How a Traditional Project Manager Transforms to Scrum: PMBOK vs. Scrum

Jeff Sutherland, Ph.D., Nafis Ahmad, PMP

Abstract— **Transitioning from a traditional project manager to Scrum is challenging. The PMI Project Manager manages the project by ensuring that intermediate deliverables are delivered at various stages of the project. Agile development emphasizes the need for producing tangible results as soon as possible and as often as possible. The resulting role of an Agile project manager is fundamentally different from a PMI Project Manager. We have provided a mapping between the PMI responsibilities to Scrum and show a project manager how to more easily make the transition to an agile practice**

Keywords: PMI; PMBOK, Project Manager; Scrum; Agile

I. INTRODUCTION

THE Project Management Institute (PMI) has defined the Project Management Body of Knowledge (PMBOK) [1] which has codified the role of a project manager in a project. PMBOK defines the knowledge areas, phases and activities of a project that are conducted during the life of a project. The PMBOK approach towards project management is plan-oriented and activity centric, and thereby, results in a number of processes that has various inputs and outputs. The PMI Project Manager (Traditional Project Manager) manages the project by ensuring that these intermediate project deliverables are planned and delivered at various stages of the project. The Agile Manifesto [11] acknowledges the need for several of these deliverables but does not put as much emphasis on their creation and control. The Agile development process is geared towards people and interactions, and emphasizes the need for producing tangible results as soon as possible and as often as possible during the life of a project. The resulting role of an Agile Project Manager is thereby fundamentally different than that of a Traditional Project Manager. This paper attempts to highlight the major differences between the Traditional Project Manager and an Agile Project Manager. We will use the simplest and most common Agile methodology – Scrum, to present the differences. Additionally, we will provide a mapping of the PMI Knowledge Areas to Agile/Scrum practices that will also help to highlight the major differences in the two approaches. Finally, the paper will provide a mapping between a PMI Project Manager role and a Scrum Master, which would help the reader who is transitioning from a traditional project management role to a more agile environment.

II. TRADITIONAL PROJECT MANAGEMENT (TPM)

A traditional project typically has the following characteristics:

Phased: TPM is divided into distinct phases of homogeneous activities like requirements, design, implementation, verification and maintenance phases. A ‘technical transfer’ or handoff, transitions the project from one phase to another.
Sequential: The phases of the project are typically sequential where one phase starts when the previous phases are completed and ‘perfected’.
Non-iterative: The phases of project are typically not repeated unless through a formal process of change control which may trigger a sequential execution of phases again.
Plan-driven: TPM is driven by a detailed project plan which describes the various activities during the phases and ‘predicts’ the chronology of the activities, tasks, phases of the project and the effort that would be required to execute them.

The progress in a traditional project is seen to be flowing downwards like a waterfall through the various phases of the project, thereby such projects are said to be using the ‘waterfall approach’. Traditional approaches to project management are largely based upon the PMI defined practices and processes of project management. Even though PMI does not specify a methodology such as waterfall, most project managers who follow the PMI processes end up using the waterfall approach.

III. PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK)

For several years the gold standard of project management has been the PMBOK in which the Project Management Institute (PMI) has described the practices, processes, phases and knowledge areas that are commonly encountered in projects. The PMBOK defines 5 Process Groups (Initiating, Planning, Executing, Monitoring and Controlling, and Closing), and 9 knowledge areas (Integration, Scope, Time, Cost, Quality, Human Resource, Communication, Risk and Procurement Management).

IV. PMI PROJECT MANAGER

The PMBOK describes the Project Manager as major stakeholder within a project who is assigned to achieve project objectives. The PMI project manager is responsible for: [6]

- Developing the project management plan and all related component plans,
- Keeping the project on track in terms of schedule and budget
- Identifying, monitoring, and responding to risk, and
- Providing accurate and timely reporting of project metrics

The PMBOK defines the project manager as “the lead person responsible for communicating with all stakeholders, particularly the project sponsor, project team, and other key stakeholders. The project manager occupies the center of the interactions between stakeholders and the project itself.” [6]

V. MAPPING PMI CONCEPTS TO AGILE/SCRUM

The PMBOK does not explicitly prescribe a development methodology, however the Waterfall methodology is most commonly associated with it. An Agile and iterative methodology like Scrum can also be mapped to the PMBOK definition of a project when a few definitions are made clear. A phase defined in the PMBOK is equivalent to a Scrum Release and the sub-phases are then mapped to individual iterations or sprints. The adjustment that the traditional project need to make is to consider each sub-phase or sprint as a complete cycle of design, development and test, such that the deliverable is working code and not an intermediate artifact as common in Waterfall methodology. In this way the product is defined, developed and enhanced as part of a “progressive elaboration”. The Scrum project manager (Scrum Master) is responsible for developing the team and instilling the Scrum values and practices, whereas the team is responsible for all the deliverables including the planning of the project. The following sections will present the formal PMI concepts – Process Groups and Knowledge Areas, in terms of their Scrum equivalents, to provide the user a comparison and transition point.

VI. MAPPING PMBOK PROCESS GROUPS

Initiation: Scrum has Initiation processes that are responsible for the definition of the product roadmap, releases and sprints, and project authorization. Project Initiation phases vary from company to company, for example, Citrix Systems uses a formal enterprise Scrum model, which requires a detailed checklist that is used to initiate and authorize Scrum projects at Citrix. In contrast, a startup would have a lot less formalism for its initiation process. The first sprint may be used to provide a general structure and plan of the project, but would still deliver at least one feature of the product.

Planning: Scrum defines a **Release Planning Meeting** which establishes a plan and goal that the Scrum teams and the rest of the organization can understand and communicate. [2] The **Sprint Planning Meeting** is used to define and plan the goals and tasks of a phase or sub-phase of the project. [2] Additionally, a **Daily Scrum** (meeting) is conducted to plan the day for individual team members. The main difference is that the whole team is involved in the planning processes at all levels in Scrum, and is thereby committed to the delivery as the team owns the plan.

Executing: The Scrum team is responsible for carrying out the sprint and release tasks that result in a deliverable defined by the sprint/product backlog. The Scrum Master plays the role of a facilitator to ensure all the roadblocks are removed so that the team can ‘execute’. Project execution is done throughout the Scrum project, with the intent of providing a piece of working software as the end of each sprint. Scrum Masters do not manage or execute the project, instead they manage the (Scrum) principles, and in turn the Scrum principles manage the teams.

Monitoring and Control: Regular reviews conducted by the team as part of sprint/release retrospectives as well as the daily scrums that help in removing any obstacles that impede the team’s progress. Additionally, burndown charts are used to monitor the progress of the team.

Closing: The final sprint (**Sprint N+1**) can be used to perform administrative closure of the project, as well as perform any product hardening, fulfill audit/assessments requirements, or develop product transitioning documentation like user/installation/configuration manuals, etc. Sprint and Release retrospectives are conducted to ensure learning and feedback to improve the process and the priorities of the product backlog.

VII. MAPPING PMBOK KNOWLEDGE AREAS

The PMBOK defines nine knowledge areas which define the roles, responsibilities and deliverables for project stakeholders, including the project manager and the team. This section describes the mapping between each of the knowledge areas and the Agile/Scrum practice or process that corresponds to that knowledge area.

a) Integration Management

In both the Agile and Traditional paradigms, Integration Management brings the various processes together to create a concerted effort to deliver customer needs. The main difference is in the role of a project manager in each approach. The Scrum Master has a **lighter touch** than the traditional project manager – the former plays the role of a facilitator who helps the team perform optimally in a self-sufficient manner, whereas the latter depends upon plans that direct all phases and aspect of the project in as much detail as possible. The Scrum approach is more like a controlled chaos whereas TPM is more scripted in nature. This requires a natural mind shift on part of the project manager where they **need to give up control** in an Agile/Scrum project and empower the team to make and live their own decisions. The resulting project environment may appear to be chaotic, where the team decides the next steps on the project while applying the empirical data from the project to adjust and adapt. However, this is inline with the modern theories of optimal execution. Theoretical biologist, Stuart Kauffman has developed mathematical models that show that rate of evolution is maximized in evolving systems near the edge of chaos. [9]

Develop Project Charter: In both approaches, development of a project charter is essential to secure funding and buy-in from senior management. In both cases, feedback and approval may be necessary to start the project but the Scrum approach emphasizes the involvement of the project team even at this early stage. Most Agile projects call this step envisioning and the resulting deliverable is a vision statement or a product description box [3] developed by the Scrum team and not just the Project Manager/Scrum Master.

Develop Project Management Plan: A detailed, integrated project plan is a main deliverable of TPM and a traditional project manager spends a lot of time and effort in creating a detailed project plan up front. Scrum also utilizes planning on release and sprint levels, but the overall approach for planning Scrum projects is more of a **just-in-time (JIT) planning approach** where the next few sprints are planned in more detail within the context of an overall release plan. The more important questions for a Scrum Master are the length of the iterations, the size and scope of a release (how many sprints in a particular release) and setting up the framework of the project where it can iterate through various phases of the project. Another important difference is that the traditional project manager is solely responsible for the project plan, whereas in the Agile/ Scrum paradigm, the project teams are active collaborators and the collective owners of the plan. In other words, project planning is a team deliverable and not just the project manager's burden to carry.

Direct and Manage Project Execution: Scrum thrives on self-organizing teams where individual team members take more responsibility of their actions. The Scrum Master does not manage the project but "leads" by facilitating the team to achieve their objectives. Scrum fosters a process of '**continuous improvement'** using Sprint and Release Retrospectives, to provide learning from the previous sprint or release and adapt to the changes and lessons learned. The Scrum Master is responsible for facilitating the retrospectives. Integrated Change Control: There is no formal Change Control process in Scrum. However, change control is implicitly built into the Scrum process, with the team replacing the Change Control Board (CCB), and the product owner as the final decision making authority on the team. The change control process thus becomes management and up keep of the prioritized product backlog. Certain amount of change is expected in every project and practices and patterns have emerged that handle interrupts and changes in a Scrum project. For example, the Interrupt Pattern (*Illegitimus Non Interruptus*) defines three rules that advocate using buffers for unexpected items based on historical data. [10]

Project Closeout: The final sprint of the final release of an Agile/Scrum project serves as the final product handout to the customer, which is also accompanied by the administrative closure of the project. A project retrospective is conducted to document and discuss any lessons learned and suggestions for improvement of the products and processes.

b) Scope Management

Scope is a part of the iron triangle of traditional project management (Time, Cost, Scope & Quality – it really is a tetrahedron). TPM tries to define all the vertices of the triangle and makes efforts to keep them from changing. The Agile approach solves this seemingly intractable problem by keeping Time and Cost fixed (time-boxed iterations with a set Agile team), the only item that is negotiable is the scope. Scope control is the explicit responsibility of the product owner, who (re)prioritizes the product backlog to manage the scope, and produces features that are actually needed instead of following a plan-based approach which may result in features that are never used.

Scope Planning and Definition: A Scope Management Plan is not needed in Scrum. Agile projects use a Product Roadmap to guide the overall release and sprint plan for the whole project. The actual specification of the features that need to be built in a particular sprint is elaborated during or just before the sprint by the product owner. Scope is defined and controlled through the product backlog.

Create a WBS: Creation of a detailed Work Breakdown Structure (WBS) is not needed in Scrum - a typical project schedule is divided into releases and sprints and generally does not document the tasks that are performed in each sprint. Only the features or user stories that are to be implemented are identified. The feature set may be further elaborated into tasks for individual team members to be implemented in that particular sprint. Some teams also create an overall Feature Breakdown Structure (FBS) which severs as a roadmap for Agile releases [3]. Systematic, a Danish IT firm that has successfully combined CMMI and Agile, replaced 3 levels of WBS by a backlog, achieving 80% reduction in project management costs. [11]

Scope Verification: Acceptance testing is part of a sprint. Thereby verification of scope is a continuous activity that is performed throughout the project and controlled by the product backlog.

Scope Control: Scope control is performed by active product backlog management and time-boxing. The time-boxed structure of the sprint prevents customers from changing the features/scope and makes them wait until the next iteration to make any changes. Exceptions and interruptions are expected and can be handled using the rules defined in the Interrupt Pattern. [10]

c) Time Management

The Scrum approach towards time management is a more **top-down approach**, whereas TPM favors a more bottom-up approach. TPM is based upon Activities - their definition, estimation, and scheduling, captured in a Project Plan (Schedule), whereas, the Scrum approach is more value-based, but there is still a need for estimation of effort which is done on a high-level (releases) and a lower-level (sprints). A **Release Plan** encompasses multiple sprints using abstract measurements, like story points, to estimate the product (release) backlog. **Sprint Plans** are developed at sprint planning meetings, and are a list of activities and tasks that are part of the next sprint.

Activity Definition: In Scrum the definition of activities and tasks that are needed to develop the feature set for the iteration is done by the project team and is facilitated by the Scrum Master. Note that a TPM project plan would have tasks or activities with work packages attached to them as the lowest level in the WBS. In Scrum the work packages equate to features which may in turn have activities associated with them.

Activity Estimation and Sequencing: In Scrum "task estimation" and "task sequencing" is done by the team during sprint planning meetings.

Schedule development and control: In Scrum a schedule is developed at the Release level and then the sprint schedule is refined during the sprint planning. Since the sprint length remains time-boxed, the only planning needed is the inclusion, estimation and elaboration of individual features targeted for the sprint. The control of the schedule is achieved by disallowing any scope change while the sprint is in progress. The sprint retrospective provides the opportunity to optimize the overall schedule.

d) Cost Management

As in TPM, the Scrum Master also needs to work within the budgetary constraints and processes of their organizations. Scrum provides better budgetary control to the customer by providing better insight into the project through direct involvement with the team. The Scrum Master facilitates the team and helps the customer to collaborate with the team to make realistic cost decisions. [4]

Cost Estimation: The costs of a Scrum project are estimated top-down – a release plan is first developed and costs are estimated by using abstract measurements like Story Points. The rolling wave nature of a Scrum project planning allows elaboration of the estimates of tasks associated with each sprint. As part of the continuous improvement, the feedback from sprint planning then helps refine the overall release estimates. The other important aspect of cost estimation in Scrum is that costs are always estimated by the team with customer as an integral part of the team. In Scrum, estimation is done at different points in the project, like the beginning, after two or three sprints, or/and at the end/start of a release.

Cost Budgeting: The **cost baseline** of a project is created by adding up the costs of all the sprints. The estimates of the project or release are refined at the end of each sprint, as new features or functionality is uncovered and as the project velocity becomes apparent. An additional sprint can be planned as **management buffer** by savvy project managers to provide for any contingency.

Cost Control: In Scrum the Scrum Master is responsible for making sure that the customer is involved with the team and thereby knows the various scope changes as well as the team velocity. As more information about the estimated (both over and under) features, the scope changes are then handled through the prioritized project backlog. The Product Burndown Charts are used to show the progress of sprint and can be used as a cost controlling aide. In more regulated and formal environments, like the Government and Federal sectors, Earned Value Management (EVM) is a project requirement, and it can still be used by measuring progress at a higher granularity, as prescribed in AgileEVM [5]. AgileEVM (as traditional EVM) will still only measure the spend on the project at the time of measurement, but would not provide the actual business value delivered to the customer. [4]

e) Quality Management

Cross-functional nature of the Scrum teams makes Quality Assurance (QA) an integral part of Scrum, where QA personnel are involved in planning, design and implementation of a product. In turn the demands of Scrum are making QA rely more on QA automation tools to keep pace with the rapid pace of development and delivery of Scrum projects.

Quality Planning: Quality Planning is part of the overall planning and is the responsibility of the whole cross-functional Scrum team.

Quality Assurance: In Scrum QA is performed by the team itself, however in more formal environments, a third party can be brought in for assessments and audits; generally an extra sprint (Sprint N+1) is used for such assessment to fulfill the regulatory or industry compliance obligations. Additionally, retrospectives and reviews throughout the project, provides continuous QA.

Quality Control: Quality control is performed as part of the iteration by a combination of automated testing tools, QA team members and acceptance testing by product owners and business representatives. Burndown charts are also used as quality tools to monitor the trends of feature development. As part of quality control, some companies are beginning to add acceptance tests in the product backlog.

f) Human Resource Management

The Agile framework establishes cross-functional teams with mutual accountability, that are self-directing and self-correcting using regular team retrospection.

Human Resource Planning: Scrum creates cross-functional teams that are dedicated to the project for the duration of the project.

Acquiring a Project Team: Scrum recommends development of a cross-functional project team that is seven plus or minus two people (strive for an odd number of team members) for the entire duration of the project. [2]

Develop the Project Team: Use the Agile and Scrum values (commitment, openness, focus, courage and respect) to develop and build the team.

Manage the Project Team: An Agile project manager facilitates and coaches their team, while providing real-time feedback to the team.

g) Communication Management

Instead of formally documenting a Communications Management Plan, rely on the informality of Scrum teams that creates and emphasizes direct and face to face communication. Daily Scrum meetings are used for direct and frequent exchange of information. In Scrum, Stakeholders are engaged via the cross-functional team and thereby 'managing' them is not necessary. Information distribution and progress reporting is also simplified because of this structure. Online communication tools, like instant messaging or Skype videos are used frequently to work with team members that are not collocated.

h) Risk Management

Scrum is a Risk Reduction System [8] that handles risk on both strategic and tactical level. On a strategic level, it handles product risks like time to market, rapid response to change, etc. On a tactical level it handles product development risks by using Impediments/Issues lists. Risk Management Planning is done informally instead of a documented plan, as risk management is intrinsically built into the Scrum process. The whole team is involved in identification and analysis of risk at various points in the sprints and the project, using the daily scrums, and Sprint and Project Retrospectives.

i) Procurement Management

Scrum teams play a more active role in evaluating and selecting the sellers, often engaging in a proof of concept, possibly as part of one of the early iterations. Several contract types are used with Scrum projects, with clauses that allow flexibility for the customers. Scrum allows for contracts with 'early termination' clause, commonly referred as '**Money for Nothing**', where a customer can terminate a contract at the end of any sprint by paying 20% of remaining contract value. Another clause, '**Change for Free**', states that a customer shall make changes to Scope without incurring any additional cost if total scope of contracted work is not changed. [12]

VIII. THE TRANSITION – ORGANIZATIONAL GAME PLAN

The transition from the Traditional to Agile Project Management approach can be achieved in at least three major ways:

A. Apply Agile Techniques

In this approach various techniques that are commonly used in Agile projects are applied in a traditional project context to gain some efficiencies. For example, develop cross-functional teams and insist on having a product owner from the business to be part of the team or at least be available to the team during the life of the project or phase. Another Agile technique would be to use the daily standup meetings (scrums) that would keep the focus on immediate deliverables and removal of impediments quickly. Various other development and coding techniques, usually defined in the XP methodology can also improve the efficiency of a Traditional Project. This type of an approach is useful for environments where there is resistance to (radical) change, such as the government, department of defense, etc. A project group at Google, which has a non-prescriptive project management culture, successfully introduced Scrum practices, piecemeal, over a period of six months; sprints were introduced in the fourth month.

B. Use An Incremental Delivery Approach

A traditional waterfall project can be divided into several mini-waterfalls, where the design, coding and testing phases are sequential within the iteration but the product is delivered in increments, and thereby there would be opportunity between the iterations to modify the features sets by changing the priority of features or modifying the feature itself before the start of next waterfall.

C. Switch to a Pure Agile Approach

This approach is the most beneficial as it provides the full benefits of an Agile approach and the efficiencies it offers. However, this is a radical mind shift and it needs a cultural and organizational change that may or may not be feasible for some organizations. Hybrid approaches, such as the ones described above, may lead into a total organizational shift towards later on as the organization sees the benefits of the Agile approach.

IX. THE TRANSITION – INDIVIDUAL GAME PLAN

A. Understand the Differences

1. Value vs. Plan driven Approach
   The main difference between an Agile/Scrum and Traditional approach is that the former is a value-driven paradigm, and the later is more driven by project plans, or a desire to plan out the entire project up front. This does not mean that Scrum projects do not do any planning. In fact, Scrum approaches define plans at both the release and iteration level but the emphasis is on providing the highest value features in each iteration or release as the project team iterates through the project. The Traditional Projects Management approach is based upon detailed planning which is typically done up front and which defines the detailed-level tasks that are to be performed throughout the life of the project.

2. Empirical vs. Prescriptive Approach:
   Scrum implements an empirical approach based in process control theory [8]. Using first-hand observations and backlog graphs, the Scrum Master and the team adapts to the results of previous or current sprints by adjusting delivery targets to the team velocity or adjusting the team to improve the velocity itself. The TPM approach is prescriptive, in that it tries to predict the future, and develops a plan upfront based upon assumptions and no empirical data.

3. Self-Organizing vs. Directed Teams
   The Scrum Master takes a step back and lets the teams organize themselves. They provide the environment and opportunity for the teams to thrive and deliver. Scrum teams are highly self-disciplined; rather than imposing discipline,

Scrum Master provides the requisite leadership and coaching to the team to balance the agility and structure.

4. Stakeholder Management vs. Stakeholder Involvement
   Business Stakeholders are not managed by project managers; instead Scrum dictates that the product owner should be a part of the Scrum team. Such an arrangement ensures that the business is part of all the decisions and obviates the use of elaborate progress and status reporting.

5. Project Manager vs. Scrum Master
   Scrum Projects replaces the role of a project manager with a Scrum Master and splits the former's responsibilities into the Scrum Master, the Product Owner and the Scrum team in general. The Scrum Master is responsible for the velocity of the Scrum sprints, whereas the Product Owner is responsible for the vision and revenue of the product. The Scrum Master manages Scrum principles, while the principles mange the Scrum teams.

6. Top-down vs. Bottom-Up Planning
   Scrum favors a top-down approach where an overall roadmap is developed, which is progressively elaborated into releases and sprints. The traditional approach emphasizes on detailing individual tasks into Work Breakdown Structure (WBS) to construct the complete project plan at the beginning of the project.

B. . Understand the Similarities

1. Project Phases and Sub-Phases
   The phases and sub-phases described in the PMBOK are similar to the releases and sprints used for the Scrum projects as long as (a) a sub-phase is defined as a complete cycle of delivery (b) a sub-phase is always of a fixed length (4 weeks or less in case of the Scrum sprint) (c) requirements, design, development, etc are considered activities and not phases.
2. Software Engineering Practices
   Software Engineering practices such as the ones described by XP can be used in either paradigm to improve efficiency of the teams. Three main XP practices – continuous integration, automated testing and pair programming/code reviews, are cited as requisite practices for Scrum and are also used by several CMM 5 organizations.

C. Learn New Skills

1. Servant-Leadership
   Learn the art of Servant-leadership to achieve results for your organization by giving priority attention to the needs of the colleagues and the teams that you are serving. Main function of the Scrum leader is to remove impediments for the team so that the team can be empowered to make decisions, execute freely and deliver value.
2. Foster Collaboration
   Improve the interaction of individuals by using concepts of self-organization, self-discipline, respect for individuals, conflict resolution and technical competency.
3. Balance Flexibility and Stability
   Too much structure stifles creativity; too little structure breeds inefficiency.[3] The traditional project manager needs to de-emphasize the focus on control and provide more flexibility for the team to function, while making sure that the project still follows the basic Scrum principles and processes. Paradoxically, a well-executed Scrum requires more (self) discipline than traditional project management, in order to achieve and maintain the right balance of flexibility and stability.
4. Scrum Values
   Embody the Scrum values of commitment, openness, focus, courage and respect to develop and build the team.
5. Embrace Changes, Risks and Uncertainty
   Scrum works on the agile concept that "opportunity, uncertainty and risk reside in the proposed product – not in the approach to project management." [3] Changes, risks and uncertainty are inherent part of new product development, and their complete avoidance is neither possible nor necessary. Using Agile/Scrum methodology provides a way to handle changes to the project at any stage.

D. Unlearn Old Skills

1. Planning Everything Up-front
   Unlearn to plan for everything upfront. Requirements are almost never completely known and are always prone to changes as the software is being developed. Thereby, complete and detailed project planning should be replaced by just-in-time planning for only the parts that are of highest values and are best understood at the time of planning.
2. Big Design Up Front (BDUF)
   In a Scrum project, a design need not be completed before implementation can start. Instead of BDUF, the iterative nature of the Scrum process favors the need for Emergent Designs and Architectures, where a just enough design is done in each iteration to support the features that are being delivered, resulting in leaner and maintainable code. The Emergent design techniques depend upon frequent code refactoring. [13]
3. Formal Change Management

A prerequisite of Scrum projects is that the teams are empowered to make decisions through the involvement of the product owner. Formal processes for change management are not needed, and are in fact anathema to a successful Scrum project.

4. Task Mastership
   Discovery, evaluation, triage and prioritization of tasks are responsibilities of a Scrum team. Scrum Masters need to "**learn to let go**" and trust the team to make the right decisions. Studies show that this is the most difficult skill to unlearn while transitioning to Scrum. [14]
5. Triple Constraints
   The PMI Iron triangle of Time, Cost, Scope and Quality (it is really a tetrahedron) is not managed in the traditional manner in a Scrum project. Time and costs are fixed (time-boxed) for all the sprints. Scope is defined at the beginning of a sprint and is then locked down. And Quality is addressed during each sprint by Acceptance testing.

E. Rinse and Repeat

Chances are that a traditional project manager transitioning to Scrum will not get everything right the first time, or the first few times. The Scrum paradigm allows the flexibility to improve the performance of the project based upon the empirical data from the previous sprints and releases. Using the data and first-hand observations, make the adjustments in the next iteration or project and optimize value creation for your customers.

X. Conclusion

The transition from Traditional to Agile Project Management is difficult due to the inherent philosophic differences between the two approaches. There are several concepts that are similar and can be mapped from one to the other, but there are others, like emphasis on upfront planning, which are fundamentally different, and thereby difficult to reconcile between the two. The practitioners who are making the transition from the traditional approaches to Agile would do well to understand and appreciate the philosophic differences between the two approaches and adhere to the underlying basic principles that form the core of the Agile manifesto and movement.

.

References

1. Project Management Body of Knowledge (PMBOK), Fourth Edition
2. Ken Schwaber, Jeff Sutherland, (2008-2010), Scrum Guide
3. Jim Highsmith (2004), *Agile Project Management, Creating Innovative Products*
4. Michele Sliger and Stacia Broderick (2008), *The Software Project Manager's Bridge to Agility*
5. Tamara Sulaiman, Brent Barton, Thomas Blackburn (2006), *AgileEVM – Earned Value Management in Scrum Projects*
6. PMBOK, Fourth Edition*, Section 2.3 – Stakeholders*
7. PMBOK, Fourth Edition*, Appendix G – Interpersonal Skills*
8. Ken Schwaber, Mike Beedle, 2002, *Agile Software Development with Scrum*
9. Stuart Kauffman, 1993, *Origins of Order: Self-Organization and Selection in Evolution*
10. *Illegitimus Non Interruptus,* https://sites.google.com/a/scrumplop.org/published-patterns/team-pattern-language/illigitimus-non-interruptus
11. Jeff Sutherland, Carsten Jakobsen, Kent Johnson, *Scrum and CMMI Level 5: The Magic Potion for Code Warriors*
12. *Money for Nothing, Change For Free, Agile Contracts, http://www.coactivate.org/projects/agile-contracts/money-for-nothing-change-for-free*
13. Scott Bain, 2008, *Emergent Design: The Evolutionary Nature of Professional Software Development*
14. Dina Friis, Jens OStergaard, Jeff Sutherland, *Virtual Reality Meets Scrum: How a Senior Team Moved from Management to Leadership*
15. The Agile manifesto*, http://agilemanifesto.org/*

Dr. Jeff Sutherland is one of the inventors of the Scrum software development process. Together with Ken Schwaber, he created Scrum as a formal process at OOPSLA'95. They have extended and enhanced Scrum at many software companies and IT organizations. Jeff is a Distinguished Graduate of the United States Military Academy, a Top Gun of his USAF RF-4C Aircraft Commander class and flew 100 missions over North Vietnam. Jeff has advanced degrees from Stanford University and Ph.D from University of Colorado School of Medicine. He is currently a Chief executive officer of Scrum, Inc in Boston, Massachusetts and Senior Advisor to OpenView Venture Partners

Mr. Nafis Ahmad is a Software Development Manager, managing multiple teams involved in software engineering projects for the Federal Aviation Authority (FAA). He is a skilled technologist and project manager, having held positions of CTO, CIO and Technology Director of several Boston area companies. Mr. Ahmad has a Masters in Computer Science from Texas &M University, and a Bachelor's in Computer Engineering. [redacted]

.