

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

August 9th, Agile 2011, Salt Lake City

Brief Bios

- **Dr. Jeff Sutherland**

- 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**

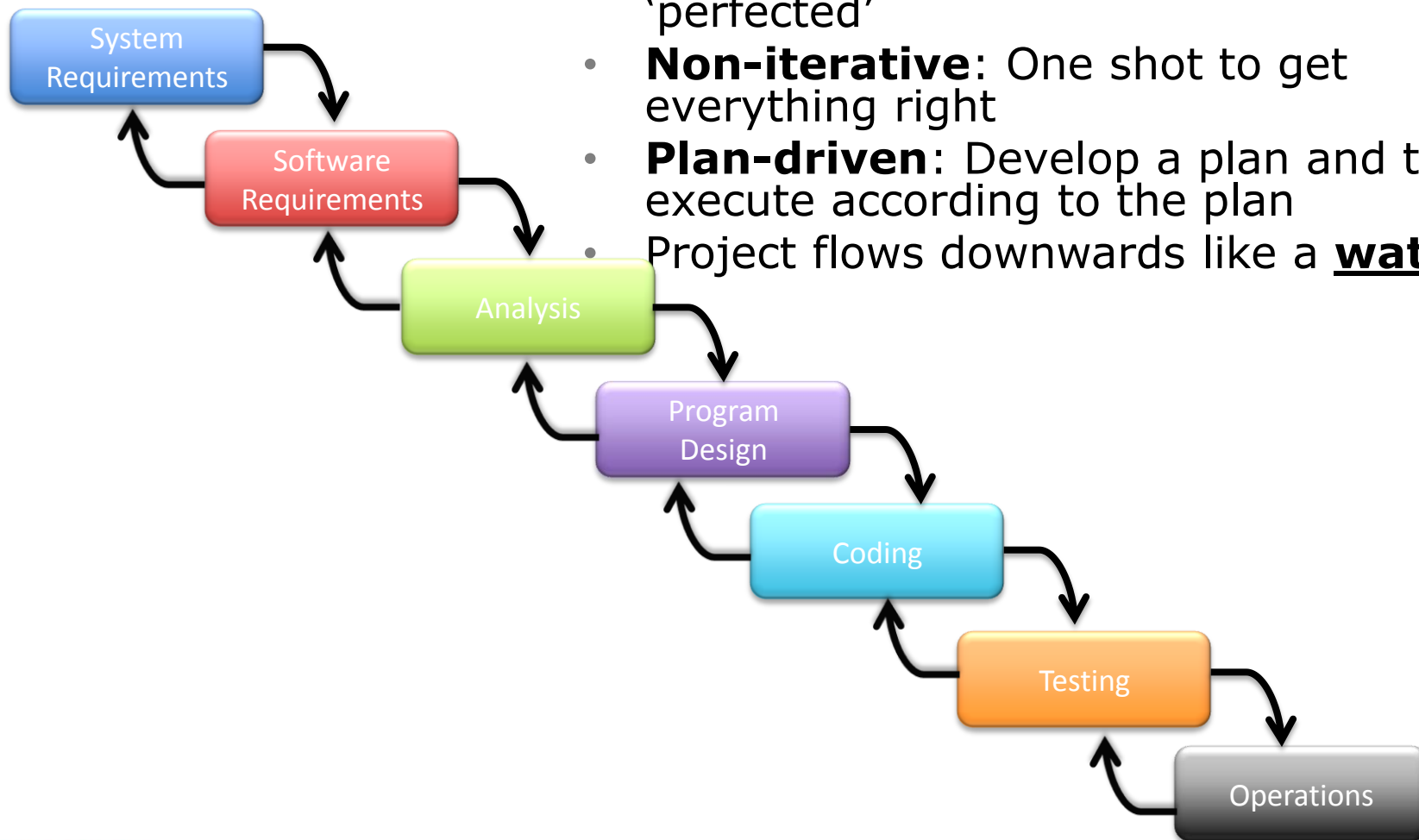
- Software Development Manager for SGT, developing Software for FAA, at the Volpe Center (DOT), Cambridge MA
- Project Management Professional (PMP)
- Certified Scrum Master (CSM)
- Uses and espouses Traditional and Agile Software Methodologies and processes in formal environments

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

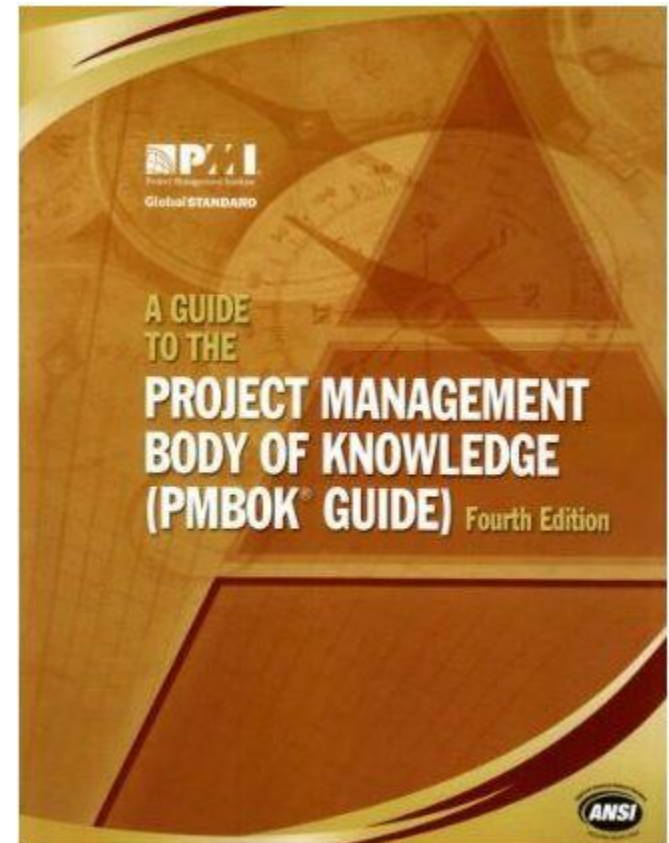
Traditional Project Management (TPM)

- **Phased:** Divided in phases, with handoffs to transition
- **Sequential:** One 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**

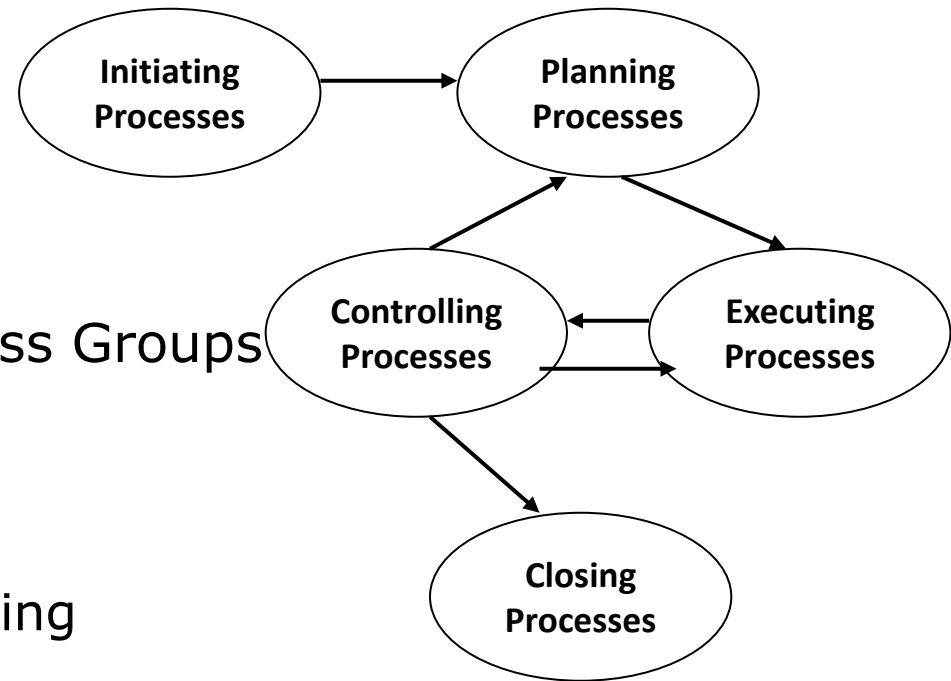


PMBOK

- Project Management Institute (PMI) was founded in 1969 at Georgia Tech
- Traditional Project Management is mainly based upon the standards of PMI
- 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



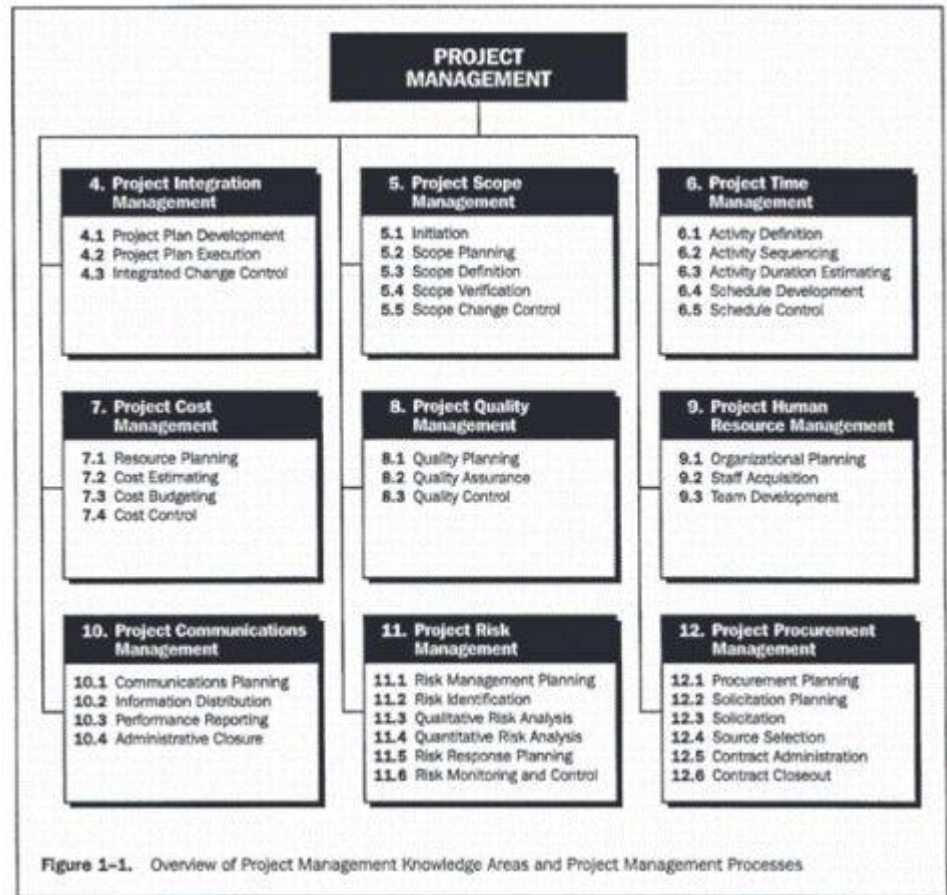
PMBOK (Cont.)



- PMBOK defines 5 Process Groups
 - Initiation
 - Planning
 - Execution
 - Monitoring & Controlling
 - Closing
- The Planning, Executing and Controlling Processes use the Deming 'Plan-Do-Check-Act' Cycle
- PM is responsible for determining which processes are appropriate and the rigor for each process

PMBOK (Cont.)

- And 9 knowledge areas
 - Integration
 - Scope
 - Time
 - Cost
 - Quality
 - Human Resource
 - Communication
 - Risk
 - Procurement



Source: Project Management Body of Knowledge

PMI - Project Manager

- As per the latest (2008) PMBOK Edition (4th), 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

Mapping PMBOK to Agile/Scrum

- Agile projects can be mapped to PMI concepts
 - 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

Knowledge Area	PMI Process/Practice	Scrum Process/Practice
Integration Management	Develop Project Charter (Project Manager)	Develop Product roadmap (Product Owner and Scrum team) using enterprise backlog
	Develop Project Management Plan (Project Manager) at the start	Develop a high-level release plan and more detailed plan for the next iteration (Scrum Team) - Just-In-Time Planning
	Direct and Manage Project Execution (CRs, PP updates)	Scrum Team executes and delivers; ScrumMaster manages Scrum principles , which in turn manage the teams
	Monitor and Control Project Work (CRs, PP updates)	Scrum team self-manages by using sprint reviews and retrospectives and adjusts to changes - continuous improvement
	Perform Integrated Change Control (via CCB and CC system)	Change control by Product Owner and Scrum team via the (ranked) product backlog , constant feedback during iteration and end of iteration demo and review
	Close Project or Phase (administrative closure/audits)	Sprint reviews/project retrospectives; Sprint N+1 for audits if necessary

Bottom Line: 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

Knowledge Area	PMI Process/Practice	Scrum Process/Practice
Scope Management	Collect Requirements (Requirements Mgmt Plan, RTM)	Develop and prioritize Product Backlog items
	Define Scope (Scope statement – deliverables, exclusions/inclusions)	Select Product Backlog items for the release
	Create Work Breakdown Structure (WBS)	Create a Feature Breakdown Structure for the release, showing features for each release. Further break it down into individual features (scenarios) per sprint
	Verify Scope (accepted features, CRs)	Via feature acceptance (by Product Owner); use product backlog and traceability tools
	Control Scope (Change Control)	Manage via product backlog and product owner

Bottom Line: All the Scope management activities are enforced by the Traditional Project Manager, whereas 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

Knowledge Area	PMI Process/Practice	Scrum Process/Practice
Time Management	Define Activity (lowest level in WBS)	Features are selected for a sprint by the team; tasks are identified to accomplish the features
	Sequence Activities (network diagram)	Conducted by team during sprint planning meetings; estimation of tasks to complete a story
	Estimate Activities	
	Schedule Development (project schedule and baseline)	An overall Release Schedule is developed; Only the features targeted for the sprints are elaborated and estimated Estimates are refined based on empirical data (team velocity)
	Control Schedule	Team manages what features are developed in which sprint

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

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

Bottom Line: Scrum provides deeper insight into the project through direct involvement of business with the team; cost control is a team function with product owner involved in estimation with the team

Mapping PMBOK Knowledge Areas

Knowledge Area	PMI Process/Practice	Scrum Process/Practice
Quality Management	Quality Planning (Quality Management Plan, Quality Metrics)	Quality is implicit through Scrum practices (Definition of Done, early & frequent testing, working software, impediment removal, etc) Quality is the responsibility of the whole cross-functional Scrum team with committed QA resources
	Quality Assurance (quality audits, QA department)	Generally, performed by the team In formal environments, a 3 rd 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
	Quality Control (Statistical QC, QC department, validated deliverables)	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) Use burndown charts to monitor trends of feature development Add acceptance tests as part of product backlog

Bottom Line: Quality Assurance and Control are an integral part of Scrum due to cross-functional nature of Scrum teams; Quality is built into the fabric of a Scrum process and team

Mapping PMBOK Knowledge Areas

Knowledge Area	PMI Process/Practice	Scrum Process/Practice
Human Resource Management	Human Resource Planning (HR plan)	Plan for the team size based upon the project needs and organizational policies Plan to have 7 plus or minus two people; Split the project into multiple teams if the scope is large
	Acquiring a Project Team (resource assignments)	Develop a cross-functional team at the start of the project and keep it intact for the duration of the project
	Develop the Project Team (team-building, training, performance assessments)	Use Agile and Scrum Values (commitment, openness, focus, courage and respect) to develop and build team Foster self-organization in team building
	Manage the Project Team (updates to PP and org)	Facilitate and coach the self-managing Scrum team by providing real-time feedback to the team Play the role of a servant-leader

Bottom Line: Scrum teams are cross-functional and self-organizing which necessitates a servant-leadership model for Scrum Masters

Mapping PMBOK Knowledge Areas

Knowledge Area	PMI Process/Practice	Scrum Process/Practice
Communication Management	Identify Stakeholders (stakeholder mgmt strategy)	Identify the stakeholders and embed a business representative (product owner) in the Scrum team itself
	Communication Planning (Communication Management Plan)	Release/Sprint backlogs and burndown charts are Visual Indicators of project status
	Information Distribution	Visual Indicators of project status are Information Radiators
	Manage Stakeholder expectation	Stakeholder management is done via Product Owners who are part of the Scrum team
	Performance Reporting (EVM, histograms, S-curves)	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

Bottom Line: This is a major difference between TPM and Scrum/Agile project, 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. Ironically, less formal communication of a Scrum project results in better communication

Mapping PMBOK Knowledge Areas

Knowledge Area	PMI Process/Practice	Scrum Process/Practice
Risk Management	Risk Planning (risk management plan)	Informal risk planning as part of sprint/release planning and review meetings Entire team is involved in risk planning, mitigation and response
	Risk Identification (risk register)	Identify risks in daily scrums, iteration reviews/planning and release reviews/planning Perform ad hoc SWOT Analysis, Checklists, Brainstorming
	Quantitative/Qualitative Risk Analysis (risk register updates)	No formal method prescribed; risk matrices (probability x impact) can be developed for special risks if needed
	Risk Response Planning (updates to PP, risk register)	Avoidance (change scope or resources), Mitigation (POC), Transfer (Outsource), Acceptance (Contingency plans)
	Monitor and Control Risks	As part of the team planning and review

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

Knowledge Area	PMI Process/Practice	Scrum Process/Practice
Procurement Management	Plan Procurements (procurement plan)	Early iterations would result in describing needs for procurement
	Conduct Procurements (awards)	Proof Of Concepts (POCs) are conducted as part of early iterations to qualify or select a product
	Administer Procurements (procurement documentation, PP updates)	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
	Close Procurements	An additional sprint (Sprint N+1) may be used for formal administrative closure

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

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

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, 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
 - Agile concept: “opportunity, uncertainty and risk reside in the proposed product – not in the approach to project management”
 - Complete avoidance of all the risks and uncertainty is neither possible nor necessary; learn to embrace it

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 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
- Understand the mapping between Traditional and Agile project management
- Identify the similarities and differences
- 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

Q&A