



Agile EVM Lessons Learned

PARCA EVM & Agile Meeting

February 20, 2015

Presented by: Dave Scott

Vice President

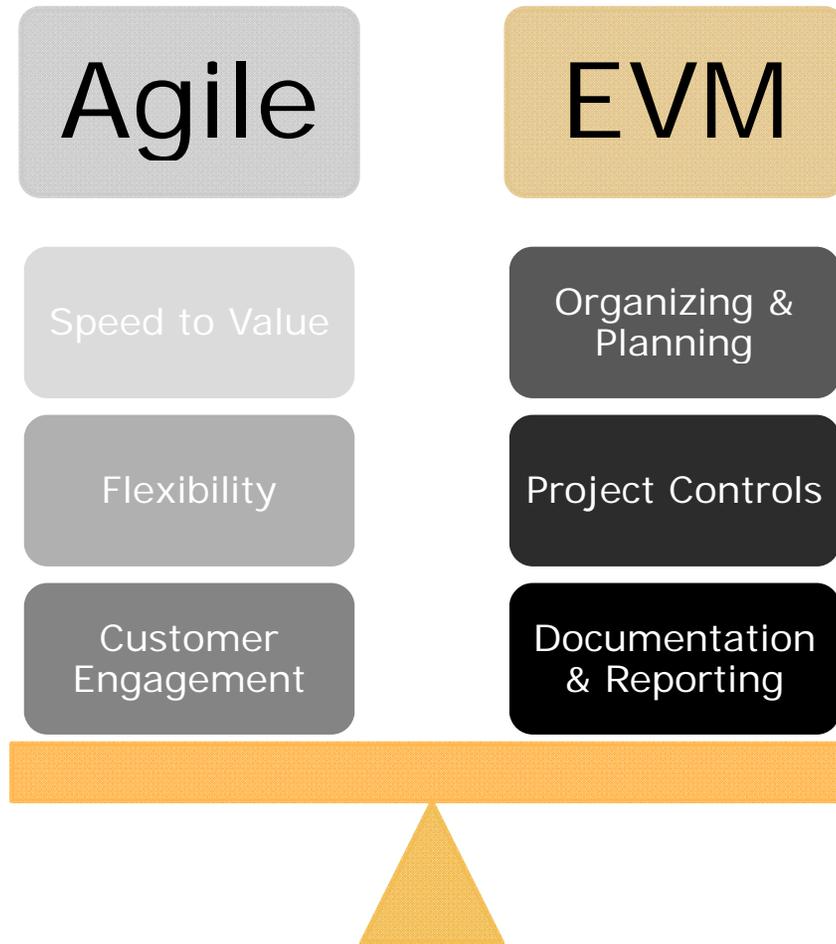
KM Systems Group

dscott@kmsystemsgroup.com

Agenda

- **Agile and EVM Balancing Act**
- Lessons Learned
- Pulling it all Together

Agile & EVM Balancing Act



Impacts: People, Processes, & Tools

People Impacts

- Smaller more nimble & autonomous teams
- Receive early customer feedback and identify failures fast
- Help the customer understand their real requirements
- Agility to quickly adapt to changing requirements

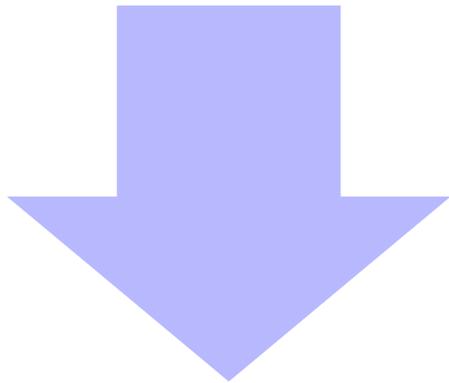
***Enable Agile Teams, Project Controls,
and Stakeholders***

Processes and Tools Impacts

- Must scale how the organization addresses the ANSI/EIA 748 EVMS standard five categories
 1. Organization
 2. Planning, Scheduling, & Budgeting
 3. Accounting Considerations
 4. Analysis and Management Reporting
 5. Revisions and Data Revisions
- May need to implement and integrate specialized tools for Agile, EVM, Scheduling, and Reporting

Business Value

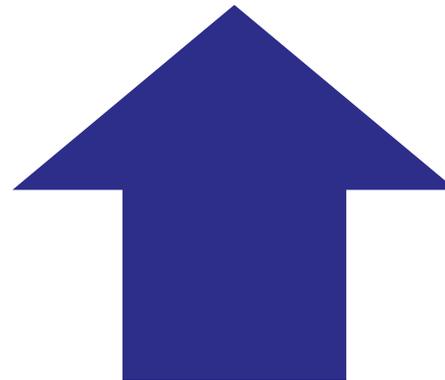
Government



Enable the
Mission, Timely
Delivery,
Achieve Cost
Objectives



Successful
Delivery,
Customer
Satisfaction,
Repeat Business



Contractor

Agenda

- Agile and EVM Balancing Act
- **Lessons Learned**
- Pulling it all Together

Lessons Learned

- Disciplined Release Planning and Estimating is Imperative
- Sprint Planning, Execution, and Measuring Technical Performance is Critical
- Critical Path Scheduling is less effective and Risk Management is important

Customer/Stakeholder engagement processes should be defined and agreed

Myth: Agile Projects Don't Plan



“If you don't know where you are going,
you'll end up someplace else.”

— Yogi Berra

- Plans guide our investment decisions
- Plans help us know who needs to be available to work on a project
- Plans help us know if a project is on track to deliver the functionality that users need

Mike Cohn, founding member of the Agile Alliance

Agile Estimating and Planning

Agile Planning Elements

- Core Capabilities – Mission capabilities to be supported
- EPIC – significant future stakeholder requirements which typically span releases
- Sub-EPIC (Features) backlog – all defined stakeholder features. Usually fit in releases.
- User Stories – features are broken down into User Stories which are developed by the Scrum Team and fit with within a sprint.
- Story Points – estimate of effort to develop user stories based on past performance or experience

Release Planning and Estimating

Lessons Learned

- Release Roadmap - document minimum deployment capabilities 'must have' as well as 'like to have'
 - Develop Product Backlog and prioritization with Stakeholders
 - Features are assigned to Releases
 - Related Features are assigned to Control Accounts
- Product Owner/Project Engineer (CAM) responsible for scope (backlog), schedule, budget

Release Planning and Estimating Lessons Learned (cont.)

Release 1 (Capabilities)

Release 2

Release n

Control Accounts (CA) Features &
Budgets

Summary
Level
Planning
Packages
(SLPP)

Summary
Level
Planning
Packages
(SLPP)

Discrete
Work
Packages
(WP)

Discrete
Work
Packages
(WP)

Discrete
Work
Packages
(WP)

Release Planning and Estimating Lessons Learned (cont.)

- Develop cost estimates for all Features
- Develop user stories for near term Sprints, at a minimum typically target 150% of user story capability
- Features are in summary level planning packages
- IBR's are important to validate release plans, baselines, and risk management

Sprint Planning, Execution, & Performance Measurement

- The Product Owner is responsible for maintaining, and prioritizing the product backlog in collaboration with stakeholders and the development team
- Product backlog management is important, cost estimates for new Features, frequent EAC's important
- The Product Owner along with the Scrum team break out additional Features into User Stories required for future sprints
- The Product Owner in collaboration with the Scrum teams decides which user stories are to be completed in the immediate and near-term sprints
- User story estimates will be rolled into the monthly budget by work package

Sprint Planning, Execution, & Performance Measurement (cont.)

- Estimating based on Story Points is a good gauge of the teams perspective regarding complexity and risk
- Story Points are also useful for determining the number of Users Stories which the team can complete in a Sprint
- But budget derived from average story point cost per team doesn't translate consistently at the Program level
- This is further exacerbated by the volatility of Agile teams
- Should consider estimating in Hours and multiply it times the labor hour rates

Sprint Planning, Execution, & Performance Measurement (cont.)

- The Product Owner and the team decompose user stories into tasks and the team finalizes task assignments
- The team agrees to a testing plan and exit criteria for each user story and task
- This information is captured in Agile PM system, where dependencies between user stories may also be identified and tracked
- With Sprints planned, the user stories (and associated hours) can be aggregated and rolled up to determine monthly budget
- Remaining Features are in the Backlog (SLPP)

Sprint Planning, Execution, & Performance Measurement (cont.)

- User stories not completed in a Sprint which has already started may 'slip' to the next Sprint but result in a variance
- User stories planned for a Sprint not started, may be 'swapped' with user stories from the backlog. Should be equivalent effort.
- User stories moved from one Sprint to another should be treated as re-planning which occurs at the beginning of each Sprint
- *Baseline changes only occur when project deliverables are removed or added resulting in a contract modification*

Sprint Planning, Execution, & Performance Measurement (cont.)

- User Stories which the team agrees on for a Sprint(s) are committed
- User Story performance is measured 0/100 and cannot be split if not completed
- Measure performance based on user stories for the performance period rolling up to work packages
- Sprints are a time box for defined work but aligning them with performance periods is difficult
- Work Packages(WP) roll up to Control Accounts(CA)
- Actual Costs/Hours may be tracked at the WP or CA

Critical Path Scheduling & Risk Management

- Government programs typically receive funding for the entire acquisition lifecycle and not only the software development.
- Apply a hybrid program schedule mix of Agile and Waterfall depending upon the type of project.
- *Results are additive and provide a view of the inter-dependent features and risks.*

Project Scheduling Process

- Agile PM system and the project schedule should share a WBS numbering system so user stories may be mapped to a work packages
- Durations, baseline hours and performance may be transferred from Agile PM system into the project schedule to provide the schedule baseline and schedule status for the Agile efforts
- Actual hours per control account or work package may be integrated from the time sheet system into the project schedule

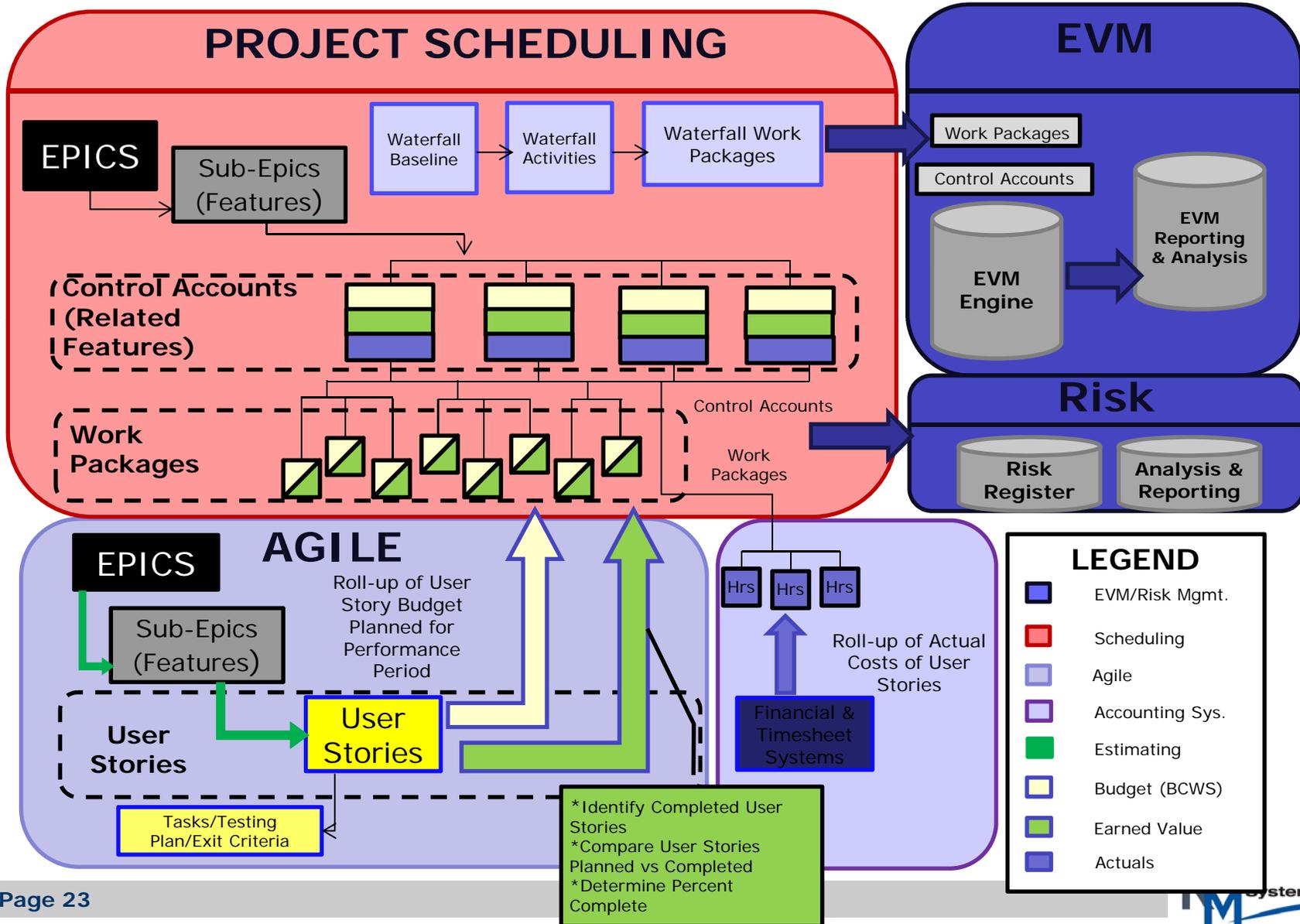
Critical Path Scheduling & Risk Management – Lessons Learned

- User stories cannot be effectively maintained in the project schedule and should be managed in the Agile PM system
- User story dependencies and risks flags should be maintained in the Agile PM system
- The project schedule may not enable an effective critical path so maintaining a risk register is important
 - Identifying and assessing the impact of potential risks
 - Assigning risk responsibility
 - Managing and mitigating risks

Agenda

- Agile and EVM Balancing Act
- Lessons Learned
- **Pulling it all Together**

Agile & EVM - Pulling it All Together





Dave Scott, dscott@kmsystemsgroup.com, (973) 349-2561