Common Software Project Management Mistakes - Personnel Issues

Common Software Project Management Mistakes - People Issues

Very few projects go as planned, however, projects that fail often follow a pattern. Normally projects fail due to issues with:

1. **Personnel** - These issues range from wrong skill sets to bad work environments.
2. **Project Management** - These issues relate to improper processes and techniques for managing projects.
3. **Application** - These are issues with the application itself.
4. **Technology** - These are issues with the technology chosen for the application.

In this newsletter, we will focus on **Personnel** issues, we will cover the other issues in subsequent newsletters. Below are the common project management mistakes relating to **Personnel**:

1. **Programmer Gold Plating** - Gold Plating is when a programmer interprets the specification in a certain way and decides to add bells and whistles that were not in the original specification. This is done because they believe the extra features will add value to the product but do not realize that adding those additional features could be counter to the overall development plan and can create a lot of additional work.

   Here is an example:
   Let's say that the specification calls for creating a logon that prompts the user for a userid and password, validates it for a match and logs the person in. The programmer decides that this not sufficient, so creates logic to check for short and long passwords, sends an email to the administrator if the person tries to logon 3 times unsuccessfully, and adds "forgot password" and "change password" features.

   While well intentioned, the programmer may not realize that these things were thought of originally, but was decided that it would be better to implement them in a future release, so that they could go to market more quickly. Also, since the programmer created 2 new screens (forgot password and change password screens), which now requires the documentation team to document the new screens and add those to the User Guides, impacting the documentation team's deliverables.

2. **Improper Skill Sets** - Another common mistake is to take superstar employees that have a certain skill set, and put them on another project that requires a different skill set. For example, a very strong Java programmer (who has had successes in the past), may not have the skills to work in a .NET environment, without additional re-training. Many managers make the mistake of thinking a programming skill set is the same, regardless of the programming language.

3. **Poor Work Environments** - Crowded and noisy work environments can wreak havoc on project deadlines. Some programming sites have multiple programmers in one room, which makes it difficult to allow them to fully focus on their tasks. Other times, management condones a work environment that encourages unnecessary meetings and allow employees to stand around the water cooler for hours, draining productivity.

4. **Problem Employees** - Some employees will be problematic. Problem employees constantly cause friction between team members, challenge authority and introduce issues that are not relevant to finishing the project.

5. **Late Comers** - When projects fall behind, the natural reaction is to simply add more bodies. Unfortunately, each additional resource requires training and can quickly drain productivity from employees that are already behind schedule.

Solutions

The list of personnel issues discussed are not a complete list, but are some of the major issues that can cause project failure. A way to mitigate the risk of having these issues is to use tools to help you manage the software lifecycle. For example, Software Planner (http://www.SoftwarePlanner.com) allows you to mitigate a lot of the risks above:

- **Programmer Gold Plating -** During the requirements phase, all requirements should be reviewed with all project participants ahead of time. The issue of Gold Plating should be presented and explained and all project participants should agree not do fall into this trap. Use the **Functional Specifications** area of Software Planner to document customer requirements. Have programmers create detailed designs so that you are clear that they are not gold plating.
  More Information on Tracking Customer Requirements…
- **Improper Skills Sets** - Have technical leads interview each of their staff members to ensure that they have the skills necessary for the job, even if they are existing employees with the company. Use the Shared Documents feature of Software Planner to post job requirements, expectations, and checklists for interviewing candidates.
  More Information on Shared Documents…
- **Poor Work Environments** - If possible, offer each employee their own office with a door, even if it is a 10 x 8 room. Studies show that this dramatically increases employee productivity. If that is not practical, have people in cubes spaced far enough to allow them room to concentrate. Let everyone know that loud noises are not tolerated in common spaces. Create and maintain policies for length of breaks and disruption of work. Finally, encourage meetings that have a real agenda and are needed to better manage your project. Discourage meetings that have no purpose other than to check status.
- **Problem Employees** - Identify these employees quickly in the process and fire them. The earlier you act on these issues, the better off your team will be.
- **Late Comers** - Identify slipping tasks early in the project management life cycle. Remedy this with existing employees. If additional resources are needed, train these resources with staff that are not working on critical path items. Software Planner has a mechanism for tracking tasks and allows each employee update their percentage complete online. It also sends an **Overdue Items Report** to you each day via email, showing items that are slipping. This allows you to learn of slipping tasks quickly so that a resolution can be found before it is too late to help the person in trouble.
  More Information on Project Tasks…

Helpful Templates

Below are some helpful templates to aid you in developing software solutions on-time and on-budget:

- **Project Management Guidelines -** http://www.PragmaticSW.com/Pragmatic/Templates/ProjectMgtGuidelines.rtf
- **Functional Specifications -** http://www.PragmaticSW.com/Pragmatic/Templates/FunctionalSpec.rtf
- **Architectural Overview -** http://www.PragmaticSW.com/Pragmatic/Templates/ArchitectureOverview.rtf
- **Detailed Design** - http://www.PragmaticSW.com/Pragmatic/Templates/DetailedDesign.rtf
- **Strategic Planning Document** - http://www.PragmaticSW.com/Pragmatic/Templates/StrategicPlanning.rtf
- **Test Design** - http://www.PragmaticSW.com/Pragmatic/Templates/TestDesign.rtf
- **Risk Assessment** - http://www.PragmaticSW.com/Pragmatic/Templates/Risk%20Assessment.rtf
- **Weekly Status** - http://www.PragmaticSW.com/Pragmatic/Templates/WeeklyStatusRpt.rtf
- **User Acceptance Test Release Report** - http://www.PragmaticSW.com/Pragmatic/Templates/UATRelease.rtf
- **Post Mortem Report** - http://www.PragmaticSW.com/Pragmatic/Templates/PostMortem.rtf
- **All Templates** - http://www.PragmaticSW.com/Templates.htm
- **Prior Newsletters** - http://www.PragmaticSW.com/Newsletters.htm
- **Software Planner** - http://www.SoftwarePlanner.com
- **Defect Tracker** - http://www.DefectTracker.com
- **Remoteus (Remote Desktop Sharing)** - http://www.PragmaticSW.com/Remoteus.asp

About the Author

Steve Miller is the President of **Pragmatic Software** (http://www.PragmaticSW.com). With over 20 years of experience, Steve has extensive knowledge in project management, software architecture and test design. Steve publishes a monthly newsletter for companies that design and develop software. You can read other newsletters at http://www.PragmaticSW.com/Newsletters.htm. Steve's email is steve.miller@PragmaticSW.com.

Pragmatic Software Co., Inc.
1745 Shea Center Drive
Suite 400
Highlands Ranch, CO 80129

Phone: 720.344.4846
Fax: 720.344.4847
Web site: http://www.PragmaticSW.com
E-mail: info@PragmaticSW.com