97 Things Every Software Project Manager Should Know

Roland Adrian P. Villanueva

De La Salle – College of Saint Benilde

Bachelor of Science in Information System

Table of Contents

Contribution 1: GET USERS INVOLVED AS EARLY AS POSSIBLE by Barbee Davis, PMP

Quote: “Today, the secret to project success is to involve the user almost as soon as there is anything visible to show them. How much better it is to find out that there are problems with what we are developing early on, rather than after the project is complete.”

Review

“Get Users Involved as Early as Possible” showcases the all too commonly overlooked factor of “end-user—programmer” interaction and communication before, during and after software programming. Projects of the past where built upon fulfilling certain “user-specifications” that were brought to life by the programmers own ideas; oftentimes leading into less than user-friendly outputs and discontent on the user’s part. The book gives a vivid example of the potential disaster that may be brought about should the user be forgotten as a major factor in software development.

Things I have learned:

- Project managers should always get the user involved in the creation process.
- The developer’s translation of the user’s specifications oftentimes do not match the user’s needs, thus much interaction between both parties is necessary to iron out such potential faults.
- The sooner you get the user anything solid to work with, the earlier they can provide feedback to you and increase your chances of hitting your mark accurately and deliver to them a well-made software that not only meets their needs but also functions for their maximum benefit.

Contribution 2: SUCCESS IS ALWAYS MEASURED IN BUSINESS VALUE by Barbee Davis, PMP

Quote: “We need to focus on the fact that the project is only as successful as the business value it adds to the organization.”

Review

“Success is Always Measured in Business Value” raises questions that make for better project management and goal-setting during the development phase. Success in considering key elements in play, pre and post production, greatly affects the marketability of any product and its overall total impact to its makers when it hits the market.

Things I have learned:

- Production of a considerably “great software” will not automatically spell “success” for the organization from which it came from.
- The project manager is a crucial tipping factor in the equation of quality over cost; cost being anything from time to money.
- Understanding gain is understanding loss; without this knowledge you are incapable of assessing the product’s potential in the market.

Contribution 3: START WITH THE END IN MIND by Luis E. Torres, PMP

Quote: “The right attitude and the right people management skills are paramount to your success as a project manager.”

Review

“Start with the End in Mind” is in itself the lesson. To come up with a solution before a problem arises, to think even just one step ahead are some of the key elements to learn from and put into practice. You’d want to deliver your best without any compromises to yourself, your co-workers, your company, and of course your client.

Things I have learned:

- Learn to differentiate a client’s “wants” from their “needs” to better come up with a realistic approach to handle and effectively deliver on both.
- Discuss with your team of co-workers the scope, quality, duration and cost of the project to set a common ground of understanding with regards to the project’s specifications, clearing up any inconsistencies along the way.
- As a project manager, you’d want not only to deliver on your duties but to also have the client come back to you; not in complaint, but for another project.

Contribution 4: THE FALLACY OF PERFECT KNOWLEDGE by David Wood

Quote: “Many demand that requirements be set in stone before a line of code is written. What nonsense!”

Review

“The Fallacy of Perfect Knowledge” exemplifies the need for continued studies regardless of your mastery of the subject matter. The requirements of projects oftentimes change during its development phase and sometimes even after it is completed. Until we acquire the means to perfectly predict software’s behavior, we must always assume that we are not fully knowledgeable about it as such we need to be in a near perpetual state of learning.

Things I have learned:

- No one, not even the developers can hold knowledge of everything about a project at any given time.
- Change is rapid, especially in the software industry; as such, so much more is demanded from its proprietors.

Contribution 5: THE FALLACY OF THE BIG ROUND BALL by David Wood

Quote: “Adaptability, flexibility of design, and readiness for change should be the cornerstones of any new software project.”

Review

“The Fallacy of the Big Round Ball” depicts software products as being “battered and bruised round balls” when in fact as they began they were intended to be “perfectly round balls”. This analogy simply implies that constant alterations to a product for its benefit will actually shape the product away from its intended form but in turn will most likely benefit its user the most. Failure to allow for changes to occur in and around the design would oftentimes lead to static and unusable products.

Things I have learned:

- Software system requirements can and oftentimes do change even after delivery.
- Maintenance crises are unavoidable for requirements can never be fully understood before coding.

Contribution 6: THE FALLACY OF PERFECT EXECUTION by David Wood

Quote: “If you think you can create flawless code if you work hard enough, don’t be embarrassed. Many others have thought so, too. Unfortunately, it is not possible. Even in theory.”

Review

“The Fallacy of Perfect Execution” recognizes the impossibility of procuring perfect software from bug-less programming. Software programming does not follow a predictable fashion of execution as it is based on each and every programmer’s analogy, logic and intent. Differences in opinion also play a part in the variety of introduced bugs in a program aside from logical errors.

Things I have learned:

- Bugs are bought about by both good and bad reasons and are unavoidable.
- Refactoring can iron out major kinks in programming but oftentimes introduces bugs of its own into the design.
- As software evolves, inconsistencies are bound to arise and are in need of new tools and techniques for refactoring.

Contribution 7: THE 60/60 RULE by David Wood

Quote: "We tend to pretend that software development is the most important part of the software lifecycle. Methodologies abound for development. Books, magazine articles, and blogs focus on development. Development, however, is just not where the money is."

Review

"The 60/60 Rule" is a law proposed for software maintenance. It states that 60% of lifecycle costs are due to maintenance and 60% of the maintenance itself is for used for enhancement. The reading underlines the factors involved in maintenance and the effect it has on overall project success.

Things I have learned:

- Software that are made must be flexible for facing new requirement in the future.

Contribution 8: CLEVER CODE IS HARD TO MAINTAIN… AND MAINTENANCE IS EVERYTHING by David Wood

Quote: “Code that is too clever will ultimately be too hard to maintain.”

Review

“Clever Code is Hard to Maintain… and Maintenance is Everything” runs through a developer’s chain of thought with regards to workarounds and general problem solving. The line “clever code” refers to fixes and patches employed by programmers when they attempt to make a new project work for antique or aged platforms.

Things I have learned:

- Let your developers learn and explore new languages and development tools, this is for your benefit as well.
- Cleverly done codes may solve the problem; oftentimes they do but when maintenance work in the future arrives, they may not be understood by other programmers and are treated as obsolete.

Contribution 9: THE WEB POINTS THE WAY (FOR NOW) by David Wood

Quote:

“We can learn important things from the Web's success. Perhaps the most important is that Moore's Law now allows us to accept a great deal of abstraction in our system design. Instead of being overly efficient with our hardware and software, we can now think about being overly stable, overly robust, overly scalable, and overly flexible.”

Review:

“The Web Points the Way (For Now)” is about viewing the Web as the epitome of all the hard work, experiences and achievements of all those who have built this technology that led to the development of all the others. Wood teaches us to view the Web as the ultimate example of stability, robustness, scalability and flexibility. He intends to show to us that from the very nature of the Web, we can derive elements, characteristics, and features that we can practically apply to the present and future problems and processes in software development. One particular characteristic of the Web that was seen as one of the most important is its being robust. In this article, Wood defined robust as being “capable of coping well with variations, sometimes unpredictable ones, with minimal damage, alteration or loss of functionality.”

Things I have learned:

- Software project managers must understand the characteristics of the Web, the ultimate invention tested and developed through time.
- The Web’s properties will continuously be the foundation of all new and innovative architectures in software development.
- Because ideas and technologies change, software project managers must put the Web behind their backs, but must be ready to build and beef up their own shoulders for those who will follow them.

Contribution 10: SERVE YOUR TEAM by Karen Gillison

Quote:

“Realize that one of the key roles of the project manager is to increase the team's velocity, and to work towards creating a team environment with few inhibitors to productivity.”

Review:

Here, Karen Gillison narrates her experience with a project manager in the agile approach. She vividly describes the procedures in the approach and the way her favorite project manager, their team leader, apparently, facilitates them to work smoothly and efficiently. In an agile approach, there are shorter planning phases, adaptability to changes, personal accountability, unit testing, and customer involvement. Here, participation of all parties at stake is highly encouraged. Gillison tries to tell us that in selflessly and flexibly serving the team, whatever position you are in, project management can flow smoothly and resiliently.

Things I have learned:

- A good project manager sees to it that there is an effective means of team communication and collaboration.
- Same as other socially-bound activities, software project management needs a competent and a harmonious team operation.
- One of the key roles of the project manager is to increase the team's velocity, and to work towards creating a team environment with few hindrances to success and productivity.

Contribution 11: AGGRESSIVELY PROMOTE COMMUNICATION CHANNELS WHILE MANAGING DISTRIBUTED PROJECTS by Anupam Kundu

Quote:

"Besides enhancing your communication strategy, there are logistics issues that need to be addressed to promote superior communication among distributed teams."

Review:

"Aggressively promote COMMUNICATION channels while managing distributed projects" is a quick list of problems, tips and steps on building a communication framework for the team working with distributed projects. Kundu starts by identifying the problems as perceived in the situation. With these problems, he enumerated and identified the steps in adding and building communication channels for a more workable setup for projects distributed in different collocations. This is a great list, but I think it lacks a deeper explanation or integration of ideas.

Things I have learned:

- Distributed projects should be tied together through communication because in this setup, there are a lot of hindrances to the project's success.
- Communication channels are an important parcel in project management, including distributed ones.
- Ensuring the success of a project includes strengthening the communication activities within the team.

Contribution 12: 9.7 REASONS I HATE YOUR WEBSITE by Barbee Davis, M.A., PHR, PMP

Quote:

"Most companies don't know that software and web development differ, so software project managers and software developers are asked to create websites. Here are 9.7 ways you can keep me from ever doing business with your company due to your annoying website."

Review:

Barbee Davis' contribution is an evaluation of annoying websites through the user's perspective. She lists 9.7 reasons or factors why a website can be perceived as annoying or badly developed. Primarily, she delves into the specific visual and operational elements in the website, evaluating their importance of use, or the lack of it. A website developer, hence, should consider these factors before planning to build a specific website.

Things I have learned:

- Software project managers and software developers are different from web development officers.
- Software developers should not be expected to be able to develop effective and user-friendly websites.
- Applications, color schemes and accessibility are important aspects in website building.

Contribution 13: IT'S THE PEOPLE, STUPID by Adrian Wible

Quote:

"Everyone on your team wants to contribute, learn, and achieve. It may be challenging at times to dig deeply enough to find this desire in some team members, but it's what makes software project management challenging and fun."

Review:

"It's the People, Stupid" highlights the importance of the project managers' competent leadership skills that should reach each and every member of the team. Wible emphasizes that each team member is a human being, with aspirations, constraints, strengths and weaknesses. He points out that a project manager should see to it that all of these predispositions of every team member must be voiced out. Again, like Contribution 11, Wible shows here that communication is crucial. For the team members to voice out their conditions, there should be a provocateur, which in this case is the project manager. Furthermore, Wible ended the article suggesting and explaining the CRAM Model (Constraints, Resources, Aptitude, Motivation).

Things I have learned:

- Project managers are depended upon their ability to lead.
- It is important to level down to the team members and dig up their predispositions to know why they have the performance problem.
- CRAM Model

Contribution 14: EVERY PROJECT MANAGER IS A CONTRACT ADMINISTRATOR by Fabio Teixeira de Melo

Quote:

"A workshop is a very effective tool to provide team members with knowledge about the most important contract aspects and the change control process. Hold this training the very beginning."

Review:

Fabio Teixeira de Melo focuses here on the change control part of project contracts. He points out that the team should be knowledgeable of the contracts extent and limitation so as not to agree with some changes that might lead more laborious responsibilities not covered by the contract. To be able to address this, de Melo says that every team member should know the contract from cover to cover for them to be ready to evaluate the client's change requests appropriately. In the end, de Melo suggests an effective change control process that would satisfy the clients' needs without sacrificing time and resources.

Things I have learned:

- Contracts should be well complied for because time and resources are always at stake.
- Workshops can be done before entering a certain project to inform the team members about the contracts' extent and the change control process they can use in case change requests are raised by the clients.
- There are third parties (providers, subcontractors, suppliers, middlemen) who might interfere in the change control process, but the project managers and clients alone should still be in control since they are the principal writers of the contract.
- The success of a project is measured primarily by client satisfaction.

Contribution 15: ENGAGE STAKEHOLDERS ALL THROUGH PROJECT LIFE by Lukeman Lawal, M.ENG, MNSE, R.ENGR.

Quote:

"Engage stockholders early and keep them involved through project completion. Be sure you know the business need for the project they support. Work towards aligning the needs of all of the key stakeholders, not just the top few."

Review:

This article of Lukeman Lawal essentially accentuates the significance of knowing the needs of all stakeholders in the project. The article emphasizes that the stakeholders must be involved all throughout the project. Lawal discusses many situations wherein this suggestion can be implemented. Again, likewise in Contribution 11 and 14, communication is seen as an important tool. Through communication, the project manager can elicit from the stakeholders pertinent ideas, complaints and clarifications. In turn, the project manager should also convey to them each and every detail of the steps in the project development.

Things I have learned:

- Project managers somewhat work as 'facilitators' in the project development.
- Engaging stakeholders might bring about many hurdles or setbacks, but they are all for the success of the project with proper guidance and facilitation of the project manager.
- Compromise and collaborate are the key verb terms in making the best out of engaging all stakeholders.

Contribution 16: CAN EARNED VALUE AND VELOCITY CO-EXIST ON REPORTS? by Barbee Davis, M.A., PHR, PMP

Quote:

“Software developers are increasingly certain that a more agile, flexible, approach to creating software is the best way to produce high-quality, working features that solve customer problems and provide business value. However, Project Management Offices (PMOs) are continuing to develop procedures and train project managers on more traditional approaches that work successfully in many non-Information Technology areas of the corporation.

Review:

“Can Earned Value and Velocity Co-exist on Reports?” raises the problem that project management offices (PMOs) use the traditional approach commonly used in non-Information Technology areas whereas software developers are increasingly convinced that a more agile and flexible approach in creating software is the best. Davis explained the process in the Earned Value approach and the Velocity and pointed out contrasts and comparisons for the reader to evaluate which one is better.

Things I have learned:

- Software project managers must also be aware of the approaches and processes done and observed by software developers.
- The value approach measures the amount of work done in relation to a specific time frame.
- The velocity approach measures how productivity of a developer in relation to how close the task is in the final output.

Contribution 17: DON'T SKIP VACATIONS FOR THE SAKE OF THE PROJECT by Joe Zenevitch

Quote:

"Certainly schedule your vacations to make sure you are available for project releases, but definitely take time off. And never cancel a vacation just because you think the project will grind to a halt without you."

Review:

Joe Zenevitch takes into account the role of the project manager as sole actor in the responsibility. Skipping vacations was an instance given to show time allotment or time sacrifice. Zenevitch proposes that the fact that he needs to skip vacations for the sake of the project means that the project is not actually being managed very well. Furthermore, he also pointed out to train someone for substitute.

Things I have learned:

- Project managers should plan their schedules to balance time for the self and time for the project.
- A well-trained substitute accustomed to the process and to the teaching of the process to others is acceptable.
- When a project manager encounters a case wherein his/her time allotted for other things is being overtaken by the project, there is no efficiency in the management.

Contribution 18: SIZE MATTERS by Anupam Kundu

Quote:

“The size of the project, the size of the team, the size of the deliverables, and the size of the checklists – everything in a project depends on its SIZE. Size changes the rules of how the game is played.”

Review:

“Size Matters” is literally about how quantity and measurements are important in project management. The main idea of the article is that everything depends on the size of the project. Hence, Kundu enumerated ways on how to adjust to the size of the project and chisel and cut them into pieces for better management distribution. This is a great technique that can be applicable in dealing with big projects. Kundu provided clear examples of situations like this and how they handled it through the “size matters” perspective.

Things I have learned:

- Breaking down a project into independent yet manageable work streams is acceptable in large projects.
- Even with independent work streams, one key contact point is responsible for the project’s cohesiveness and delivery.
- Documentation and communication is important to track down progress and performance of the work streams.

Contribution 19: ALIGN VISION AND OUTCOME EXPECTED by David Diaz Castillo, MBA, PMP

Quote:

"We should discipline ourselves to truly understand both the technical project requirements and the business value it is intended to provide. With this knowledge, we will be prepared to create better software results and manage uncertainty in a professional way throughout the project lifecycle."

Review:

"Align Vision and Outcome Expected" makes it clear that for a project to successfully end or complete, the beginning must be well-defined. In doing this, Castillo proposes that as the project manager, he/she must align team members with the vision and their outcomes expected through the triad perspectives: Business View, SMART View and Subjective View. The article explains further how each of these perspectives can be applied in the project development. It simply emphasizes that all the team members must understand why the project is the solution to the business problem, what the software should do about it, and how the clienteles think about the solution. It's a matter of having a concrete perception of the project from its conception to its marketing.

Things I have learned:

- Software project managers must make it clear to his/her team members the purpose and scope of the project.
- The SMART View: Specific, Measurable, Agreed Upon, Realistic, and possible to do with the Time constraint.
- In case there are troubles in change control or decision-making when it comes to the project, the team can and should always go back to the main purpose, scope and aim of the project.

Contribution 20: IMPORTANT NOT URGENT by Alex Miller

Quote:

“As a software project manager, you are in a unique position to focus the work of your entire team on the Important, but Not Urgent activities. Your job is to buffer your team from meaningless tasks and urgent but unimportant requests from other teams. You as the manager have the power to say no to these requests. Do them yourself, hire a lackey to do them, or just say no!”

Review:

In any project or any work-related undertaking, “Important Not Urgent” classifies and delineates among the kinds of specific jobs or activities the team (or a team member) does. The Important but Not Urgent activities are said to be the most significant in the project. Meanwhile, Important and Urgent activities should be dealt with instantly, but must be handled in such a way that they would be prevented. This article might seem to be a no-brainer contribution but it is an excellent management technique for improving the efficiency of the team as well as for saving resources and effort.

Things I have learned:

- Project managers must differentiate the kinds of activities his/her team members are doing.
- It is always best to plan and work for the future than to make solutions. (Prevention is better than cure.)
- To be highly effective in work means doing important but not urgent activities.

Contribution 21: A VOICE FROM THE OTHER SIDE by Marty Skomal

Quote:

"Please try to understand what your not-for-profit client can successfully implement, before exhausting your entire technology toolbag on an emerging market."

Review:

"A Voice from the Other Side" is an article by a client, advising based on a project failure due to over misinterpretation of the users' needs and plans. Skomal narrates what happened in the project, how the software developers penetrate the non-for-profit agencies like his. However, because of too much thinking ahead or exposure of the technical capabilities of the software developers, much has been done in the project that the clients no longer needed. In relation to Contribution 19, the team was not acquainted with the business problem. Further, in relation to Contribution 15, the stakeholders were not tapped or taken into consideration. Their needs were overlooked, and the project became an exaggeration. This article is a clear and tangible example of a project failure, and must be looked into very carefully by project managers.

Things I have learned:

- Project managers must not overlook the needs of their stakeholders and clients.
- There should be a progressive feedback mechanism for each activity done to prevent overlooking the user's perceived needs.
- A "Users' Needs Analysis" must be conducted prior to the project. It is equivalent to the Business View of the planning and conception process.

Contribution 22: CHART A COURSE FOR CHANGE by Kathy MacDougall

Quote:

"Key to understanding the impact of the change is to understand how people currently work and exactly how the new software will change that process. This increases the chances the users will adopt the new system, and also improves the design of your end-product, as it ensures that it will fit the needs of users."

Review:

In any new technology being marketed to its potential users, change is the most inevitable factor for its acceptance or rejection – how the users' activities will change, how their lifestyles can change. This article by MacDougall focuses on explaining the need to communicate the change that the software offers and create ways to make the users embrace this change. This is a great article because it does not only provide how to market a certain innovation through the concept of change, but also gives premium the users' needs and end goals.

Things I have learned:

- There is always a before-and-after situation in diffusing a certain innovation, in this case is software.
- Process flow diagrams can be used to monitor and evaluate the changes occurring in the users' activities and behavior.
- Planning is needed for a smooth transition into a new system.

Contribution 23: ROADMAPS: WHAT HAVE WE DONE FOR YOU LATELY? by Kathy MacDougall

Quote:

"This method of creating a project roadmap gives project stakeholders a voice and lets them know what to expect. And last, but by no means least, it affords your team a regular method by which to communicate to others what they have successfully delivered during previous quarters."

Review:

The importance of creating a project roadmap is discussed in this article. Further, the procedure and tips are also provided. The project roadmap is suggested to be a bridge between the team and the higher-level stakeholders. According to the article, it provides information on the changes planned, their risks and potentials. The explanations and narrations are pretty well written, but the article lacks more practical and vivid real-situation examples.

Things I have learned:

- A project roadmap is different from project plan.
- A project roadmap is a communication tool between the top project stakeholders and the team.
- There should always be a business value in every element of the project roadmap.

Contribution 24: DOCUMENTS AS A MEANS, NOT AN END by Patrick Kua

Quote:

"Successful project managers do just enough planning, capture just enough detail, realize that issues will invariably arise as the project progresses, and recognize when plans need to change because of new or unanticipated needs. They remember that the documents from the planning process are the means to a well-run project, not an end on in and of themselves."

Review:

Evaluating how a project manager fared in a specific project is believed to be oftentimes based on the end documents of the process. Kua, on the other hand, states that a successful project manager is the one has documented how the team dealt with unexpected changes, unanticipated needs, emerged issues, and unplanned diversions. In this article, Kua gives importance to the documentation of all the processes undergone in the project, no matter how big or small. He stresses out the need to record and keep track of the change control mechanisms, etc.

Things I have learned:

- A document is a tangible proof of project completion and team performance.
- It is important to keep track of the progress, changes and disabilities throughout the whole project.
- A successful project manager takes into account the process into which the project has been done, not only the output.

Contribution 25: YOU ARE NOT IN CONTROL by Patrick Kua

Quote:

"Great project managers exert just the right level of control, respecting what skills, experiences and connections team members bring to their project at hand. They recognize the signs when more control may help move the group towards its ultimate goal, as well as recognizing the signs when the same control may be slowing the group down."

Review:

"You Are Not In Control" is a narrative of Kua's experience with a project manager who acted like a totalitarian in their project meetings disrupting the harmony and participatory decision-making in the discussions. This was the start-off point of Kua's advice about avoiding being authoritative and bossy in such meetings. Being in control should not mean meddling and imposing decisions into the group, but rather in facilitating the exchange of ideas and voicing out of concerns and suggestions. According to earlier contributions, involving everyone is mandatory. It just essentially shows a democratic methodology.

Things I have learned:

- A project manager is a facilitator, not a dictator.
- Great project managers must know how to respect and accept the ideas of others as much as to encourage them in voicing out their thoughts and ideas.
- Project managers must have people skills and excellent communication skills.

Contribution 26: PAY YOUR DEBTS by Brian Sletten

Quote:

“The best way to pay off your technical debt is to assess what "loans" were taken at the end of each iteration. Ask your developers to identify specific hacks** they would like to unwind, and quantify how much time they think they need to do so. They do not need to pay debt off immediately, but it is good to gauge the extent of the needed repair while the shortcuts are still fresh in the developer's minds.”

Review:

“Pay Your Debts” is a metaphorical explanation of the notion of technical debts incurred by the software developer. Sletten explains that because of iteration, there is a need for the software developer to deliver the codes, even if they are still immature. Those codes which still need to be developed or changed or repaired, are the debts divided into single individual codes. In this article, the author also listed down some tips on how to avoid the piling up of debts. Some which are the software analysis tools, balancing workload, etc.

Things I have learned:

- When there is iteration, the software developers need to finish each of their own mark-up.
- When the software is equivalent of bankruptcy, it just means that the software has to be redone all over again.
- Just like in the actual concept of debts, the software developer must be able to pay technical debts even in minimal amounts.

Contribution 27: SOFTWARE FAILURE IS ORGANIZATIONAL FAILURE by Brian Sletten

Quote:

"The software industry has a lot of work to do to help its practitioners be more consistent in the delivery of high quality, on-time releases. Organizations that build software must be engaged in the process at all levels to improve their own chances for continued, repeatable success."

Review:

This article is a creative explanation to the number of incidences concerning software failure. Analysts have overlooked the fact that this software, after being completed by the software developers, project managers and contractors, are packaged, subjected to the assembly line, and delivered to the end users. Software failures, Sletten suggests, can be attributed to the defunct organizational structure of the company. They are the ones who are responsible for monitoring and evaluating "industry trends, acquiring tools, and adopting practices that demonstrate productive influences on how programmers work." In a sense, the success of the software developers, also lie within the hands of the organization, as part of group of stakeholders.

Things I have learned:

- It is not enough that the software developers or software project managers succeed, but also the organization's structure and system.
- The software project manager needs to measure and track success and failure records.
- Following the perspective that all stakeholders must be involved in the project from start to its end, software project managers should be responsible in binding and encouraging these stakeholders to exercise their rights and responsibilities in the participating in the project.

Contribution 28: PROJECT MANAGEMENT IS PROBLEM MANAGEMENT by Lorin Unger Hoboken

Quote:

"The truth is, our role exists because that is not the reality. Resources are over allocated, technologies and skill sets are incompatible, requirements are unclear, and timelines are unrealistic."

Review:

"Project Management is Problem Management" creatively puts the role of project managers as a need for the natural chaos in the workplace. Hoboken tries to tell us that we are problem solvers. The misfit resources, imbalance of work force, conflict of interests, etc. are all handled by the project manager. He says that of all the problems underlying in software projects, the most difficult to deal with are those dealing with people. Indeed, because of the relative differences of your team, you can never avoid to encounter conflicts and discrepancies in the work place. But this is where project managers are best paid for.

Things I have learned:

- Project management is an innately difficult job because it deals with problems, but management is all about dealing with problems anyway.
- The heart of the project management is actually about solving problems.
- There should always be a project manager in every organization.

Contribution 29: UNDER-PROMISE, OVER-DELIVER by Joe Zenevitch

Quote:

"At the end of the project, deliver less than you said you would and you are a bad software project manager. Deliver more than you said you would and you're the hero. Actually, you should strive to deliver exactly what you promised. No more, no less."

Review:

When it comes to the preparation and delivery of the product, this article, "Under – promis, Over-deliver", was written to let us understand up to what extent must we project managers show our capabilities in relation to satisfying the business owner or the client. In this article, Zenevitch teaches us to be wise and cunning in dealing with the business owners, in terms of releasing the product, delivering them, and exposing the features as planned or as a surprise. All in all, this is a great tip on how to control your clients, in such a way that they would always want to avail your team's services.

Things I have learned:

- High-medium-low categorizations must be avoided.
- Whatever the promised delivery amounts to, it should always be complied.
- Project managers must learn to deal with business owners.

Contribution 30: USE A WIKI by Adrian Wible

Quote:

“Wiki's are a great mechanism to centralize access to your project information. Hopefully, the wiki will be updated multiple times daily and will always be open in a window on a team members' desktop.”

Review:

“Use a Wiki” is a simple straight suggestion by Wible to put up a Wiki for all the information about the project. This is for all the team members to a have an easy reference on what the project is, how it’s going and all other things important in each of their own tasks. Wible identified some topics that should be included, namely: stakeholders, developers, general information, team calendars, meeting minutes, meeting agenda, business analyst and testers.

Things I have learned:

- It is okay to use Wikis.
- It is important to always update and inform your team about everything that’s happening about the project.
- Your team must have a communication specialist to do all those communication-related things.

Contribution 31: THE PMBOK® GUIDE IS NOT THE HOLY BIBLE by Fabio Texeira De Melo, PMP

Quote:

"All the project management knowledge you have at your disposition is a means, not an end. Besides, your team will naturally give importance to the things you, as a project manager, give importance. If your focus is the full establishment and compliance with all the project management processes, that will be the focus of your team, too. And then who will create the software?"

Review:

This is a great article of De Melo teaching us not to depend solely on text books and references, but in real-life situations wherein we involve our team, our stakeholders, their dispositions and backgrounds, their ideologies and beliefs, and all the things that can affect the way they work or contribute in the project. It is nice to read this article because it focuses on human interaction, that we learn best through communicating with our group. Through this, De Melo said, we will expertly know what approach to use, what processes to undertake for all of use to work harmoniously and give our best in doing the project.

Things I have learned:

- Since project management deals with people, approaches are subjective in nature. There are no strict rules, it depends on who you work with.
- Experience is the best teacher.
- You should conduct product analysis, needs analysis, contract analysis, risk analysis. In short, you must analyze!!!

Contribution 32: FAVOR THE NOW OVER THE SOON by Scott Davis

Quote:

"It is not that planning isn't a crucial part of successful software projects. Just do your planning based on modern software practices and expectations. Methodologies that date back to a time when code was written out in long-hand, meticulously transferred to punch cards, then hand-carried in a shoebox to a system administrator, don't translate well to an era where software is easy, free, and instantaneous. We are in the era of the now, and your processes should be adjusted accordingly."

Review:

This article teaches us to be a doer. Davis tells us that if we have something in our mind, let alone all the modern technologies do the planning and get the work started at once. However, I think that this is not acceptable in serious business projects. First, because you have to secure and save up on resources; second, you must analyze all the aspects of it; and third, it will be a waste of time since there are many projects out there which already have sure buyers. I think this suggestion is only for freelancing teams, with hardly a project signed upon a contract. Yes, inventions come from innovation, but when you innovate, you need resources and time.

Things I have learned:

- You can sometimes try to be an Einstein and surprise yourself to discover and invent new things.
- Optimism and determination are helpful traits in project management.
- You have to be prudent to create new things.

Contribution 33: FAVOR THE SIMPLE OVER THE COMPLEX by Scott Davis

Quote:

"Simplicity doesn't happen accidentally. It needs to be actively cultivated. Complexity is what happens when you aren't paying attention."

Review:

Much like other projects in other fields like graphic design, journalism, interior design, etc., simplicity is the key technique in developing convenient, accessible and practical outputs. Davis puts here very well the simpler and more ordinary situations wherein simplicity can is the most preferred. He says that in software projects, this technique also follows. Our task as project managers is to clean up the complexities of the software developers in such a way that our products would be appealing to the end users. This article very well explains how simplicity prevails over complexity in terms of software products.

Things I have learned:

- Simplicity is elegant and sleek.
- You do not have to overdo things. You just need to keep in mind what your users want and need.
- Software solves complex problems, but users need to use them in the simplest way possible.

Contribution 34: DEVELOPER PRODUCTIVITY MEAN VS. MEDIAN by Neal Ford

Quote:

"In software development, what is programmed today becomes the foundation for tomorrow. If you have mediocre developers building your foundation, the really good developers have to go back and fix the flaws before they can move on. Hiring mediocre or average developers slows project velocity*. Frequently, taking a poor performer off the team is more beneficial than adding a good one."

Review:

Ford in this article, "Developer Productivity Mean vs. Median" is about the difference between hiring skilled programmers and hiring average programmers. There are formulas used to determine which one is more efficient to hire. A project manager must know this in order to maximize productivity. In the end, the results show that it is best to hire skilled programmers since it pays off both in the long run and in the short run.

Things I have learned:

- There is always a compromise regarding the issue here, but in the end, what prevailed was to hire skilled workers.
- Mediocrity should be tolerated in serious business projects.
- Skilled programmers must be the ones hired in software development.

Contribution 35: SAVOR SUSTAINABLE PROGRESS, NOT SPEED by Venkat Subramaniam

Quote:

"When measuring time for a feature implementation, do not consider only the time it takes to write it in the first place. Add the time it takes to enhance, fix, and improve the code. Writing good quality code and tests takes time. It appears to be a short-term loss. However, it comes with a long-term gain."

Review:

Subramaniam compares and contrasts two programmers who both are beating deadlines. One is fast but turned out his product is not flexible for changes. The other is slow but his product was fast to change, alter and improve. This is how the concept of sustainability is pointed out in this article. It is deemed more logical to be slower in pace and ensure your products' stability and flexibility, than a software product not ready for user's feedbacks.

Things I have learned:

- It is always helpful to think ahead, anticipate problems and drawbacks.
- Long term projects and sustainability of products involve ensuring flexibility of codes so as to adjust with changes deemed necessary or preferred by the end user.
- "Slowly by surely!"

Contribution 36: VALUE RESULTS, NOT ONLY EFFORTS – WORK EFFICIENTLY, NOT LONGER by Venkat Subramaniam

Quote:

“Encourage programmers to report the progress they make, rather than how long they worked. Let them know that you care about getting results rather than keeping track of how long they spent at the computer. Once your team realizes you are a results-oriented manager and not a “put in hours” manager, their focus will shift to achieving results rather than merely clocking hours at work.”

Review:

“Value Results, Not Only Efforts – Work Efficiently, Not Longer” is an output-oriented way of micromanaging your human resources. It focuses on monitoring the progress of your team through their results and not only based on how long they work in that particular assignment. Just like Contribution 35, Subramaniam puts emphasis on the durability, quality, flexibility and competence of the product above all. I think it is logical because you work for the product, not just for the sake of working. You work for the user’s satisfaction, not for the satisfaction of the project manager or the contractor or the business owner. At the end of the day, what would really matter is the customer’s positive feedbacks.

Things I have learned:

- Focus on the product, not on the work hours.
- Efficiency is the amount of work done in a given duration of time. Efficiency is the priority.
- There should always be an output, not hours worked.

Contribution 37: DON'T THROW SPREADSHEETS ON PEOPLE ISSUES by Anupam Kundu

Quote:

"Most conflicts are a threat to productivity and efficiency; and resolving conflicts satisfactorily can actually strengthen relationships, foster creative change, and improve results. All conflict resolution tactics depend on proactive communication, active listening, compassionate understanding, and some effective negotiation and/or arbitration. Skilled software project managers are needed, because you can't solve people issues with spreadsheets."

Review:

"Don't Throw Spreadsheets on People Issues" is Anupam Kundu's article on conflict management and human resource supervision. In times of conflict and hard times, the software project manager is expected to be the negotiator, the middleman in the chaos. He/she is expected to stand between them or among them team members to work things out, work with their differences and build a proactive team.

Things I have learned:

- Software project management is about managing people, and that's what should be the expertise of project managers: conflict management.
- You don't give a silver plate to a stork or a bronze bottle to wolf.
- Project managers are referees.

Contribution 38: BUILD TEAMS TO RUN MARATHONS, NOT SPRINTS by Naresh Jain

Quote:

"If the project manager focuses on team building and individual growth, on time and within budget deliveries will automatically fall into place. This also ensures that teams are self-organized and don't need a baby sitter if the project manager needs to guide multiple projects simultaneously."

Review:

Software development is likened here to a marathon. And in that race, Jain says, the athlete must build teams and not sprint on his own. This article is well-written, but the topic is I think too cliché now. Although this is really an important trait (read: to work harmoniously with the team), there are I believe many more topics that the contributors can dwell into. This is just not the article that can make my head nod and make me say 'ohh.'

Things I have learned:

- Nothing much, really. Everyone's talking here about teamwork and cooperation, but this article does not offer really much for me.

Contribution 39: <u>GO AHEAD, THROW THAT PRACTICE OUT</u> by Naresh Jain

Quote:

"A knowledgeable project manager should sift through all the possible activities a team might be asked to do and retain only those that are vital to the success of this specific project. Once the leftover practices from projects past are swept away, the team's productivity and throughput should get better in a short period of time."

Review:

Jain tells us here that practice makes perfect, and that when it comes to software development and teamwork, the less processes undergone, the better the outputs. I like the part here in the article that Jain gave key indicators that things are broken in the software development processes. There are processes that somehow, he implies, give exaggeration to the supposedly simple process.

Things I have learned:

- Less is more when it comes to navigating the team toward software development. Less process, more ideas, better output.
- Practice does not necessarily mean piling up processes and tools.
- Perfection does not necessarily mean more processes and tools.

Contribution 40: YOU GET WHAT YOU ASK FOR by Naresh Jain

Quote:

"It is a well-known fact that if you measure the wrong things, you encourage wrong behavior. Software teams suffer daily because their managers are tracking and measuring them against the wrong parameters."

Review:

Software project managers should get straight in their minds what to expect from his/her team members. If the direction of the project manager is towards the wrong, then the team would go wrong as well. Project managers must have a clearer and more appropriate perception of what they should actually expect from the team. "You Get What You Ask For" simply stresses out that the team is not really hard to instruct or navigate towards the direction you want them to partake. However, that direction must be the right one. In a sense, project managers must therefore be careful in terms of asking favors, giving orders or playing out the rules.

Things I have learned:

- Measuring two to three parameters at a time (in terms of performance of the team members) is the preferred method.
- Project managers must have a well defined criteria for evaluating a team members performance.

Contribution 41: THE IMPORTANCE OF THE PROJECT SCOPE STATEMENT by Kim Heldman, PMP

Quote:

"Regularly reviewing the project scope statement can increase your chances for a successful project and keep your stakeholder's expectations aligned with the goals of the project."

Review:

Just like other contributions here, this article is just about the importance of ensuring the concreteness, solidity and accuracy of the definition of the project details in order for the team to execute the project as the client wants them to be. What I found unique here, however, is the reiteration of the need to review the project scope in the progression of the project, and aligning the team with all its goals.

Things I have learned:

- Project managers should always have with them the pertinent information regarding their project's scope, goal, status and requirements.
- The client's demand must meet the developers' and contractors' procedures and output.
- Communication is very important in every aspect of software development.

Contribution 42: HOW DO YOU DEFINE “FINISHED”? by Brian Sam-Bodden

Quote:

“The project manager must be the arbiter between the development team and the other stakeholders to define what it truly means to be “finished”.”

Review:

“How Do You Define Finished” is Bodden’s reiteration of the importance of well defined requirements, objectives, scope, and project status. The relay of information, from the one who handles the work packages, to the one who handles a deliverable up to the project manager, the contractor, the sponsor and the end users, there should be a similar consideration and set of criteria to be able to evaluate at the same level, at the same viewpoint. This is what Bodden is trying to say here in this article. He just clouds it with jargons that’s why it seems alien.

Things I have learned:

- There should be a well-defined, measureable and empirical set of criteria for the evaluation of the product, the developers and all other stakeholders in the software development process.
- Aligning all the stakeholders into one viewpoint, one playing field is close to impossible.
- The project managers are called to do their tasks here.

Contribution 43: INTRODUCE A MORE AGILE COMMUNICATION SYSTEM by Brian Sam-Bodden

Quote:

“By attacking the problem of keeping information loops tight and noise-free, software project managers can help avoid the breakdown in communications typically blamed for project failures. A project manager’s responsibility and challenge is to streamline the feedback loops at every level of a project.”

Review:

“Introduce A More Agile Communication System” is the ultimate key to all software project managers in effectively doing all their roles in the software development process. All along, what I thought lacks here is this, that there should be someone to advise to put up a competent, unbiased and operational communication system that can bind all the stakeholders together, though not at the same playing field (but at the same arena). Bodden did well here, noticing the strong need to establish a tool for the software project managers to hear out voices of his/her team, to have a working feedback mechanism, to extend his advice and suggestions, to retrieve project status reports, etc. Every software project manager who wishes to succeed in his niche must read this article. Now.

Things I have learned:

- A communication system is indeed a vital part of any team process.
- Transmission and feedback mechanisms can bridge information gaps.
- Project failures can best be avoided through an agile communication system.

Contribution 44: THE BEST PEOPLE TO CREATE THE ESTIMATES ARE THE ONES WHO DO THE WORK by Joe Zenevitch

Quote:

“On our projects, we do group estimation using a wideband-delphi approach. We start by having our business analyst describe the requirements for a feature, the development team listens, and then they ask clarifying questions. Once they are ready to give their initial estimate, they each write their individual figure on a card. When everyone has finished, on the count of three they all hold up their cards.”

Review:

Joe Zenevitch gives us a practical lesson here in this article. The title says it all. In doing estimates for the project, the best people involved must be the ones who will do the actual work. It is a no-brainer, but in this article, Zenevitch reminds us one of I think the most overlooked things in software development.

Things I have learned:

- I think it’s not really wrong to do the estimates WITH those who will do the work. It could be for the purpose of monitoring and evaluation.
- Software project managers must learn to trust their team members.

Contribution 45: ALICE DOES NOT LIVE HERE ANYMORE by Barbee Davis, PMP

Quote:

"Current discussions among software developers tend to revolve around the best programming language, systems architecture, operating platform, or project methodology. No one seems to notice that one of our team members, Alice, doesn't live here anymore. Where does Alice live now, and how will it effect our software development plans?"

Review:

"Alice Does Not Live Here Anymore" is an article about the characteristics of the virtual team members in the software development process. Barbee Davis tries to point out the significance of 'knowing your people', much like knowing your audience. Indeed, there are some individual characteristic, upbringings, dispositions, or life situations of the team members that may cause, in one way or another, a disruption of the project or the process.

In order to address this issue, Davis gives us example guide questions on how to know our team members and anticipate possible effects or influences on the project.

Things I have learned:

- There are many Alice's in software development teams.
- Alice's comprising the virtual team of software development may influence the process.

Contribution 46: A PROJECT IS THE PURSUIT OF A SOLUTION by Cindy Berg, PhD (ABD), PMP

Quote:

"Include the team, sponsors, and other stakeholders when creating a WBS. This ensures that the work of the project is fully defined and represents the needs of all of the participants. Why include the team? Well, who knows the work that needs to be done better than the project team members who will actually do those tasks. Projects are doomed to fail when the project manager assumes that he/she alone knows how to list every facet of the work of the project."

Review:

This article is Dr. Cindy Berg's explanation of the Work Breakdown Structure (WBS) and how it works in software development. According to Dr. Berg, the WBS is a hierarchical view of the project, its entire scope and how it's broken down into deliverables. Conceptualizing a solution, and how a specific plan of software development covers this solution, can be done through the WBS. The project roadmap is suggested to be a bridge between the team and the higher-level stakeholders. According to the article, it provides information on the changes planned, their risks and potentials. The explanations and narrations are pretty well written, but the article lacks more practical and vivid real-situation examples.

Things I have learned:

- The Work Breakdown Structure (WBS)
- WBS as the backbone of all the sub-processes in software development.

Contribution 47: TRUE SUCCESS COMES WITH A SUPPORTING ORGANIZATION by Cindy Berg, PhD, PMP

Quote:

“Project managers must provide objectivity toward the project. They occupy the unenviable role of owing their first allegiance to the organization that pays them, while at the same time needing to build a trust situation with the developers. If the project outlook looks bleak, an astute and principled project manager should make a recommendation that it be cancelled until peripheral problems can be addressed.”

Review:

“True Success Comes With A Supporting Organization” is about the role of the organization as the embodiment of the values and standards that project managers and his/her subordinate must adhere to. Dr. Berg emphasizes the role of the organization as the higher office, and realizes the responsibility of the project manager to adhere to this body together with his/her team members. When the organization and the project manager attains a mutual working relationship, that would be the true success that Dr. Berg talks about.

Things I have learned:

- Project managers are servant leaders.
- Project managers have the right to cancel a project depending on the policies and ideals of the organization.
- Communication bridges non-supportive or dysfunctional organizations and the team.

Contribution 48: A PROJECT DEPENDS ON TEAMWORK by Lelio Varella

Quote:

“To delegate you need to take into consideration the adequate combination of technical and managerial competencies required for each task. Once delegated, do not interfere. As the software project manager you are needed to monitor, give support, and ask about results. In doing so, you will provide motivation, earn respect, and foster team member “response-ability”.”

Review:

This is again a cliché – very much like the others about teamwork.

Things I have learned:

- None from this. It is too redundant already.

Contribution 49: ESTABLISH PROJECT MANAGEMENT GOVERNANCE by Ernani Marques de Silva, MBA, PMP, PgMP

Quote:

"Governance is a management method that is used to develop, communicate, implement, and monitor polices, procedures, practices, and other acts used to run a project. Putting an effective project governance structure and procedures into place helps ensure the project alignment, monitoring and controlling of threats and opportunities, decision-making, and delivery of project packages which are focused on the project planned. It allows you to appropriately address the risk and consequently meet the project requirements."

Review:

"Establish Project Management Governance" is a clear-cut suggestion for budding or up and rising software project managers. Ernani Marques makes it clear that establishing the governance of this profession, is an effective enforcement of the policies, procedures and structures of teams working on projects, whether on software development or other fields. Marques gives the reader practical and helpful tips on how to establish that office. The project board's functions are well defined.

Things I have learned:

- A role is not substantive without enforcement or written mandate or duty.
- An office, the symbol for governance, enforces the role the project managers.
- The Project Management Office (PMO)

Contribution 50: DO NOT FALL INTO THE NIGHT INVENTED HERE SYNDROME by Dr. Paul Giammalvo, CDT CCE, MScPM

Quote:

“Software project managers need to be willing to look outside their own IT world and learn what has been successful in other applications of project management, especially medicine and commercial airline piloting which enjoy significantly higher success rates than do IT projects.”

Review:

Project management can indeed be applied in other fields of specialization besides Information Technology and Computer Science. This article shows how project management works, the process associated with it and how project management in other fields can be viewed and related to software project management. Dr. Paul Giammalvo did very well in this contribution.

Things I have learned:

- Project management can be used in many ways outside Information Technology.
- Processes associated with project management.

Contribution 51: PMO – PROJECT MANAGEMENT OFFICE, EFFECTIVENESS IN PRACTICE by Angelo Valle

Quote:

“The PMO provides guidance in suitable standardized and validated tools, techniques, and software; reducing problems due to uncertainty and the growing emphasis on cheaper/better/faster projects. The PMO applies a standardized methodology where necessary and effective: project identification; data collection; analysis; information gathering; distribution; reporting; risk management; procurement; quality; and other project management knowledge areas, such as documentation and communications.”

Review:

Angelo Valle features project management as an important part of corporations the world over. In this article, he points out the role project managers share in businesses. This is inevitable since capitalism has always been part of society. Project management, as a whole, is is discussed in this article. However, in contrast with this article’s title, the content does not really direct me to the effectiveness of project management offices in corporations in real situation. There are no example, no drawings, no narrations.

Things I have learned:

- Project management will always be needed by organizations and corporations all over the world.
- Project Management Offices’ function within the business: strategic, directive, supportive.
- Software project managers can have a vast set of references for self-improvement in the game.

Contribution 52: MANAGING HUMAN FACTORS IN IT PROJECT MANAGEMENT by James Graham, PMP

Quote:

"People are human, so human emotions are natural in the workplace. But only people can develop software. So, nurture and manage your human capital as carefully as you monitor and protect your non-human resources."

Review:

"Managing Human Factors in IT Project Management" leads to one recurring point: that the team must be treated the way a father treats his family. The importance of creating a project roadmap is discussed in this article. Further, the procedure and tips are also provided. The project roadmap is suggested to be a bridge between the team and the higher-level stakeholders. The article provides information on the changes planned, their risks and potentials. The explanations and narrations are pretty well written, but the article lacks more practical and vivid real-situation examples. Over-all, James Graham points out that we need to fill in the gaps of each team members and make room for improvement and cooperation.

Things I have learned:

- Human capital must be nourished; hence emotional capital is the key to nourishment.
- We must always keep in mind that whether we work virtually or in a common office room, we should always keep track of our team member's predispositions.
- As the software project manager, it is our job to look out for symptoms of stress that can lead our team members to regress to old behaviors.

Contribution 53: THE VALUE OF PLANNING by Derry Simmel

Quote:

"Each time you plan you are thinking and communicating about your project. These necessary and fundamental activities will never fail to yield benefits well beyond their cost."

Review:

Derry Simmel tells us the value of planning as software project managers. The laying out of plans for our resources, procedures and work distributions might seem a bit boring for a well-experienced project manager, but in the end, these plans serve as the guiding light, star at Bethlehem, or the footprints in the sand, where the journey of software development would be achieved. In this article, Simmel gives specific tips on how to do the planning, what factors to consider, and basically the importance of it in project making.

Things I have learned:

- When you don't plan, you're gonna be blind.
- In any undertaking, records of plans, ideas and events must be kept.

Contribution 54: WHO IS MY CUSTOMER by Pavel Simsa, PMP

Quote:

"It makes them spend hours each week writing down what seems to them to be obvious, redundant information. For you as a software project manager, however, this is data used to get a bigger picture of your project progress, and then passed on to upper management. On average, a project manager helms five to seven projects at a time. Both you and your senior management team need you to collect and pass on this project data."

Review:

"Who Is My Customer?" is about software developers' resistance to sending their project status reports and how a project manager can do about this. This is very practical and helpful, since in real life, this might be true. These are the tips from Pravel Simsa I liked the most:

• Help them understand why this report is important to other team members or other departments who need to plan based on team progress. People work harder to help their peers.

• If the project progress was slow, know what the team was doing. Were they learning a new tool or language? Were there unexpected problems and challenges this week? When you compile the status reports, add the explanatory information to help others interpret the numbers.

• If you're managing more than one developer, create a group incentive. "If I get all status report by 3 pm every Friday from all of you for one month, everyone gets the next Friday afternoon off"….or, "I'll bring in food for a group lunch". Nobody wants to be the one who keeps their team from the reward.

Things I have learned:

- Developers hate to write status reports.
- It is the task of the project manager to encourage, or extract from them project status reports.
- How to make developers become less resistant to sending their status reports

Contribution 55: IMMORTALITY OF SCOPE CHANGES by Pavel Simsa, PMP

Quote:

“If you plan possible scope reductions from the beginning, it will make your decision about what to cut and how to cut it easier, should it become necessary.”

Review:

Just like other contributions here, this article is just about the importance of ensuring the concreteness, solidity and accuracy of the definition of the project details in order for the team to execute the project as the client wants them to be. What I found unique here, however, is the reiteration of the need to review the project scope in the progression of the project, and aligning the team with all its goals.

Things I have learned:

- The triple constraint: cost, time, scope.
- Scope changes from time to time; project managers must be ready and flexible for these changes.
- A lot are affected when scope changes.

Contribution 56: KNOW YOUR INTEGRATION POINTS by Monte Davis, MCSE

Quote:

“The heartache of every systems administrator, development engineer, and software project manager is systems integration. No matter how promising a newly created application; a freshly purchased software package; or a long-awaited, new-feature laden upgrade; the business value rests in getting it to work smoothly within the existing company system.”

Review:

It is true that integrating works of many diverse individuals, most of the time virtual, is a very difficult job. This article is a big help in addressing this issue. It is purely conversational, tutoring you on the know-hows of systems integration and tips on documentation, recording, and analysis.

Things I have learned:

- Linking together various software programs is a difficult but tolerable taks.
- When you look into system more intently than ever, you would find that there are actually points you can use for integration and organization.
- Documentation is an important process in many project management operations.

Contribution 57: A WORD THAT MAKES YOU MISS YOUR DEADLINE by Pavel Simsa, PMP

Quote:

"Which word can make you miss your deadline? The answer is "any word." When you are developing a product that will be released in other languages than English, you are adding numerous new risks and constraints to your project."

Review:

The importance of creating a project roadmap is discussed in this article. Further, the procedure and tips are also provided. The project roadmap is suggested to be a bridge between the team and the higher-level stakeholders. According to Simsa, in order to do the translation, these tasks are needed to be done:

• Add a "localization buffer" to the end of your schedule. End of schedule means the effective deadline for any work on the English product included in your project schedule. Any changes that need to be done after that targeted end date must meet very specific and very strict criteria to "get in" to the rework queue. Every change to this version also necessitates changes to the foreign ones.

• Sequence the tasks in a way that quality control of functionality is done separately from quality review of the English text. That can be as simple as copying all of the English text to a spreadsheet for proofing. That way, unclear wording can be found before the test cycle reveals it on an otherwise functioning product. Now, the necessary change can be done earlier and may not necessitate reworking other language versions.

Things I have learned:

- Even translating the language of the software is a complex job.
- Software project managers and software developers must be cautious in doing changes and attributes to the product.

Contribution 58: CLEAR TERMS, LONG FRIENDSHIP by Matteo Becchi, PMP

Quote:

“The title of this tip is from an old Italian saying, "Patti chiari, amicizia lunga", meaning "Clear Terms = Long Friendship". I think this mantra applies to many aspects of project management discipline. On a broader, methodological level, this saying summarizes in my mind the idea behind scope statements, setting goals and deliverables, and creating project definition documents. Really, all project artifacts are geared towards stating up-front the terms and goals the project team setting out to accomplish.”

Review:

I think that this article is plainly based on common sense. There are no new knowledge shared here. The use of the aphorism was unnecessary and I just think that this article is not developed really well.

Things I have learned:

- Clear Terms = Long Friendship

Contribution 59: EMPOWERING DEVELOPERS OR STORY OF A MAN NAMED TIM by Ken Sipe

Quote:

"Often the best thing a software project manager can do is set the vision, set the priorities, and get out of the way."

Review:

The true story about Tim is a great way to develop the topic. Ken Sipe drew the situation very well. Strategically, he put his ideas not excessively but in minimal yet effective amounts. "Empowering Developers or Story of a Man Named Tim" captures how a single man can outweigh the unity of a team and disrupt their work environment.

Things I have learned:

- A project manager must take into account these kinds of situations.
- A team must work together and should not be overpowered by one man.
- With a clear vision, well-defined acceptance criteria, and shared project priorities, one team should work what one man can.

Contribution 60: TO THINE OWN SELF BE TRUE by Harry Tucker

Quote:

“While life will always throw curveballs at all of us, having short and long term goals helps provide us with targets that help us realign our personal and professional course after the turbulence has passed. With such a plan in hand, we are enabled to focus more on the tasks at hand, including managing our teams, to empower them towards success.”

Review:

This contribution of Harry Tucker is the best motivational and inspiring article here in this book. Lists were put strategically and uncluttered; narration colorized the article; and quotable quotes were put in the end.

Things I have learned:

- In working, you need to be true to yourself and to the people around you.
- Reflections are important. Evaluate and reflect on how you are with your life; how you have been or what life offers you.
- In everything we do, we can always have something to offer to others.

Contribution 61: MAKE MONEY ON YOUR ISSUES by Randy Loomis, PMP

Quote:

“When negotiating a contract with a vendor, specify that both vendor and your project team's time be tracked against each separate issue that is encountered. This will allow the software project manager to have an accurate record, and be able to lower charges when there are issues with the vendor's original product, as opposed to problems with the contractual project work to implement it.”

Review:

The importance of creating a project roadmap is discussed in this article. Further, the procedure and tips are also provided. The project roadmap is suggested to be a bridge between the team and the higher-level stakeholders. According to the article, it provides information on the changes planned, their risks and potentials. The explanations and narrations are pretty well written, but the article lacks more practical and vivid real-situation examples.

Things I have learned:

- In computer programming, a program or sequence of instructions that is carried out by a program rather than by the computer processor is the script.
- Beware of bugs.

Contribution 62: EFFECTIVE MANAGE THE DELIVERABLES by Ermani Marques de Silva, MBA, PMP, PgMP

Quote:

"Projects are comprised of a set of deliverables which, when completed, generate the completion of the whole product, service or result. For software development projects, integrating all of components will be crucial for the final result to work properly. The components, of course, vary depending on the kind of software you are building. So, the deliverables are the major components which should actively be planned, controlled, monitored, and managed."

Review:

This article is specific to deliverables. When likened to a crop produce, deliverables are raw products ready for transfer to those who will package it, cut it into pieces or bundle them altogether, depending on the customer's needs. The importance of creating a project roadmap is discussed in this article. Further, the procedure and tips are also provided. The project roadmap is suggested to be a bridge between the team and the higher-level stakeholders. According to the article, it provides information on the changes planned, their risks and potentials. The explanations and narrations are pretty well written, but the article lacks more practical and vivid real-situation examples. However, all in all, Jared Richardson did well in this article "Effective Mange the Deliverables", although this title is erred in structure.

Things I have learned:

- Like any ordinary manufactured product, deliverables are processed into marketing.
- Deliverables must be protected in order to protect the completion of the whole project.
- It is ok to contribute knowledge even with wrong grammar.

Contribution 63: YOU AREN'T SPECIAL by Jared Richardson

Quote:

"Don't let your programmers reinvent the wheel. When they come to you explaining how special their problems are, point out that their Mothers may have stretched things when they made that "you're special" assessment. Be knowledgeable about what's available and guide your team towards high-quality open source or commercial tools."

Review:

Jared Richardson leads us to the point that the team must adhere to the known open sources and tools that can help them analyze and test their project statuses. The importance of creating a project roadmap is discussed in this article. Further, the procedure and tips are also provided. The project roadmap is suggested to be a bridge between the team and the higher-level stakeholders. The article provides information on the changes planned, their risks and potentials. The explanations and narrations are pretty well written, but the article lacks more practical and vivid real-situation examples. Jared Richardson, in this article "You Aren't Special" points out that we need to fill in the gaps of each team members and make room for improvement and cooperation.

Things I have learned:

- There are many available tools that can cater every software developer.
- As a project manager, our task is to make our team members use these kinds of tools for ensuring and testing the progresses attained.

Contribution 64: SHARE THE VISION by Jared Richardson

Quote:

"Remember, your team, and everyone at your company, wants to succeed. Share your vision and ask others to share theirs. You'll find most of those idiots you thought were out to close the company are actually people who will work side-by-side with you to solve mutually understood team challenges."

Review:

"Share the Vision" is about the need for involving the team in the vision of the project. Jared Richardson revealed that some team members might by in the dark, working only for the sake of work and do not actually understand the why's and what for's of the project.

The funny thing about this article is it started with a blast. "Do you work with idiots?" At first you wouldn't learn that it is about sharing the vision with the team. Anyway, Jared Richardson did it pretty well and I guess I learned from it.

Things I have learned:

- Sharing the vision means conveying to your team members the purpose of the project.
- A project's substance can be more valuable in terms of output if the team working on it knows it inside-out.
- Team challenges can be overcome by team spirit and a working knowledge of the project's vision.

Contribution 65: <u>THE MISSING LINK</u> by Paul Waggoner

Quote:

"Software project managers agree that one of their most difficult challenges is keeping team members properly engaged in the details of the project, and on top of their assigned tasks and schedules. They understand that team members are conflicted between the routine, operational responsibilities of processing daily work, troubleshooting problems, coordinating departmental issues, and answering everyday communications, and completing the time-sensitive work of project development."

Review:

This is again a reiteration of the importance engaging the team of software developers in the details of the project. Many contributions in this book have in one way or another tackled this topic. What is unique about this article is that it enumerated tips on how to avoid missing parts in a team (when no one's left to do a specific role in the team).

Things I have learned:

- Team members can be flexible and can work multiple tasks at the same time.

Contribution 66: ESTIMATE, ESTIMATE, ESTIMATE by Richard Sheridan

Quote:

"So often in project management, we get an estimate for a project at the beginning of the project (when we know the least) and then never revisit the estimate during the course of the project (when we know more than we did at the beginning). Worse, we never compare our original estimate with actual results to hone our future skills."

Review:

"Estimate, Estimate, Estimate" is Richard Sheridan's observation on software developers overlooking of a potential evaluation criterion. We estimate the project firsthand and never compare it with the actual numbers during or after the project. In a sense, it is a situation where in one forgets the importance of creating a project roadmap is discussed in this article. Further, the procedure and tips are also provided. The project roadmap is suggested to be a bridge between the team and the higher-level stakeholders. According to the article, it provides information on the changes planned, their risks and potentials. The explanations and narrations are pretty well written, but the article lacks more practical and vivid real-situation examples. The game at the end of the article was a great idea though.

Things I have learned:

- We get better at estimating the more we do it.
- Sometimes we learn we didn't know as much as we thought we did, and that helps our estimating.
- Estimating is a great "conversation" in our world, since we estimate as a group activity.

Contribution 67: ADD TALENTS, NOT SKILLS, TO YOUR TEAM by Richard Sheridan

Quote:

"Hiring for talents, not for skills, is a radically different way to build a team. However, I want to work with those who are poised to move enthusiastically beside me into exciting, new future technology."

Review:

Skill is different from talent. Talent is innate. You are born with talent and you can make it grow and bear fruits. Skill can be acquired, learned or studied. This article tells us that talents are more preferable than skills. Anyone can have the skills you have now. Anyone can learn it, master it. But talent is always there. That's a Richard Sheridan's suggestion – to choose talent over skill.

Things I have learned:

- Some people might appear to be excellent and expert. Now I know what to look for when hiring: talent.
- Skills can be taught, talents inherited.

Contribution 68: INCREASE COMMUNICATION BY HAVING FEWER MEETINGS by Richard Sheridan

Quote:

"Software project managers often fall into the deadly trap of regularly scheduling their teams for painful meetings that have the unfortunate, unintended effect of actually decreasing communications. One of the all-time dreaded meetings is the classic Monday morning status meeting. As if Mondays weren't bad enough already!"

Review:

Richard Sheridan gives us a tip about lessening meetings and intensifying communication within the company. He suggests an all-company meeting wherein all members of the company meet. In doing this, he pointed out ways on how to make this meeting the most productive of them all. Here they are:

1. Invite everyone involved in the project. We often have 50 to 60 people in this meeting.
2. Call the meeting with an alarm clock loud enough for everyone to hear. An impartial device calling the meeting is more likely to get participation. We use a dartboard that has an alarm clock in it.
3. Use a speaking token. We use a plastic Viking helmet to control the meeting. Just hand it around the circle of people STANDING (no sitting allowed). The person who has the token has the floor.
4. Have people report what they recently completed, what they are working on, and where they need help. Help doesn't come during the meeting, but afterwards.

Things I have learned:

- Software project managers can always adjust with the prevalent characteristics of his/her team members.

Contribution 69: TEACH THE PROCESS by Richard Sheridan

Quote:

"For a process to be truly effective there must be a common understanding of the process among all stakeholders. One of the ways we make sure this happens at my organization, is to teach formal classes in our processes to all stakeholders involved in a project."

Review:

Like other contributions here, "Teach the Process" by Richard Sheridan just repeats the lesson that the process must be communicated with all the stakeholders of the project. Sheridan provided stories of his experiences of teaching the processes, which ended up empowering the 'students'.

Things I have learned:

- Teaching the process empowers your team.
- There must be a communication tool between the top project stakeholders and the team.

Contribution 70: IT PROGRAM MANAGEMENT: SHARED VISION by David Diaz Castillo, PMP

Quote:

"To have a successful methodology, we need common, but flexible documents or templates that we are going to use for all projects. When we manage Information Technology (IT) programs, we typically need goods or services from vendors with their own unique methodology and templates, or outputs from other projects being run simultaneously with are own. So before we begin, all parties have to agree which documents we are going to use."

Review:

David Diaz Castillo talks here about Information Technology Program Management and how it can go together with software project management. In a sense, Castillo tries to imply that we have the same vision as IT Program Management and this shared vision can be used in our goals. Bundling projects into programs is the main proposal in this article.

Things I have learned:

- Information Technology Program Management
- Documents and templates must be saved and collected for references.
- Software project managers must maintain the credibility and independence of projects besides programs.

Contribution 71: <u>BUYING READY-MADE SOFTWARE</u> by Ernani Marques de Silva, MBA, PMP, PgMP

Quote:

"Currently, it is very common and useful to buy software that is ready-made and ready to be tested, implemented, and used out of the box. Why? Such software allows organizations to leverage their efficiency and optimize their effectiveness by cutting time spent in the developmental and implementation phases. In this kind of purchase, you are not only buying the software, but the know-how of the company that wrote the software."

Review:

Ernani Marques de Silva discusses the problem of buying ready-made software. In this article, he lists down what are the possible consequences of buying ready-made software, and gives this list of specific steps before signing the contract:

a) prepare a very detailed checklist regarding the company's software needs;
b) visit the company and prepare a Due Diligence Report;
c) prepare a Vendor Evaluation Report, Test Cases, and Test Plan;
d) make sure the Test Case is completed and documented;
e) follow the Test Plan/Cases before the contract is signed.

De Silva says that before buying ready-made software, we must first evaluate and study the real need for it and the specific requirement and limitations we must consider.

Things I have learned:

- Before buying ready-made software, one must study it hard and consider possibilities and risks.
- One must be vigilant and cautious in buying ready-made software.

Contribution 72: DOCUMENT YOUR PROCESS, THEN MAKE SURE IT IS FOLLOWED by Monte Davis, MCSE

Quote:

"Document your processes and make sure the process is followed. Although the name change process had been documented, it was not being followed. Otherwise, Sally's user account on the old mail server would have been updated with her new, married name migration e-mail address, and the issue would have been avoided."

Review:

Monte Davis shares a situation in which a series of changes in email wherein addresses are changed, causing the migration of emails from one address to another. The situation presented here in this article is a very nice way to describe the importance of tracking down changes in the processes in software development or anything else. Changes, when not documented and followed can cause the disturbance of some other processes and hence the failure of the larger body.

Things I have learned:

- It is important to document and follow changes in processes.
- Make sure you notify everyone concerned regarding the changes.
- Project managers must see to it that proper documentation of processes are done in order to monitor changes.

Contribution 73: DON'T ALWAYS BE THE MESSENGER by Matt Secoske

Quote:

"Providing clear, open channels for communication, along with the archival of discussions and decisions, allows all team members to interact directly with each other. This keeps The Messenger and The Scrambler project manager at bay, and keeps the software project moving forward."

Review:

"Don't Always Be The Messenger" focuses on the role of the software project manager to facilitate dialogue and open forums among the team as well as among other stakeholders. Being a bridge does not necessarily mean being a 'messenger' alone, but also being the referee, the host or arbiter. The importance of creating dialogues and open forums must be fully understood by the software project manager.

Things I have learned:

- A software manager is a facilitator of dialogue. Hence, he/she is like an emcee of a program.
- A software project manager must know then everything about conflict management.
- Without a facilitator, the team cannot naturally have a productive and constructive discussions with one another.

Contribution 74: PLANNING FOR REALITY by Craig Letavec, PMP, PgMP, MSP

Quote:

"It's amazing how often software projects to fall into late, over budget, off-quality situations. Even in highly touted software shops with international certifications and maturity assessments lining the walls, the trials of managing the very fluid environment of software development are many."

Review:

"Planning for Reality" tackles many techniques and approaches with time as a key factor. Craig Letavec discusses here the different paces software project managers could follow. He presents how each work, and the possibilities of each approach. However, there were no specific guides on how to do it in practice or the dos and don'ts. One the whole, Letavec presented here very helpful information for software project managers who are having a difficult time figuring out which approach to use.

Things I have learned:

- Buffer time is a good technique for flexibility.
- Being ahead of time can be both a disadvantage and an advantage when it comes to software development.
- The best approach is to view the project as a lifecycle.

Contribution 75: AVOIDING CONTRACT DISPUTES by Jorge Gelabert, PMP

Quote:

"The best way to avoid possible conflicts is to be fair when negotiating the contract. For example if the contract has penalties, make sure it includes bonuses as well. Both parties need to feel they have an equal chance to win and lose. Even with a fixed price contract, it may be better to renegotiate then to have the buyer (your customer) walk away from the project because they feel they have more to lose by completing the project than by walking away."

Review:

Jorge Gelabert points out the most important kind of work ethics every corporate man should know and practice: being honest and fair. "Avoiding Contact Disputes" focuses on the problem contract conflicts or disputes. Gelabert proposes one general solution. That is to always be honest and fair. In whatever work-related problems, this I think always works and Gelabert just had to point it out here to tell us that software project managers is balanced and fair when it comes to project contracts. There should always be a win-win situation.

Things I have learned:

- Contract disputes can easily be settled. There is no need for lawsuits.
- If you know your limitations, take them into consideration before entering commitments such as legal written contracts.
- Project managers must be fair to all his stakeholders – contactors, team members, clients, etc.

Contribution 76: PROJECT SPONSORS – THE GOOD, THE BAD, AND THE UGLY by Jorge Gelabert, PMP

Quote:

"Whether "Good," "Bad," or "Ugly," it is your responsibility as software project manager to manage the sponsor, just as you manage the project. Keep the sponsor well-informed, involve him/her only when necessary, and avoid allowing the sponsor to take control of the project. Learn to recognize the sponsor types and prepare accordingly."

Review:

Like what is pointed out in many contributions here, the software project manager must be able to cater all the needs of the stakeholders and setup a forum wherein these needs can be heard and addressed. "Project Sponsors – The Good, The Bad, and The Ugly" is about the three different classifications of project sponsors as Jorge Gelabert puts it. In the end, Gelabert recommends that the most ideal sponsor you should have is the Good sponsor.

Things I have learned:

- Three flavors of sponsors.
- Good sponsors are the most preferable.
- Software project managers must also evaluate and scrutinize the characteristics of sponsors in order to be able to deal with them accordingly.

Contribution 77: NOT SUPERHEROES by Angyne J. Schock-Smith

Quote:

“Make sure that you have teammates with complimentary skills work in partnership with you in areas where your weaknesses lie. But you don’t have to tell them that’s what you are doing, right? Keep some mystique about it and maybe you can convince the team that you are a Superhero. I won’t tell anyone otherwise.”

Review:

This article is about knowing and familiarizing yourself with your own strengths and weaknesses as well as your team members’. “Not Superheroes” means everyone has limitations and you cannot expect everything from anyone. This seems true not only in Information Technology environments but also in other fields. Angyne Schock-Smith tells us to know ourselves and our team members so that we would not expect too much and be disappointed. Nevertheless, encouragement with each other is ideal.

Things I have learned:

- Overcoming weaknesses and building up on our strengths is a logical thing to do for a team member.
- Project managers must be able to know his own strengths and weaknesses in order for him to work well with the stakeholders.
- Encourage your team members to deal with their weaknesses.

Contribution 78: HOW TO SPOT A GOOD IT DEVELOPER by James Graham, PMP

Quote:

"No matter how personable and skilled your applicant, always verify credentials with the issuing institutions and check out resume entries with the former employers. Careful hiring practices."

Review:

"How To Spot A Good IT Developer" is about James Graham's thoughts on hiring excellent developers. Creatively, he characterizes IT Developers that are preferable for software development. Software project managers must know these information in order for them to have the best progressive team they can ever have.

Things I have learned:

- A project manager must not only look for the skill and competence from an applicant but also his/her interpersonal relationships.
- Information Technology Developers must be team players.
- There should always be a business value in hiring developers.

Contribution 79: COMMUNICATING IS KEY by Gennady Mironoff

Quote:

“The most critical knowledge the Project Manager in any industry should have is how to be a good communicator. The person may have many different certifications and a list of titles and accreditations after his/her name, but without the basic knowledge of how to collaborate with others, the work of the project cannot be accomplished properly.”

Review:

This topic is also covered in many contributions here since a software project manager indeed needs excellent communication skills and systems to do all his tasks very well. “Communicating Is Key”, however, focuses on the importance of person to person communication. Even with the advent of information technologies, in every team, face to face communication still is the best method. I can attest to that because in face to face communication, you can take into consideration the non-verbal cues, there is a faster feedback mechanism, fast response time, and never costly (except for virtual team working in different places).

Things I have learned:

- Face to face communication is still preferable.
- The key to success is effective communication skills.
- Software project managers must be able to communicate well.

Contribution 80: RESTFUL ARCHITECTURE MAKES PROJECT MANAGEMENT EXTREMELY SIMPLE by Krishna Kadali

Quote:

"Leaving your mind open to new paths through software design provides a pleasant way to handle software projects in the world of constantly changing requirements. This flexibility will simplify your project management challenges, and creating fresh weapons and plans keeps your workday interesting and enriching."

Review:

Krishna Kadali nicely started the article "RESTful Architecture Makes Project Management Extremely Simple" with a nice quotation from Miyamoto Musashi. The importance of creating project architecture is discussed in this article. However, the procedure and tips are not provided. The project architecture is suggested to bridge the gaps on tools and ready-made softwares. The article, it provides information on the changes planned, their risks and potentials. The explanations and narrations are pretty well written, but the article lacks more practical and vivid real-situation examples. Kadali did this article well.

Things I have learned:

- You and your clients must know what resources are available for the project.
- With the necessary information, an architectural design can be planned and implemented.
- Software project managers must have an open mind when it comes to dealing with these kinds of processes and decision-making activities.

Contribution 81: KEEP YOUR PERSPECTIVE by James Graham, PMP

Quote:

"Perspective means looking for the best solution, not the fix that feels right to the users. Remember, users have a deep understanding of their business area and can make impressive contributions to a project by sharing that knowledge. But how should we use their input?"

Review:

"Keep Your Perspective" is about the need to be patient and focused when it comes to dealing with honest but negative feedbacks about the project or the project status perhaps. James Graham did well here in pointing out that feedbacks and responses are normal and constructive in nature. What we need to do according to the article, is to keep the perspective, become focused on the problem and try to find solutions. You cannot be easily affected because that is the same with all other jobs in the world. Everyone's not free from criticism.

Things I have learned:

- Always keep your cool when reading responses from customers or team leaders.
- It is hard to be affected by comments and criticisms at work because the quality of the performance would be compromised.
- The term perspective.

Contribution 82: <u>KEEP IT SIMPLE SIMON</u> by Krishna Kadali

Quote:

"Stakeholders of the project often make things more complicated than they need to be. This a common cause of software project failures. The team members of the project must have the ability to completely visualize the objectives of the project and complete actual work. Stakeholders, however, can accomplish the project in several simple, magical steps in their own minds. They imagine achieving the end solution quickly and easily, no matter how complex it is."

Review:

And how about the software project managers? Krishna Kadali presented here the problems of exaggeration and over-complication of stakeholders, most of them sponsors and clients. What software project managers can do about this is their capability to keep his/her team members on the operation. Kadali pointed out two ways to make things simple. One is to keep requirements simple and the other is to follow agile development processes. Project managers should know this.

Things I have learned:

- Simplicity is elegance.
- There should always be a halfway of mindsets between the team members and the clients.
- Project managers must keep in mind, again, to define simple requirements for the project.

Contribution 83: THE HOLY GRAIL OF PROJECT MANAGEMENT by Paul Waggoner

Quote:

“The challenge for the project manager (PM), especially when working with a new team, is to convey the essence of project management in a 30 minute overview, without overwhelming the team with methodology details.”

Review:

“The Holy Grail of Project Management is about the importance of setting out the pace with his/her team members through introducing the core concepts of project management without overwhelming them with methodologies, approaches and theories. Waggoner, to give justice to his suggestion, pointed out specific steps on how to do this.

Things I have learned:

- Project management, in its very essence, must be understood by the team.
- Everyone must know your role and responsibility as a software project manager.

Contribution 84: MEETINGS DON'T WRITE CODE by William J. Mills

Quote:

"Too often, people who could be doing something more productive are trapped in meetings. Meetings that have wandered off their intended purpose, run over time, or have trapped an entire team in the room when a more limited set of people would be just as effective. Only schedule meetings that have a specific purpose, and only include people on the invitation list who need to be there. Here's an obvious list of things to avoid, as software project manager, when you are planning your team meetings."

Review:

"Meetings Don't Write Code" is about the importance of documentation in meetings and assemblies. William Mills discusses here the different ways software project managers could follow. He presents how each work, and the possibilities of each approach. However, there were no specific guides on how to do it in practice or the dos and don'ts. One the whole, Millls presented here very helpful information for software project managers who are having a difficult time figuring out which approach to use.

Things I have learned:

- It is important to document discussions, forums, meetings and any event wherein there is an exchange of ideas.

Contribution 85: PROVIDE REGULAR TIME TO FOCUS by James Leigh

Quote:

“Planned meetings are especially problematic for programmers, as they might waste time when they know there is an upcoming item on the schedule. They think, “Why get started only to be interrupted in 30 minutes.” And great ideas that come during meetings may be lost or stale by the time the developer gets back to his computer to capture them.”

Review:

“Provide Regular Time to Focus” is a lecture of James Leigh’ experience in project management who acted like a totalitarian in their project meetings disrupting the harmony and participatory decision-making in the discussions. This was the start-off point of Leigh’s advice about avoiding being authoritative and bossy in such meetings. Being in control should not mean meddling and imposing decisions into the group, but rather in facilitating the exchange of ideas and voicing out of concerns and suggestions. According to earlier contributions, involving everyone is mandatory. It just essentially shows the need for the team’s breathers or vacant time.

Things I have learned:

- Team members must simultaneously focus on the project’s visions.
- Project managers must ensure that his/her team members do not have any other external distractions that can hinder them from focusing.

Contribution 86: WORK IN CYCLE by James Leigh

Quote:

“Our bodies are full of natural cycles and our productivity is no different. The human brain cannot focus on any single issue for more than a few hours at a time. Ideal work days are designed to ensure that the body has time to rest and refocus every 90 to 180 minutes. Productivity has been shown to degrade after about 90-120 minutes of work, requiring the brain to change focus before productivity can increase.”

Review:

James Leigh points out that every cycle should include: a planning stage, an action stage, a completion stage, and a reward stage. Before beginning on any action item, ask yourself or your team these questions. Why, when, how, what and who? Why are we doing this? When is this going to be complete? How are we going to do this? What are we going to accomplish? Who on our team will be responsible for each portion of the item? With proper communication and understanding, the action stage can be effective and productive, contributing to the overall success of the project.

Things I have learned:

- Cycles can be a good perspective in accomplishing the project.
- The team can work well in cycles and distributions.
- Project managers must plan or devise how to distribute the team into the cycles identified.

Contribution 87: RESPONDING TO A CRISIS by James Graham, PMP

Quote:

"Establishing clear responsibilities for dealing with crises is a good start. That is a task that can be done in advance, as can the preparation of checklists, processes, and procedures for each critical project phase. These can be incorporated in the project management plan and it's subsidiaries, and communicated so that all the team is clear."

Review:

The article can be best summed up by this guide provided by James Graham. • We have a risk register with appropriate responses identified.

• Our risk register is regularly updated and current.

• Our specialists on the team are trained to the appropriate level.

• We have a crisis management plan, with key responsibilities assigned.

• Our crisis management plan has a clear internal and external communications strategy/plan.

With a yes answer to all of this, Graham proposes that your project management is effective enough that it can endure crises through time.

Things I have learned:

- A project manager must be ready for conflicts and crises.
- Project management must not only devise ways to respond to a crisis but also to prevent it.
- Guide questions to know whether or not your job is done very well as a project manager.

Contribution 88: WHAT DO THEY WANT TO HEAR, ANYWAY? by Martha Legare

Quote:

"When in doubt, delete all but the essentials. You can prepare a handout for people to read later if they want more detailed information, and a take-away document will insure your facts won't get distorted. This approach will insure you will present your essentials succinctly. And when you find out that the President has cut your presentation from 30 minutes to 5 minutes in order to make his golf tee-time, you'll be prepared to summarize on the spot."

Review:

In "What Do They Want To Hear, Anyway?" Martha Legare tells us that we must engage both left brain logic and right brain creativity in order to effectively sell our ideas. Use the statistical proof, but showcase it in a memorable format. Use color, easy-to-understand charts and graphs, and just a few bullet points. Explain the story behind each bullet point rather than using too much text and reading it aloud a beat after participants have already read it for themselves. In essence, Legare tries to explain to us to save on time and workload.

Things I have learned:

- We must need to listen while speaking and to speak while listening.
- We project managers can learn from our team members as well.

Contribution 89: MAKE PROJECT SPONSORS WRITE THEIR OWN REQUIREMENTS by Miyoko Takeya, PMP

Quote:

"Without serious, specific consideration of what is to be created on this project during the requirement definition phase, the success of the project is severely jeopardized. Remember, they need to convey what they want this software to do, not how the programmers will go about producing that result."

Review:

Japanese contributor Miyoko Takeya shares that project failure also happens in Japan and other countries. One of the primary reasons is that requirements are not well defined. Takeya suggests that software project managers should spend time talking with those people funding the project in order to concretize the definition of the requirements, scope of the project and all other pertinent details that would be staunchly needed by the team. Takeya shares here some tips on how to ensure a concrete definition for the project.

Things I have learned:

- All project details must be defined very, very well!!
- It is important to coordinate with the sponsors of the project.
- Project managers should be confident and straight to the point in dealing with the clients.

Contribution 90: ONE DELIVERABLE, ONE PERSON by Alan Greenblatt

Quote:

“Every deliverable should have a single person who is responsible for its completion. Everyone working on the project should clearly understand who is responsible for the delivery of each item. Actual development of the item may involve a large group of people, but ultimate responsibility for ensuring it’s on time completion, and for understanding the technical issues surrounding that item, should be associated with one person.”

Review:

“One Deliverable, One Person” is a good concept of micromanagement. It distributes the workload into individuals’ own responsibilities. I think it is best for large projects. Greenblatt discusses here very well how this technique works. It provides us information on its ups and downs. We can learn a lot from this article, and we can use these information practically in our job as software project managers.

Things I have learned:

- It is logical to assign one specific responsibility to one person.
- However, there should be someone who binds or collates all the work done by individuals.
- You empower your team members when you give them responsibilities.

Contribution 91: REQUIREMENT SPECIFICATIONS – AN OXYMORON by Alan Greenblatt

Quote:

"Blurring the lines between requirements and specifications leads to the wrong people making the decisions. You either end up with the software developers making decisions about what features are important to a user, or with a software project manager telling a developer how to structure code. Either way, the result is a poor final software product."

Review:

"Requirement Specifications – An Oxymoron" is a great explanation of the difference between specifications and requirements. Greenblatt defined these terms excellently and provided examples. Hence, he classifies the role what the roles of the programmers are, the project managers' and the salesmen. This book must contain these kinds of information. However, most of the information here is spread out because this is organized per contribution. On the whole, the article was great.

Things I have learned:

- The difference between requirements and functionality.
- The difference between requirements and specifications.
- Programmers, software project managers and marketing executives must know these kinds of information.

Contribution 92: SPEED IS LIFE: MORE IS BETTER by Matt Daniel

Quote:

"Ask yourself, how do you as a software project manager balance speed to release with ensuring long-term relevance? What are the tools or practices you use to make sure that your new solution does not fall victim to obsolescence?"

Review:

Daniel is straightforward guy who talks about the issue of speed in software project management. He made examples of similar situations like in the field of aviation and electronics. From his presentation in the article, it can be implied that he opts for the optimum speed. He tells us that success is not about speed, but how you use the speed. Again, efficiency is the key here. How much work is done in a given time. I agree with him, because success in software development is measured by the satisfaction of the users. The less revisions, changes and improvements needed, the greater time saved, the best use of the speed.

Things I have learned:

- Optimum speed is preferable, not maximum speed.
- "Slowly but surely!"
- Analyze and study everything.

Contribution 93: <u>THE 2-DAY RULE</u> by Udi Dahan

Quote:

"I finally understood why I always needed to work with users to have them evaluate each feature as it was created, to be sure it added customer-perceived value. That way the project status reports, converted to earned-value reports, show the true percentage of earned value created rather than only showing how much work is left."

Review:

Udi Dahan narrated very vividly the problem he encountered with his team regarding conveying status reports. I think the problem here is that Dahan perhaps did not became hands-on in the project. He was just supervising, I think. Apparently, his team members were not 100% or even '95%' honest so as to give Dahan a more concrete, accurate, precise evaluation of the status of the project. It turned out that there still many problems and that they could not beat the deadline.

Things I have learned:

- Be hands-on in the project.
- Create a more tangible mechanism to gather and evaluate progress reports from your team.
- Do not trust too much, or be lax to overlook the problems.

Contribution 94: SCROLLING THROUGH TIME by Kim MacCormack

Quote:

"Fortunately, agile project management practices have alleviated some of these issues. By recognizing the importance of nose-to-nose interfaces between the developer and the real customer, we have evolved to collectively creating User Stories, and prioritizing features based on the business value they will provide to the customer, rather than requirements lists. A one or two week iteration process means we have early and frequent feedback, and the opportunity to clarify customer expectations."

Review:

This is a good sharing by Kim MacCormack about team-client relationship. Because of unclear requirements, incomplete information transmitted and a long line of product delivery, the team of MacCormack was not able to meet the end client's preferences. This is a very good lesson for contractors, subcontractors and project managers. In dealing with projects, contracts, requirements, scope and business value should be well defined to prevent loss of time, money, and effort.

Things I have learned:

- Do not engage in a long line of marketing (client-middlemen-subcontractors-my team) to eliminate information gaps.
- Deal with the end clients myself.
- Project managers and contractors should have excellent writing skills.

Contribution 95: HOW TO KILL MORALE by David Bock

Quote:

“High morale results in greater satisfaction among your employees, lower turnover, and higher productivity. On top of all that, its just nicer to be around happy people. Don’t you agree?”

Review:

I don’t know what is with the title, but this article is about the importance morale in a team. His narration of a true situation of a team that had drained of their morale and cheery environment is a good way to explain this topic. You would actually feel what happened and suddenly realize how you, as a project manager, can do something about this. As a whole, this article is great! It teaches project managers how to maintain a good working relationship among the team members, provide a happy atmosphere and boost their morale.

Things I have learned:

- Environment is indeed a very important factor in team work.
- It is the responsibility of the project manager to level the playing field and make at ease with his/her team members.
- It is always good working with happy people.

Contribution 96: CAN YOU MEASURE MORALE by David Bock

Quote:

"In your organization there will be unique opportunities to improve morale. Consciously look for them and take advantage of them. If they work, share them with others."

Review:

"Can You Measure Morale" is David Bock's inspirational tips on how to boost the team spirit. Morale, in the dictionary, means the level of confidence and spirits of a person or group. In every team, this is very important because it drives the team in a good working environment and a comforting atmosphere. Bock gave specific tips and guides on how to boost the morale of the team. My favorite tip among all is this:

"Make your team feel like a team. One team had a Player of the Week Award that changed hands at the team meeting each week. Russ might say, "I'm giving Mary the team player award because she worked late Thursday night. I was late getting her the documentation, but thanks to her efforts, we still finished the iteration Friday morning". The next week, Mary would recognize another team member's contribution and pass along the award."

That just says it all.

Things I have learned:

- You should make your team feel like it's a family. That every one works for the good of all.
- Encouragements and rewards are helpful.
- Software developers are humans, not robots. They just need a little tickling. ☺

Contribution 97: WE HAVE MET THE ENEMY..AND HE IS US by Barbee Davis, M.A., PHR, PMP

Quote:

"Open your mind to this new world of software development, and you can be a support for your software development team, not the enemy."

Review:

This article is especially for those new to software project management. The main tip is to not overdo things. Relax and feel at ease with the team and with the stakeholders. Just like any other group undertaking, here, project managers are the facilitators, not bosses or lecturers or punishers. Project managers are taught to work with the team. Davis points out that there are times when the hindrance to success is ourselves. Here we must learn to fight our biases, predispositions, frustrations and insecurities to work with the team well and be productive.

Things I have learned:

- We must learn to fight internal enemies.
- Project managers maneuver the direction of the team. He is responsible in leading the team to success.
- If the project manager is stupid and incompetent and immature, the project will always be a failure.