How to Use Earned Schedule on Agile Projects

Name of Presenter: Robert Van De Velde, Ph.D., P.M.P., C.S.M.       Date: 22 May 2014
Contact: Robert.VanDeVelde@ProjectFlightDeck.com
Context

Over the past decade, Earned Schedule has been elaborated, empirically verified, and adopted by diverse industries.

During the same period, the Agile framework has become widely used, especially in engineering and technology projects.

Earned Schedule for Agile projects (AgileES) combines the speed and responsiveness of Agile with the accuracy and insight of Earned Schedule.

Here is the road map for today’s session.
Basic Concepts

Agile (Scrum) Definitions

Product Backlog
  Prioritized list of requirements

Release Point
  Numerical value of work required to produce a backlog item

Velocity
  Planned productivity per sprint

Sprint
  Basic unit of Agile delivery, usually 1-4 weeks in length

Release Plan
  Roadmap of sprints that achieve project goal
Earned Schedule Definitions

Planned Value (PV, BCWS)
Value of work planned

Earned Value (EV, BCWP)
Value of work completed

Earned Schedule (ES)
“The amount of time earned on a project is the time at which the value currently earned should have been earned.” (Lipke, *Measurable News*, 2003)
Basic Concepts

Earned Schedule Calculation

Given that
- 6 sprints, 2 weeks per sprint
- $10K of Planned Value per sprint
- $45K of Earned Value at Actual Time (AT)

Calculate amount of Earned Schedule
- EV at Actual Time versus PV at end of each sprint
- Count the sprints in which the EV_{AT} > PV_i
- Add in any fractional amount \((\$50K - \$45K)/(\$60K - \$50K) = .5\)
- Result is 4.5 periods (9 weeks) of Earned Schedule

\[
ES = \sum_{i=1}^{4} (EV_{AT} \geq PV_i) + \frac{EV_{AT} - PV_i}{PV_{i+1} - PV_i}
\]

<table>
<thead>
<tr>
<th>Sprint</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>Equation</th>
<th>ES</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV_i (K$)</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>60</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EV_{AT} (K$)</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td></td>
<td>4.5</td>
</tr>
</tbody>
</table>

College of Performance Management
Agile Metrics

Burndown Chart
Shows the number of release points remaining and, thus, the number finished

Ideal Burndown Line
Shows the number of release points that should have been completed from the first sprint to the last planned sprint, given the mean velocity

Schedule Status
Burndown above Ideal = late and Burndown below Ideal = early

![Test Project Burndown and Ideal Burndown Graph]

- **Release Points (RP)**: 6000, 5000, 4000, 3000, 2000, 1000, 0
- **Sprints**: 1 to 10
- **Index Value**: 1.00, 0.80, 0.60, 0.40, 0.20, 0.00
- **Test Project**: Burndown and Ideal Burndown
- **Start**: Release Point Start
- **Planned Finish**:
Earned Schedule Metrics

Schedule Performance Index for time ($SPI_t$)
The ratio of the time earned to the actual time, i.e., the efficiency of time utilization on the project (Lipke, Measurable News, 2003)

Estimate at Completion for time ($EAC_t$)
Unlike standard EVM, ES offers an estimate at completion for time, i.e., the estimated duration of the project given the ratio between the Planned Duration and the $SPI_t$ (Henderson, Measurable News, 2004)
Schedule Performance Efficiency Example

Burndown shows steady decline
Initially, Burndown on or slightly above the Ideal line = on or slightly behind schedule
Then, Burndown jumps above the Ideal line = definitely behind schedule.

SPIₜ clarifies what was happening
Initially, schedule performance efficiency improves but then declines after Sprint 4
Root cause analysis of the delay should start at Sprint 4, rather than later
What ES Adds to Agile

Schedule Performance Efficiency Explanation

Burndown shows relationship between elapsed time and Release Point completion

SPI_t shows relationship between elapsed time and earned time

SPI_t quantifies how well or poorly time is being used on the project

Threshold values mark boundaries between project status

<table>
<thead>
<tr>
<th>Sprints</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI_t</td>
<td>.63</td>
<td>.92</td>
<td>.94</td>
<td>.96</td>
<td>.88</td>
<td>.87</td>
</tr>
</tbody>
</table>
SPI Correction

Schedule Performance Index (SPI)
Standard Earned Value metric for schedule performance equals the ratio between Earned Value and Planned Value

SPI Problem
At the end of a project, the ratio equals 1—even late projects end with a perfect SPI

AgileES Solution
SPI\textsubscript{t} accurately reflects performance throughout project life cycle

![Graph showing SPI\textsubscript{t} and SPI](image)
From Theory to Practice

Show how Earned Schedule is used to manage Agile projects

Components commonly associated with schedule performance management on plan-driven projects
  - Baseline schedule
  - Schedule performance data
  - Schedule performance metrics
  - Threshold values
  - Re-baselining

Same components are used on Agile projects but with a uniquely Agile interpretation

Agenda

Opening Thoughts
  - Context
Basic Concepts
  - Definitions
  - Metrics
What ES Adds to Agile
  - Schedule Performance Efficiency
  - SPI Correction
Using AgileES
  - Set the Baseline Schedule
  - Capture the Data
  - Calculate the Metrics
  - Analyze and Respond to Results
How to Re-baseline the Schedule
After a Schedule Re-baseline
Future Directions
  - Schedule Burndown
  - AgileEVM + AgileES
Closing Thoughts
Summary
Set the Baseline Schedule

From number of Release Points, use velocity to set total Sprints and Planned Finish
From number of Release Points and Planned Finish, set the Ideal Burndown line
From risk analysis, determine the contingency for Release Points and timeline
Add product owner reserve (aka, management reserve)
Capture the Data

AgileES uses the same data that is ordinarily collected on Agile projects, minimizing the overhead for Agile teams

Count Release Points that have been completed

Add new Release Points not already included in the total

Remove Release Points that are no longer required from the total

<table>
<thead>
<tr>
<th>Start Date</th>
<th>1/14/2013</th>
<th>2/25/2013</th>
<th>As of: Base 2, Sp 6</th>
<th>Baseline 1</th>
<th>Baseline 2</th>
<th>Base2 Tot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint Length</td>
<td>2 weeks</td>
<td>2 weeks</td>
<td>2 weeks</td>
<td>2 weeks</td>
<td>2 weeks</td>
<td>2 weeks</td>
</tr>
<tr>
<td>Actual cost</td>
<td>$450,000</td>
<td></td>
<td></td>
<td></td>
<td>$450,000</td>
<td></td>
</tr>
<tr>
<td>Sprint</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>Mean Planned Rel Pts</td>
<td>352</td>
<td>352</td>
<td>352</td>
<td>780</td>
<td>780</td>
<td>780</td>
</tr>
<tr>
<td>Release Pts Completed</td>
<td>88</td>
<td>80</td>
<td>128</td>
<td>424</td>
<td>848</td>
<td>0</td>
</tr>
<tr>
<td>% Complete</td>
<td>0.03</td>
<td>0.05</td>
<td>0.08</td>
<td>0.07</td>
<td>0.20</td>
<td>0.65</td>
</tr>
<tr>
<td>Earned Value</td>
<td>8,800</td>
<td>8,000</td>
<td>16,800</td>
<td>42,400</td>
<td>127,200</td>
<td>408,100</td>
</tr>
<tr>
<td>Earned Schedule</td>
<td>5.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPIt</td>
<td>0.87</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EACt</td>
<td>9.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EACt/AT</td>
<td>1.53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release Date</td>
<td>7/15/2013</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: 9zzz
Calculate the Metrics

Ratio of current total of Planned Release Points to baseline total

Earned Schedule → Schedule Performance Index for time

Ratio of current estimated Release Date to baseline Release Date
Analyze and Respond to Results

Thresholds
Total of Release Points varies from contingency + reserve
SPI<sub>t</sub> less than .8
Estimated Release Date varies from contingency + reserve

Is a new baseline schedule required?
No: within thresholds
Yes: 3 consecutive threshold breaches in same, worsening direction
(Kesheh and Stratton, Measurable News, 2014)

Test Project
Burndown, SPI<sub>t</sub>, and Ideal Burndown

<table>
<thead>
<tr>
<th>Release Points</th>
<th>Index Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>6000</td>
<td>1.00</td>
</tr>
<tr>
<td>5000</td>
<td>0.80</td>
</tr>
<tr>
<td>4000</td>
<td>0.60</td>
</tr>
<tr>
<td>3000</td>
<td>0.40</td>
</tr>
<tr>
<td>2000</td>
<td>0.20</td>
</tr>
<tr>
<td>1000</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sprints</th>
<th>Release Points: @ 64% vs 16%</th>
<th>SPI&lt;sub&gt;t&lt;/sub&gt;: .24</th>
<th>Release Date: @310% vs 16%</th>
</tr>
</thead>
</table>
How to Re-baseline the Schedule

New baseline, new project
Freeze completed sprints
Move unfinished Release Points to new sprint(s) and re-set total and velocity
Re-set the Start Date, Planned Finish, and the Ideal Burndown
Re-set sprint numbers and total sprints
Re-set contingency and reserve
After a Schedule Re-baseline

Return to normal data capture and analysis and response activities

Release contingency without a re-baseline if within thresholds

Release of owner reserve often requires re-baseline

Importance of estimated Release Date increases late in project

**Test Project**
Burndown, SPI, and Ideal Burndown

<table>
<thead>
<tr>
<th>Sprint 6</th>
<th>Release Points: @13 vs 16%</th>
<th>SPI: .87 vs 15 vs 16%</th>
<th>Release Date: @15 vs 16%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sprint 8</td>
<td>Release Points: @12 vs 16%</td>
<td>SPI: .88 vs 14 vs 16%</td>
<td>Release Date: @14 vs 16%</td>
</tr>
<tr>
<td>Sprint 10</td>
<td>Release Points: @12 vs 16%</td>
<td>SPI: .80 vs 16 vs 16%</td>
<td>Release Date: @16 vs 16%</td>
</tr>
</tbody>
</table>

Index Value

0.00 0.20 0.40 0.60 0.80 1.00

Burndown SPI Ideal Burndown

Re-baseline

Planned Finish

College of Performance Management
New Approaches

Earned Schedule is a robust concept that offers new approaches for Agile projects.

Earned Schedule makes possible the Schedule Burndown, a whole new metric for Agile projects.

There is also an opportunity to integrate Earned Schedule into EVM practices currently being applied to Agile projects for cost management.
Schedule Burndown

Time earned vs. ideal time line rather than points completed vs. ideal velocity line

The Ideal Schedule Burndown line runs from first to last planned sprint

ES Burndown decrements sprint total by amount of Earned Schedule per sprint

ES Burndown above Ideal line = late and ES Burndown below Ideal line = early

Plot SPI_t on same chart for more complete picture of schedule performance
AgileEVM + AgileES

AgileEVM successfully applies EVM to Agile projects for cost management.

AgileES corrects the problem with SPI and offers improved schedule management.

Integration of the two offers Agile teams a powerful tool for enhancing their project management.
Summary

AgileES combines Earned Schedule and Agile project management

AgileES adds value to familiar Agile schedule management techniques

To implement AgileES on your project, you need a baseline, actuals, metrics, and know when and how to re-baseline

AgileES offers a springboard to further innovation in the future
AgileESM© Calculator

Here are the steps for downloading the Excel™ Calculator:

1. Go to www.AgileESM.com
2. Select Members tab
3. Enter Username: CPM1
   Password: CPM1
4. Download Calculator—instructions for use are included in the file

Contact: Robert.VanDeVelde@ProjectFlightDeck.com
References


