PMI – Southern Ontario Chapter
Greater Toronto Information Systems (GTIS) Branch

Presents: How to Apply Earned Value Management for Agile Projects

Presenter: Ivan Papes
Facilitator: Helen Jansons / John Kaldor
Objective

- The Objective of the presentation is to suggest following techniques:
  - Identify criteria when Agile EVM should be utilized
  - How do we gain value for the projects utilizing Agile EVM technique?
  - Walk trough a practical example how to calculate EVM in Agile
  - Best practices to present and explain the results
  - How Agile EVM can be applied at the portfolio level
Agenda

- Review origins and concepts of Earned Value Management (EVM), and its application in traditional projects.
- Concept of Earned Value Management applied in Agile software projects.
- Walk through a practical Agile project EVM calculation
- Discuss and explain results
- How Agile EVM can be applied at the portfolio level?
- Questions and Answers
What is Earned Value Management?

- Earned Value Management (EVM) is a project management technique for measuring project performance and progress in objective manner.
- It identifies the progress and performance of a project against the plan, and estimates future performance. Therefore, EVM can be very useful in project forecasting.
- The *project baseline* is an essential component of EVM and serves as a reference point for all EVM related activities.
- EVM provides quantitative data for project decision making.
History of Earned Value Management

- The concept of Earned Value began in the 1890’s as the early industrial engineers measured performance in American factories. They defined a “cost variance” to relate “earned standards” against “actual expenses” to determine performance.
- It was only in 1962 that Earned Value was formally introduced on projects by the US Navy, as part of the development of the PERT/Cost methodology.
- In 1996, the National Defense Industrial Association (NDIA) developed Earned Value Management System (EVMS), currently embodied in ANSI/EIA 748.
- The Project Management Body of Knowledge (PMBOK), developed by the Project Management Institute, recommends utilizing set of Earned Value criteria, as part of Project Cost and Project Communications Management (Performance Reporting).
Earned Value Management measure the 3 key values:

- **Planned Value (PV):** The budgeted cost for the work scheduled to be completed up to a given point in time.
  - $PV = \text{Hourly Rate} \times \text{Total Hours Scheduled}$
  - As the planned work is accomplished, it’s budgeted value becomes earned value.

- **Earned Value (EV):** The budgeted amount for the work actually completed during a given time period (expressed in monetary value).
  - $EV = \text{Baselined Cost} \times \% \text{Actually Completed}$

- **Actual Cost (AC):** The total actual cost incurred in accomplishing work during a given time (reporting) period.
  - It indicates if it is spend less or more for the work performed up-to-date.
  - $AC = \text{Hourly Rate} \times \text{Total Hours Spent}$
Calculating Project Performance (CPI)

- The PV, EV, and AC values are combined in various ways to provide project performance metrics.
- Cost Performance Indicator (CPI) is used to show efficiency of project resources utilization. It is calculated as:
  - CPI = EV / AC
    - CPI > 1 = good resource utilization
    - CPI < 1 = indicates resource utilization issues
- BAC: Budget at completion of project
- EAC: Estimate at completion, or expected total cost of the project at its end (EAC = BAC / CPI)
Calculating Project Performance (SPI)

- The Schedule Performance Index (SPI) is a measure of schedule efficiency expressed as the ratio of earned value to planned value.
  - $\text{SPI} = \frac{\text{EV}}{\text{PV}}$
  - If the SPI is greater than one, this means more work has been completed than the planned work. In other words, you are ahead of schedule.
  - If the SPI is less than one, this means less work has been completed than the planned work. In other words, you are behind schedule.
  - If the SPI is equal to one, this means all work is completed.
EVM – All Data Elements

- **BCWS**: Budgeted Cost of Work Scheduled
- **BCWP**: Budgeted Cost of Work Performed
- **ACWP**: Actual Cost of Work Performed
- **FCST**: Forecast of Remaining Work
- **BAC**: Budget at Completion
- **EAC**: Estimate at Completion
- **EAC**: Estimate to Go

Performance Measurement Data Elements:

- **Target Schedule**
- **Current Schedule**
- **Projection of Schedule Delay at Completion**
- **Projection of Cost Variance at Completion**
- **Reserve Funds Available**

Legend:

- **DATA DATE**
- **TIME**
- **COST (WH)**
- **TO-DATE SCHEDULE VARIANCE**
- **TO-DATE COST VARIANCE**

Graph showing the relationship between cost, time, and performance metrics.
Earned Value Management - Summary

- EVM integrates the areas of scope, schedule and cost in a single integrated system, Earned Value Management.
- The most important benefit of utilizing EVM in projects is to provide cost efficiency indications. Therefore EVM provides early warning of performance problems while there is time for corrective action.
- Validity of method – well known, PMI standard, in use for Project Management for a long time.
In 1991, Secretary of Defense Dick Cheney canceled the Navy A-12 Avenger II Program because of performance problems detected by EVM. This demonstrated conclusively that EVM mattered to secretary-level leadership.

Why is EVM important for Projects?
EVM in Traditional Projects - Summary

- Traditional Project Assumptions
  - Scope of the project is well understood, determined by work breakdown structure (WBS).
  - Project deliverables are well known.
  - There are three baselines – schedule, cost and scope baseline. The combination of all three baselines is referred to, as the performance measurement baseline.
  - After the initial iterative planning process, the planning baselines are subject of change control process.
  - If there is an approved change, the project is re-baselined.
EVM in Agile Software Projects

- Agile Software Project Assumptions
  - Project work is completed iteratively, where outcome from each previous iteration impacts the next one.
  - In Agile software project management, initial scope is not assumed to be complete. Scope is defined at a high level at the start of the project, and reiterated frequently over the course of the project.

- What is Agile EVM?
  - Agile EVM = Traditional EVM + Scrum
Why do we want to use Agile EVM?

- Benefit for Management:
  - Add a cost component to Agile metrics (increase visibility)
  - Optimize the value of releases
  - Make better business decisions

- Produce added value for the Business
  - Reduce product release time (allow faster marketing vs. competition)
  - Increase value and quality of the product
  - Reduce product costs
Scrum – Quick Review

**Product Roadmap** identifies the product functionality, and when features will be available.

**Product Backlog** is a prioritized features list, containing short descriptions of all functionality desired in the product.

**Release Plan** commits to a plan for delivering an increment of product value.

**Story points** represent the effort involved to deliver a product backlog item considering risk, uncertainty, and complexity.
Agile Software Project Tracking Options

Burn-Up Charts - shows the increasing amount of functionality accomplished as a function of time, and is reported on a regular basis.

This is conceptually equivalent to the Earned Value accumulated at a specific date.
Defining the Initial Release Baseline

What do we Need?

(1) Start Date
(2) Sprint length
(3) Number of Sprints
(4) Planned story points for the release

Release Date = Start Date + (# of Sprints) * Sprint length
# Standard vs. Scrum EVM Metrics

## Traditional Projects

- **PV** = Hourly Rate * Total Hours Scheduled
- **EV** = Baselined Cost * % Actually Complete
- **CPI** = EV / AC
- **SPI** = EV / PV
- **EAC** = BAC / CPI
- **ETC** = EAC – AC

## Agile EVM

- **PV** = Expected % Complete * BAC
  - Expected % Complete = Number of Sprints completed/Total Sprints planned
- **EV** = Actual % Complete * BAC
  - Actual % Complete = Story points completed/Total story points planned
- **CPI** = EV / AC
- **SPI** = EV / PV
- **EAC** = BAC / CPI
- **ETC** = EAC – AC
Measuring Progress in Agile (Release Level)

- Measuring Progress (calculated at each sprint)
  - What we are measuring in Agile EVM?
    - BAC (at release)
    - Planned vs. Completed Sprints
    - Story Points Planned vs. Completed or Added
    - Actual Cost to date

The CPI & SPI

<table>
<thead>
<tr>
<th>The CPI &amp; SPI</th>
<th>CPI &gt; 1</th>
<th>CPI = 1</th>
<th>CPI &lt; 1</th>
<th>SPI &gt; 1</th>
<th>SPI = 1</th>
<th>SPI &lt; 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>That Means</td>
<td>Under Budget</td>
<td>On Budget</td>
<td>Over Budget</td>
<td>Ahead of Schedule</td>
<td>On Schedule</td>
<td>Behind Schedule</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

That Means

- CPI > 1: Under Budget
- CPI = 1: On Budget
- CPI < 1: Over Budget
- SPI > 1: Ahead of Schedule
- SPI = 1: On Schedule
- SPI < 1: Behind Schedule
Measuring Progress in Agile (Release Level)

- Example
  - Budget at Release Completion = $250,000
  - Planned Sprints = 5 / Completed 2 Sprints
  - Planned Story Points = 1000 / Story Points Completed = 410
  - Actual Cost = $45,000 + $55,500 = $100,500

<table>
<thead>
<tr>
<th>Release 1.0</th>
<th>Sprint 1</th>
<th>Sprint 2</th>
<th>Sprint 3</th>
<th>Sprint 4</th>
<th>Sprint 5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAC</td>
<td>$50,000</td>
<td>$50,000</td>
<td>$50,000</td>
<td>$50,000</td>
<td>$50,000</td>
<td>$250,000</td>
</tr>
<tr>
<td>Story points planned</td>
<td>200</td>
<td>250</td>
<td>250</td>
<td>150</td>
<td>150</td>
<td>1,000</td>
</tr>
<tr>
<td>Story points done</td>
<td>210</td>
<td>200</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>AC</td>
<td>$45,000</td>
<td>$55,500</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>EV</td>
<td>$52,500</td>
<td>$102,500</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>PV</td>
<td>$50,000</td>
<td>$100,000</td>
<td>$150,000</td>
<td>$200,000</td>
<td>$250,000</td>
<td>$250,000</td>
</tr>
<tr>
<td>CPI</td>
<td>1.167</td>
<td>1.02</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>SPI</td>
<td>1.05</td>
<td>1.025</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>
Calculating Initial Release Baseline

- PV = Expected % Complete * BAC
  - Expected % Complete = The number of sprints completed divided by the number of sprints planned in the release. We have completed 2 of 5 planned sprints. Therefore 2/5 (%) = 40%
    - Expected Percent Complete = 40%, BAC = $250,000
    - PV = 0.4 * $250,000 = $100,000
- EV = Actual % Complete * BAC
  - Actual % Complete = Story points done/ Story points planned for the release. We have 1,000 story points planned in the release. At the end of Sprint 2 we have completed 410 story points. Therefore Actual % Complete = 410/1000 = 0.41 = 41%
    - Actual % Complete = 41%, BAC = $250,000
    - EV = 0.41 * $250,000 = $102,500
Summary of EVM for our Example

- Budget at Release Completion: BAC = $250,000
- Earned Value (after second sprint): EV = $102,500
- Planned Value (after second sprint): PV = $100,000
- Actual Cost (after second sprint): AC = $100,500
- CPI: CPI = 1.02 (102%)
- SPI: SPI = 1.025 (103%)

- EAC: Estimate at completion EAC = BAC/CPI = $245,098
- ETC: Estimate to complete, or expected additional cost needed to complete the project from the point of calculation ETC = EAC – AC = $144,598
How are we doing with our Agile Project?

- Current SPI  
  SPI = 1.025 (103%)

- Current CPI  
  CPI = 1.02  (102%)

- In this example, the SPI > 1 (1.025). It indicates that the project is ahead of schedule. It is a clear sign for the Scrum Master to accept more changes from the Product Owner since it will increase the profitability of the project.
Conclusion

- Simplified EVM approaches are relevant to both the agile world and the classic EVM world.

- Key points for agile teams:
  - You can, and often should, examine scope up front. For instance, you may populate your “Product Backlog” with an initial approximation of the full feature set.
  - You can, and often should, add “budget burn” (AC) to your burn chart. Doing so gives a simple, but effective, earned value chart that conveys considerably more information than a standard burn chart.
When Agile EVM Should be Utilized?

- Agile EVM is helpful when specific performance information (e.g. budget) on a fast moving project using the Scrum framework is needed.

- Example:
  - Problem:
    - The final release plan is fairly fluid. The Product Owner is identifying new features every week that should be included in the release.
  - Resolution:
    - Adapting the traditional earned value management metrics to the Scrum project will allow the Scrum Master to accurately predict the impact of these requirement changes, despite a fluctuating velocity, on budget and schedule for the release. By sharing information with the Product Owner and the team, it helps to make timely and accurate decisions.
Agile EVM at the Program Management Level

- Earned Value Management (EVM) metrics on large projects or programs can be measured across projects and integrated up into a single set of metrics reflecting progress against a plan at the program and/or release level.

- Example:

<table>
<thead>
<tr>
<th>Team</th>
<th>Budget</th>
<th>PV</th>
<th>EV</th>
<th>AC</th>
<th>CPI</th>
<th>SPI</th>
<th>EAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team A</td>
<td>$15,000</td>
<td>$7,000</td>
<td>$7,000</td>
<td>$7,000</td>
<td>1</td>
<td>1</td>
<td>$15,000</td>
</tr>
<tr>
<td>Team B</td>
<td>$40,000</td>
<td>$24,000</td>
<td>$20,000</td>
<td>$28,000</td>
<td>0.86</td>
<td>0.92</td>
<td>$41,250</td>
</tr>
<tr>
<td>Team C</td>
<td>$25,000</td>
<td>$9,000</td>
<td>$14,000</td>
<td>$12,000</td>
<td>1.11</td>
<td>1.16</td>
<td>$22,000</td>
</tr>
<tr>
<td>Program</td>
<td>$80,000</td>
<td>$40,000</td>
<td>$41,000</td>
<td>$47,000</td>
<td>0.99</td>
<td>1.03</td>
<td>$78,000</td>
</tr>
</tbody>
</table>
Summary

- In this presentation we highlighted following topics:
  - Traditional definition of EVM
  - Scrum Methodology – Quick Overview
  - Agile EVM
    - What is Agile EVM
    - Why we want to use Agile EVM?
    - What is needed to calculate?
    - How to analyze the results of Agile EVM
  - When Agile EVM should be applied – practical example
  - Agile EVM at the portfolio level
Questions and Answers

- Anything I missed?

- If yes, please send your questions, comments or suggestions to my LinkedIn profile or directly to my email address:
  - Ivan Papes
  ipapes@hotmail.com