- **Get your customers involved, early!** Make sure customers are involved in the earliest stages of requirements and keep them involved. Often, a customer requests something and then developers disappear to figure it out. Big mistake! Come back to your clients with explanations of how you envision the feature and use mockups whenever possible so they can visualise it. You will find that making even a simple drawing of something will not only allow the customer to grasp it more easily, but you will quickly spot problems you have not considered.

- **QA starts at requirements.** QA should be involved in the beginning not just the end of the project as is so often the case. Let them freely review vision documents and requirements. They will often view potential issues
that other departments may miss.

**Change Boards**

When done properly, Change Boards can almost single-handedly manage even the most challenging project. However, if you do not have the correct team environment in place as mentioned above, they will be ineffective at best and at worst will create more animosity.

- **What it is.** Very simply the Change Board is a meeting where each of the key departments and sometimes clients are represented and have a chance to discuss the project from every angle. The idea is to make sure everyone is aware of the status and can speak to the impact any change in requirements or schedule will have on his or her respective department.

- **Who is there?** Usually, the list consists of the following: Marketing, Sales, QA, Operations, Development, IT, Project Management, Clients. Essentially everyone who has a stake in or is affected by the project. Depending on the nature of the project, Marketing or Sales will often represent the client. The most important rule here is, if someone is identified as a stakeholder in the project, do not have a meeting without them. If they cannot be there, reschedule.

- **Where to start?** A great way to start is usually to have everyone provide a brief update on what he or she is doing regarding the project. This helps remove the dark corners, often points out areas of disconnect and helps ease the tension of these sometimes contentious meetings by giving everyone an easy topic to cover and maybe brag a bit to start.

- **Where to focus?** The key issue in early Change Board meetings is scope. What is in and what is out? This will be a push and pull between what Sales and Marketing want and what those responsible for delivering can handle. After the scope settles down it is about status. Are the requirements still correct? Have priorities adjusted? Are we on target? Most importantly each group is represented so if Marketing says: "I must have this," Development is there to speak to the impact of this on the schedule, in real time. Again it promotes visibility and keeps things from being changed without everyone being able to speak to the ramifications. It simultaneously controls and informs.

- **How often?** They are very useful so have them as often as you need. This will depend on the nature of the project, but every 1-2 weeks is best. Longer and you start to have too little communication and too many potential areas for slippage, any more frequent, and you eat into too much work time.

- **What not to do.** Do not allow the meeting to descend into arguments and finger pointing. The Change Board is a tool that serves all departments. It is essential it remains a place where people can talk openly about issues. It is a place for reality, not spin. This will be harder than it sounds at times but resist the temptation to avoid them. There is no meeting more
important to the success of a project than these.

**Post-Mortem**

The post-mortem is a meeting to get together after the project has completed. This is not a post-release party although depending on the success it may have that atmosphere. It is a chance for some straight talk on what went wrong and more importantly how to address it in the future. Everyone lines up for post-mortems when things went well, but you can learn more from your failures than your successes. So if you had many problems do not miss this opportunity to address them when they are still fresh in everyone’s mind! Also, teams need a sense of closure, and this helps them do that as well as vent so you can clear the air before your next project starts.

- **Leave your ego at the door.** Nowhere are straight talk and the ability to provide and accept constructive criticism more critical. This meeting cannot be about egos, or CYA, it has to be a frank discussion about the mistakes made by everyone (we all make them) or areas in the process that need to be improved. Again, to set the tone, try leading off the meeting by the most senior person in the room discussing mistakes they made or things they learned. It helps set the right tone and eases the tension.

- **Take notes, then action.** This is the time to learn, and too often people discuss the issues then go off and do nothing. This is a chance to take corrective action to save you time and money on the next project. So take copious notes and put them into action while the iron is hot.

Follow these steps in any process you adopt or any project you manage and you should find it really will improve your chances at success.

---

*Randy McGowan has over 15 years experience developing various software programs both in and out of the Film and Television industries and has managed over 100 software releases.*

© 2014 Project Smart. All rights reserved.