How a Traditional Project Manager Transforms to Scrum
– Jeff Sutherland & Nafis Ahmad

August 11th, Agile 2011, Salt Lake City
Brief Bios

• Dr. Jeff Sutherland (jeff@scruminc.com)
  • Co-inventor of Scrum
  • Chairman of Scrum Foundation and CEO of Scrum Inc.
  • Consultant and Advisor to many companies on Scrum and Agile
  • Expert at implementing Scrum in all environments, including CMMI Level 5 companies
  • Former CTO/VP of 9 software companies

• Nafis Ahmad (nafis_ahmad@hotmail.com)
  • Software Development Manager for SGT, developing Software for FAA, at the Volpe Center (DOT), Cambridge MA
  • Project Management Professional (PMP) & Certified Scrum Master (CSM)
  • Former CTO/VP of 3 software companies
  • Uses and espouses Traditional and Agile Software Methodologies and processes in formal environments
Contents

• Define Traditional Project Management
• Define PMI and PMBOK
• Provide a mapping between PMBOK and Agile/Scrum
• Transition plan - Organizational
• Transition plan - Individual
• Conclusions
• Q&A
Traditional Project Management (TPM)

- **Phased**: Divided in phases, with ‘technical’ handoffs to transition
- **Sequential**: A phase starts when previous phases are completed and ‘perfected’
- **Non-iterative**: One shot to get everything right
- **Plan-driven**: Develop a plan and then execute according to the plan
- Project flows downwards like a **waterfall**
PMBOK

- Traditional Project Management is mainly based upon the standards of PMI
- Project Management Institute (PMI) was founded in 1969 at Georgia Tech
- PMBOK (PM Body of Knowledge) defines project management standards used worldwide
- PMBOK does not explicitly prescribe a methodology but waterfall is generally used by PMI practitioners
PMI - Project Manager (PMBOK)

• As per the latest PMBOK 4\textsuperscript{th} Edition (2008), Section 2.3 (Stakeholders), a PMI project manager is responsible for:
  • 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.

• PMI project manager plays a central role between project stakeholders and the project itself
• PMBOK defines 5 Process Groups
  • Initiation
  • Planning
  • Execution
  • Monitoring & Controlling
  • Closing
• The Planning, Executing and Controlling Processes of PMBOK maps to the Deming ‘Plan-Do-Check-Act’ Cycle
• PM is responsible for determining which processes are appropriate and the rigor for each process
And 9 key knowledge areas:

- Integration
- Scope
- Time
- Cost
- Quality
- Human Resources
- Communications
- Risk
- Procurement

Source: Project Management Body of Knowledge
Mapping PMBOK to Agile/Scrum

• Agile projects can be mapped to PMBOK, since:
  • PMBOK does not prescribe any methodology; PM determines what processes are appropriate and to what degree applicable to the project
  • A phase defined in the PMBOK is similar to a Scrum Release
  • The sub-phases of a project can be mapped to individual iterations or sprints

• Main adjustments:
  • Consider each sub-phase of a traditional project to be a complete cycle of design, development and test, resulting in working software (mini-waterfalls)
  • Consider Requirements, Design, Development, Testing and Deployment to be activities and NOT phases, where each phase encompass all of these activities, always resulting in working software
## Mapping PMBOK Knowledge Areas

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>PMI Process/Practice</th>
<th>Scrum Process/Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INTEGRATION MANAGEMENT</strong></td>
<td>Develop a <strong>Project Charter</strong> to justify, define, authorize project</td>
<td>Product Owner and Scrum team develops <strong>Product Roadmap, Vision and Backlog</strong></td>
</tr>
<tr>
<td></td>
<td>Develop a formal <strong>Project Management Plan</strong> at the start (15 component plans)</td>
<td>Scrum Team develops a high-level release plan and more detailed plan for the next sprint (<strong>Rolling wave Planning</strong>)</td>
</tr>
<tr>
<td></td>
<td><strong>Direct and Manage</strong> Project Execution (Conformance to plan, CRs, PP updates) <strong>Monitor and Control</strong> Project Work (CRs, PP updates)</td>
<td>Scrum Team executes and delivers; ScrumMaster manages <strong>Scrum principles</strong>, which in turn manage the teams Scrum team self-manages by using sprint reviews and retrospectives and adjusts to changes - <strong>continuous improvement</strong></td>
</tr>
<tr>
<td></td>
<td>Perform <strong>Integrated Change Control</strong> (via CCB and CC system)</td>
<td>Change control by Product Owner and Scrum team via the <strong>(ranked) product backlog</strong>, constant feedback during iteration and review</td>
</tr>
<tr>
<td></td>
<td><strong>Close Project</strong> or Phase (administrative closure/audits)</td>
<td>Sprint reviews/project retrospectives; Sprint N+1 for admin closure or audits if necessary</td>
</tr>
</tbody>
</table>

**Bottom Line:** A Traditional PM is responsible for bringing all processes together into a unified whole, yet a ScrumMaster has a **lighter touch** as the team is responsible for planning, execution and delivery; a Traditional PM needs to **learn to give up control**
# Mapping PMBOK Knowledge Areas

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>PMI Process/Practice</th>
<th>Scrum Process/Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SCOPE MANAGEMENT</strong></td>
<td><strong>Collect Requirements</strong> (Requirements, RMP, RTM)</td>
<td>Develop and prioritize <strong>Product Backlog</strong> items</td>
</tr>
<tr>
<td></td>
<td><strong>Define Scope</strong> (Scope statement – deliverables, exclusion/inclusions, assumptions/constraints)</td>
<td>Select <strong>Product Backlog</strong> items for the release or sprints</td>
</tr>
<tr>
<td></td>
<td>Create <strong>Work Breakdown Structure</strong> (WBS, WBS diagram, WBS dictionary)</td>
<td>Create a <strong>Feature Breakdown Structure</strong> for the release, showing features for each release. Further break it down into individual features (scenarios) per sprint</td>
</tr>
<tr>
<td></td>
<td><strong>Verify Scope</strong> (accepted features, CRs)</td>
<td>Via feature acceptance (by Product Owner); use product backlog and traceability tools</td>
</tr>
<tr>
<td></td>
<td><strong>Control Scope</strong> (Change Control)</td>
<td>Manage via product backlog and product owner; protect the iteration</td>
</tr>
</tbody>
</table>

**Bottom Line:** Scope management is inherently **built into** the Scrum process. Scrum keeps Time and Costs fixed, the only negotiable item is Scope which is fixed at the beginning of the sprint; it **solves the intractable TPM iron triangle of Time, Scope and Cost.**
## Mapping PMBOK Knowledge Areas

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>PMI Process/Practice</th>
<th>Scrum Process/Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TIME MANAGEMENT</strong></td>
<td>Define Activity <em>(Iteration-level, activity list, lowest level in WBS)</em></td>
<td>Features are selected for a sprint by the team; tasks are identified to accomplish the features</td>
</tr>
<tr>
<td></td>
<td>Sequence Activities <em>(Network Diagram, Critical Path)</em></td>
<td>Conducted by team during sprint planning meetings; estimation of tasks to complete a story</td>
</tr>
<tr>
<td></td>
<td>Estimate Activities</td>
<td>An overall Release Schedule is developed; Only the features targeted for the sprints are elaborated and estimated <em>(Just-In-Time planning)</em> Estimates are refined based on empirical data (team velocity)</td>
</tr>
<tr>
<td></td>
<td>Schedule Development <em>(Strategic level, project schedule and baseline)</em></td>
<td>Team manages what features are developed in which sprint</td>
</tr>
<tr>
<td></td>
<td>Control Schedule <em>(PP updates, SV, SPI)</em></td>
<td></td>
</tr>
</tbody>
</table>

**Bottom Line:** Scrum uses a more **Top-down approach for time management** – plan releases, then individual sprints and then daily activities/tasks; chose the highest value features as opposed to executing tasks defined in a project plan.
## Mapping PMBOK Knowledge Areas

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>PMI Process/Practice</th>
<th>Scrum Process/Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COST MANAGEMENT</strong></td>
<td><strong>Cost Estimation</strong> (Activity cost estimation for <strong>summary and detail</strong> using Analogous, Parametric, 3-point (PERT) estimation, Expert Judgment or PM software)</td>
<td>Perform <strong>top-down estimation</strong> of the releases and sprints, using Project Velocity, Ideal Days, Analogy, Expert Opinion or Disaggregation. Perform a <strong>bottom-up estimation</strong> of the sprint in question to validate or fine-tune the top-down estimates. <strong>Refine the estimates</strong> further accounting for team changes, esoteric/new functionality and new technology. Add a Feature or Schedule buffer</td>
</tr>
<tr>
<td></td>
<td><strong>Cost Budgeting</strong> (Cost performance baseline)</td>
<td>Create a Cost Baseline after doing the above; revise the cost baseline after a couple of sprints (when <strong>actual team velocity</strong> is known)</td>
</tr>
<tr>
<td></td>
<td><strong>Cost Control</strong> (Earned Value Management, CPI, EAC)</td>
<td>Use <strong>Product Burndown Charts</strong> as a Cost controlling aide; use AgileEVM in more formal environments</td>
</tr>
</tbody>
</table>

**Bottom Line:** Estimates are performed **top-down** and are refined throughout the project’s lifecycle; cost control is a **team function** with product owner involved in estimation with the team
# Mapping PMBOK Knowledge Areas

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>PMI Process/Practice</th>
<th>Scrum Process/Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUALITY MANAGEMENT</td>
<td><strong>Quality Planning</strong> (Quality Management Plan, Process Improvement Plan, Quality Metrics)</td>
<td>Quality is implicit through Scrum practices (Definition of Done, early &amp; frequent testing, working software, impediment removal, coding/testing standards, metrics) Quality is the responsibility of the whole cross-functional Scrum team with committed QA resources</td>
</tr>
<tr>
<td></td>
<td><strong>Quality Assurance</strong> (quality audits, process analysis, QA department)</td>
<td>Generally, performed by the team In formal environments, a 3rd party can be engaged to perform QA as part of an extra sprint (Sprint N+1) to fulfill regulatory and compliance requirements Use sprint reviews and release/project retrospectives</td>
</tr>
<tr>
<td></td>
<td><strong>Quality Control</strong> (7 basic tools of QC, Statistical sampling, validated deliverables, QC department)</td>
<td>Performed by the team itself using Unit Testing or Test-driven Development (developers), integration and feature testing by testers and user acceptance testing (product owner) and automated testing Use Burndown Charts to monitor trends of feature development Add acceptance tests as part of product backlog</td>
</tr>
</tbody>
</table>

**Bottom Line:** No formal Quality planning in Scrum yet **Quality is built into the fabric of Scrum** due to the nature of processes and practices and cross-functional nature of Scrum teams
## Mapping PMBOK Knowledge Areas

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>PMI Process/Practice</th>
<th>Scrum Process/Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HUMAN RESOURCE MANAGEMENT</strong></td>
<td>Human Resource Planning (HR Plan using project schedule and PP)</td>
<td>Plan for the team size based upon the project needs for the <strong>entire project duration</strong>&lt;br&gt;Plan for 7 (plus or minus 2) <strong>fully-dedicated</strong> members&lt;br&gt;Split the project into multiple teams if the scope is large</td>
</tr>
<tr>
<td></td>
<td>Acquiring a Project Team (Staffing Mgmt Plan: staff acquisition, calendars, staff release)</td>
<td>Develop a <strong>dedicated cross-functional</strong> team at the start of the project and keep it intact for the duration of the project</td>
</tr>
<tr>
<td></td>
<td>Develop the Project Team (team-building, training, soft skills development, performance assessments)</td>
<td>Use Agile and <strong>Scrum Values</strong> (commitment, openness, focus, courage and respect) to develop and build team&lt;br&gt;Foster self-organization in team building; co-location</td>
</tr>
<tr>
<td></td>
<td>Manage the Project Team (updates to PP and org, corrective/preventative actions)</td>
<td>Facilitate and coach the <strong>self-managing Scrum team</strong>&lt;br&gt;by providing real-time feedback to the team&lt;br&gt;Play the role of a <strong>servant-leader</strong></td>
</tr>
</tbody>
</table>

**Bottom Line:** Scrum teams are **dedicated, cross-functional and self-organizing**, with mutual accountability which necessitates a servant-leadership model for Scrum Masters.
## Mapping PMBOK Knowledge Areas

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>PMI Process/Practice</th>
<th>Scrum Process/Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMMUNICATION</strong></td>
<td><strong>Identify Stakeholders</strong> (stakeholder register and management strategy)</td>
<td>Identify the stakeholders and embed a business representative <em>(Product Owner)</em> in the Scrum team itself</td>
</tr>
<tr>
<td></td>
<td><strong>Communication Planning</strong> (Comm. Mgmt Plan)</td>
<td>Release/Sprint backlogs and <em>Burndown Charts</em> are <em>Visual Indicators</em> of project status</td>
</tr>
<tr>
<td></td>
<td><strong>Information Distribution</strong></td>
<td>Visual Indicators of project status are <em>Information Radiators</em></td>
</tr>
<tr>
<td></td>
<td><strong>Manage Stakeholder expectation</strong></td>
<td>Stakeholder management is done via Product Owners who are part of the Scrum team</td>
</tr>
<tr>
<td></td>
<td><strong>Performance Reporting</strong> (EVA, histograms, S-curves)</td>
<td>Cost and Schedule are steady and predictable, use Release/Sprint Burndown Charts to show real-time performance of feature development, i.e. Visual Indicators of project status</td>
</tr>
</tbody>
</table>

**Bottom Line:** This is a **major difference** between Traditional and Scrum/Agile projects, since the cross-functional nature of a Scrum team creates and emphasizes **direct/face to face and frequent communication** within team members and business stakeholders, obviating the need for traditional reporting; less formality of a Scrum project results in better communication.
# Mapping PMBOK Knowledge Areas

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>PMI Process/Practice</th>
<th>Scrum Process/Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RISK MANAGEMENT</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk Planning</td>
<td>(Risk Management Plan – methodology, risk categories, risk (probability x impact) matrix, tolerances, reporting)</td>
<td><strong>Informal risk planning</strong> as part of sprint/release planning and review meetings <strong>Entire team</strong> is involved in risk planning, mitigation and response</td>
</tr>
<tr>
<td>Risk Identification</td>
<td>(Risk Register)</td>
<td>Identify risks in daily scrums, iteration/release planning and reviews; Perform ad hoc SWOT Analysis, Checklists, Brainstorming</td>
</tr>
<tr>
<td>Quantitative/Qualitative Risk Analysis</td>
<td>(risk register updates)</td>
<td>No formal method prescribed; risk matrices (probability x impact) can be developed for special risks if needed</td>
</tr>
<tr>
<td>Risk Response Planning</td>
<td>(strategy for +/-ve risks, Avoidance, Mitigation, Acceptance, Transfer)</td>
<td><strong>Avoidance</strong> (change scope or resources), <strong>Mitigation</strong> (POC), <strong>Transfer</strong> (Outsource), <strong>Acceptance</strong> (Contingency plans)</td>
</tr>
<tr>
<td>Monitor and Control Risks</td>
<td></td>
<td>As part of the team planning and review</td>
</tr>
</tbody>
</table>

**Bottom Line:** Scrum is a Risk Reduction system that handles risk at **strategic** (rapid response to change, time to market) and **tactical** (product development risk via issues/impediment lists) level
### Mapping PMBOK Knowledge Areas

<table>
<thead>
<tr>
<th>Knowledge Area</th>
<th>PMI Process/Practice</th>
<th>Scrum Process/Practice</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PROCUREMENT MANAGEMENT</strong></td>
<td><strong>Plan Procurements</strong> (procurement mgmt. plan – make or buy, criteria, SOWs)</td>
<td>Team provides input for describing needs for procurement using early iterations or Proof of concepts (POCs)</td>
</tr>
<tr>
<td></td>
<td><strong>Conduct Procurements</strong> (awards)</td>
<td>Team conducts evaluations and provides input into contract documentation</td>
</tr>
<tr>
<td></td>
<td><strong>Administer Procurements</strong> (procurement documentation, PP updates)</td>
<td>Scrum allows for contracts with ‘early termination’ clause, referred to as ‘Money for Nothing’ contract type – a customer can terminate a contract at the end of any sprint by paying 20-30% of remaining contract value ‘Change for Free’ contract type is used so that a customer can make changes to scope without incurring any additional costs if total scope of contracted work is not changed</td>
</tr>
<tr>
<td></td>
<td><strong>Close Procurements</strong></td>
<td>An additional sprint (Sprint N+1) may be used for formal administrative closure; exit retrospectives</td>
</tr>
</tbody>
</table>

**Bottom Line:** Scrum plays a more active role in evaluating and selecting the sellers, often engaging in a proof of concept as part of the early sprints; several contract types are used in Scrum projects to provide flexibility for the customers.
The Transition: Organizational Game Plan

• **Use an Incremental Delivery Approach**
  • Divide the project into mini-waterfalls, delivering the product in increments
  • Useful in more formal environments with set milestones, like Federal agencies

• **Apply Agile Techniques Incrementally**
  • Introduce Agile techniques one by one
  • The Google AdWords development group applied Agile techniques piecemeal, successfully over a period of 6 months
  • Applicable in environments that are looking to gradually transition towards more Agile methodologies

• **Switch to a Pure Agile Approach**
  • Approach with most benefits but most radical transformation
  • Requires an organizational mind shift
  • Ideal for companies with intense time-to-market needs

Bottom Line: **Balance Agility and Control** based upon the needs and nature of the organization
The Transition: Individual Game Plan

• **Understand the Similarities**
  • **Similar goals** – deliver a product within budget and schedule, and with the highest possible quality
  • **Project Phases and Sub-Phases** exist in both TPM and Agile/Scrum projects; interpretation of what a phase/sub-phase is supposed to achieve is different
  • **Good Software Engineering Practices** can be used in both methodologies, e.g.
    • Continuous Integration
    • Unit Testing
    • Test-Driven Development,
    • Pair Programming, etc.
The Transition: Individual Game Plan

- **Understand the Differences**
  - **Value vs. Plan driven Approach**
    - Provide the highest value features instead of developing and following a project plan
  - **Empirical vs. Prescriptive Approach**
    - Use project results and metrics to drive the project instead of a prescriptive plan
  - **Self-Organizing vs. directed teams**
    - Adopt a lighter touch, remove impediments, and let the team self-manage
  - **Stakeholder management vs. Stakeholder involvement**
    - Product owner is an integral part of the team
  - **Project Manager vs. ScrumMaster**
    - Servant-leadership instead of task mastership
  - **Top Down Planning vs. Bottom-Up Planning**
    - Use the principle of Progressive Elaboration
The Transition: Individual Game Plan

• **Learn new skills**
  • **Servant-Leadership**
    • Remove impediments for the team; team is empowered to make decisions
  • **Foster Collaboration**
    • Promote self-organization, self-discipline, respect for individuals, conflict resolution and technical competency
  • **Balance Flexibility and Stability**
    • Too much structure stifles creativity; too little structure breeds inefficiency
  • **Scrum values**
    • Commitment, Openness, Focus, Courage and Respect
  • **Embrace changes/risks/uncertainty**
    • Complete avoidance of all the risks and uncertainty is neither possible nor necessary; learn to embrace it
    • Agile concept: “opportunity, uncertainty and risk reside in the proposed product – not in the approach to project management”
The Transition: Individual Game Plan

- **Unlearn old skills**
  - Planning everything up-front
    - Use Just-In-Time planning for what is known instead of ‘predicting’ everything upfront
  - **Big design Up Front (BDUF)**
    - Requirements are never completely known upfront, and prone to change
    - Use Emergent designs and architectures
  - **Formal Change Management**
    - Use Product Owner and product backlog
  - **Task Mastership – learn to let go**
    - Lighter touch; manage the Scrum principles, which will govern the team
  - **Triple Constraints**
    - Time and Cost are frozen, only Scope is variable and is frozen at the start of an iteration
The Transition: Individual Game Plan

• **Rinse and Repeat**
  • Rinse
    • Use Agile Reviews and Retrospectives at the end of an iteration, release or project to see what worked and what did not
    • Make the changes
  • Repeat
    • Iterate
    • Continuous learning
  • There is no perfect process or methodology; keep learning and adapting
Conclusions

• Transitioning is difficult due to inherent philosophic differences and requires more discipline
• Understand the mapping between Traditional and Agile project management
• Identify the similarities and differences; learn the balance between agility and control
• Develop individual and organizational skills, culture and environment for the transition
• Stick to the basic agile principles and look for ways to produce value for the customers rather than focus on following an agile or traditional process or practice