Agile Project Performance Management
(a.k.a. “Agile Metrics”)

Kevin Aguanno, PMP, MAPM
What will we cover?

• How traditional planning & metrics fail in high-change projects
• What is agile project management?
• Benefits of agile approaches
• Agile performance metrics
• Agile adoption statistics
Traditional Planning Shortcomings

- We plan by task not by feature
- Tasks rarely finish early (Parkinson’s Law)
- Delays cascade through the schedule
- We tend to ignore uncertainty
- Estimates become commitments
Uncertainty Diminishes Over Time

![Graph showing the reduction in uncertainty over time during different stages of product development.](chart.png)
And we rarely prioritize features

Feature Usage Within Deployed Applications

- Always: 7%
- Often: 13%
- Sometimes: 16%
- Rarely: 19%
- Never: 45%

Source: Chaos Report v3
Multi-tasking causes further delays

Effects of Multi-Tasking on Productivity

© 2010 Kevin Aguanno. All rights reserved.
So we Search for New Methods

- < 1970: Iterative Incremental Development (IID)
- 1970: Waterfall Model
- 1975: Iterative Enhancement
- 1988: Spiral Development Model
- 1990s: Crystal Methods, Lean, DSDM
- 1993: Capability Maturity Model
- 1996: Scrum
- 1998: Extreme Programming (XP)
- 2001: Agile Manifesto
- 2002: Agile Modeling
What is Agile?

Agile Delivery is Iterative AND Incremental
Agile Methods Reduce Risk

- Risk of building the wrong thing
  - “what I asked for, but not what I meant/wanted”
  - Being responsive to changing requirements

- Risk of poor quality
  - More time to correct defects

- Risk of getting stuck in “design churn”
  - Build what you know first
Agile Methods Improve Control

- Frequent delivery of working code means progress is objectively measurable
- More chances for sponsor/stakeholders to provide early feedback to redirect project priorities where necessary
- Misunderstandings are surfaced earlier
- Can end project early and still get measurable benefits
Agile Methods Improve Communications

- Co-located teams
  - Whiteboarding
  - Ability to look over each other’s shoulders to share ideas
  - Full communications (incl. 70% nonverbal)

- Daily meetings
  - Knowledge transfer and collaboration

- Daily customer/sponsor involvement
  - Quick and easy escalation
  - Snap decision-making
  - Instant feedback
Risk Benefits

In summary, it is all about reducing RISK:
- Risk of building the wrong thing (requirements)
- Risk of building a shoddy product (testing)
- Risk of budget and/or schedule overruns

Project success helps build TRUST
- Success in early iterations can have the same effect well before the project is complete
Agile Project Metrics
Why use status reports and metrics?

- Build Trust – Where trust does not exist.
- Measure Risk Level – Mostly budget and schedule risk.
- Communicate Progress
- Communicate Issues Requiring Assistance
- Assess Impact of Process Improvements
As Predictability Varies, so should Metrics

Deterministic (Plan-Based)
* LOW CHANGE *

Empirical (Observation-Based)
* HIGH CHANGE *

Focus is on adherence to plan

Focus is on delivering value to customer

Linear  Incremental  Iterative  Agile
Problems with Linear (Waterfall) Tracking Techniques

- % Complete can mean many different things:
  - % estimated work effort complete.
  - % work products complete.
  - % tasks complete.
  - % customer-requested functionality/content delivered.
  - % budget used.
  - % schedule used.
  - % that I think you will accept as reasonable.
“Best Practices” Try to Solve this Problem with EVM

- Earned Value Management System = A method for tracking project progress and efficiency against schedule and budget targets.

- Using efficiency scores, EVM users can forecast completion costs fairly accurately.

- A “true” EVM implementation has 30+ criteria to fulfill and is quite challenging to set up and manage.

- Many see it as bureaucratic and cumbersome.
EVM Example – Schedule Performance

- SV = BCWP - BCWS
- SPI = BCWP / BCWS
- SV% = SV / BCWS

Cost ($)

Time

$30k

$40k

SV = -$10k
SPI = 0.75
SV% = -25%
(Behind Schedule)
But EVM is not Perfect…

- While EVM is very good at:
  - Building Trust
  - Measuring Budget and Schedule Risk Level
  - Communicate Progress

- It also has some weaknesses:
  - You can work ahead on easy tasks, ignoring important tasks, and these metrics won’t catch that.
  - EVM metrics do not warn you that you are building the wrong thing.
  - EVM loses most of its benefits when your baseline keeps shifting (high change projects).
The 90% Complete Syndrome

- A weakness of Linear Method metrics (including EVM) when encountering high-change situations.

- Usually occurs in design phase (25% complete syndrome) or testing phase (90% complete syndrome).

- Our tools (MS Project, etc.) typically cannot adapt to this reality.
Challenges with “Traditional” Metrics

- Most tools do not permit conditional branching
- Difficult to use EVA when BCWS baseline changes with each iteration
- How do you calculate % Complete or other progress indicators?
Incremental & Agile Methods Use Velocity

- # Features Completed at the end of each iteration versus the # Features Planned at that point (helps forecast team performance).

- Can also use Function Points, Use Cases, or another substitute for # Features. This allows for consistent metrics even when content of future increments has been changed.
Velocity (a refresher)

- A measure of the team’s rate of progress. Calculated by summing up the number of story points or ideal days that the team completed in its most recent iteration.
- Can also use function points or some other metric that includes a measure of complexity.
- If all features are about the same in their level of complexity (e.g. writing athlete bios for the entire Canadian olympic team), then you can use # features completed for your velocity.
Using Velocity Metrics to Manage a Schedule

- Velocity can also be measured as a percentage (\# story points completed / \# story points planned). This gives you a metric for your estimating accuracy and efficiency.
- Velocity % can be used to adjust the forecast schedule across multiple future iterations.
- Is used very similarly to EVM’s Schedule Performance Index metric.
Velocity is Easy to Calculate

- For each iteration…
  \[ \text{Velocity} = \frac{\# \text{ Features completed}}{\# \text{ Features Planned}} \]

- For an iteration (n)…
  \[ \# \text{ Features Planned} (n) = \text{Estimated} \# \text{ Features} \times \text{Velocity(n-1)} \]

- First iteration requires special handling
- Velocity self-corrects over time
## Earned Value vs. Velocity

<table>
<thead>
<tr>
<th>Measures efficiency &amp; estimating accuracy (CPI or SPI)</th>
<th>Measures efficiency &amp; estimating accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used to adjust estimates to complete</td>
<td>Used to adjust estimates to complete</td>
</tr>
<tr>
<td>New Forecast = ( \frac{ETC($)}{CPI} ) (or SPI)</td>
<td>New Forecast = ( \frac{\text{Est. Cost}}{\text{Velocity}} ) (Can use Est. # Iterations)</td>
</tr>
</tbody>
</table>

© 2010 Kevin Aguanno. All rights reserved.
By summing up the total # story points remaining and updating this at the end of each iteration, we can also use velocity (represented as the slope of the line on the graph) to forecast a product release date / project completion date.
Mid-Iteration Performance Tracking

Within an iteration, we track \#hours or \#days remaining to complete each task and sum them up into a total estimate to complete.

### Book Writing Project

**Iteration 4 (Theme: "Write the chapter on agile testing")**

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task Name</th>
<th>Assigned</th>
<th>Original Effort</th>
<th>Effort Remaining</th>
<th>Story #</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Prepare chapter outline</td>
<td>Kevin</td>
<td>3</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>4.2</td>
<td>Gather editorial feedback</td>
<td>Kevin</td>
<td>5</td>
<td>0</td>
<td>22</td>
</tr>
<tr>
<td>4.3</td>
<td>Write chapter body</td>
<td>Kevin</td>
<td>65</td>
<td>25</td>
<td>22</td>
</tr>
<tr>
<td>4.4</td>
<td>Copy editing</td>
<td>Francis</td>
<td>10</td>
<td>10</td>
<td>22</td>
</tr>
<tr>
<td>4.5</td>
<td>Prepare illustrations</td>
<td>Carrie</td>
<td>2</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>4.6</td>
<td>Index chapter</td>
<td>Tony P.</td>
<td>1</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
<td></td>
<td>86</td>
<td>37</td>
<td></td>
</tr>
</tbody>
</table>

© 2010 Kevin Aguanno. All rights reserved.
Iteration Burndown Charts

- We plot this total on a chart periodically (every day is ideal) and use the slope of the line to forecast a completion date.
Project Budget Burndown Charts

- Similar to other burndown charts, you can use this chart to track your forecasted budget use across multiple iterations. You can use your “budget velocity” (slope) as a rough means of forecasting budget at completion.

- It is sometimes useful to plot the budget burndown and the project burndown on the same graph for comparison purposes.
Iteration Budget Burndown Tracking

Within an iteration, we track the updated effort to complete on a daily basis, all we need to do is to add an hourly cost rate column beside each task to forecast and track our costs within the iteration.

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task Name</th>
<th>Assigned</th>
<th>Cost per Hour</th>
<th>Original Effort</th>
<th>Effort Remaining</th>
<th>Original Budget</th>
<th>Budget Remaining</th>
<th>Story #</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Prepare chapter outline</td>
<td>Kevin</td>
<td>$100</td>
<td>3</td>
<td>0</td>
<td>$300</td>
<td>$0</td>
<td>22</td>
</tr>
<tr>
<td>4.2</td>
<td>Gather editorial feedback</td>
<td>Kevin</td>
<td>$100</td>
<td>5</td>
<td>0</td>
<td>$500</td>
<td>$0</td>
<td>22</td>
</tr>
<tr>
<td>4.3</td>
<td>Write chapter body</td>
<td>Kevin</td>
<td>$100</td>
<td>65</td>
<td>25</td>
<td>$6,500</td>
<td>$2,500</td>
<td>22</td>
</tr>
<tr>
<td>4.4</td>
<td>Copy editing</td>
<td>Francis</td>
<td>$85</td>
<td>10</td>
<td>10</td>
<td>$850</td>
<td>$850</td>
<td>22</td>
</tr>
<tr>
<td>4.5</td>
<td>Prepare illustrations</td>
<td>Carrie</td>
<td>$70</td>
<td>2</td>
<td>1</td>
<td>$140</td>
<td>$70</td>
<td>22</td>
</tr>
<tr>
<td>4.6</td>
<td>Index chapter</td>
<td>Tony P.</td>
<td>$85</td>
<td>1</td>
<td>1</td>
<td>$85</td>
<td>$85</td>
<td>7</td>
</tr>
</tbody>
</table>

TOTALS: 86 37 $8,375 $3,505

© 2010 Kevin Aguanno. All rights reserved.
Quality Metrics are More Rigidly Tracked

- Same metrics we use today:
  - Defect injection rate per iteration
  - # Outstanding defects sorted by severity
  - Etc.

- Short iterations demand automated testing tools to perform regression tests and gather, track, and report on results metrics
- Frequent builds demand more focus on quality
Other reporting methods exist

- Feature Driven Development uses domain feature progress charts.

Colour is status/risk (green/yellow/red)

**Reporting**
- 27 User Stories
- 75% Complete

**Editorial Interface**
- 15 User Stories
- 20% Complete
Agile Adoption Statistics
Agile Adopters Usually Meet with Strong Resistance

Kuhn coined “Paradigm Shift” to describe the impact on thinking caused by new scientific approaches.

Traditionalists:
- Marginalize proponents
- Express disbelief
- May express anger
- Eventually resign themselves to the new thinking as it gains critical momentum

By Thomas S. Kuhn
Written in 1962

The Structure of Scientific Revolutions
Third Edition

© 2010 Kevin Aguanno. All rights reserved.
That Critical Momentum is Here

Has Your Organization Adopted One or More Agile Techniques?

Source: 2007 Agile Adoption Survey, *Dr. Dobbs Journal*
That Critical Momentum is Here

Of the 241 Who Had Not Yet Adopted Agile, When Will Will They?

Source: 2007 Agile Adoption Survey, Dr. Dobbs Journal
Agile Improves Overall Success Rates

<table>
<thead>
<tr>
<th></th>
<th>Project Success Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile</td>
<td>71.5</td>
</tr>
<tr>
<td>Traditional</td>
<td>62.84</td>
</tr>
<tr>
<td>Data Warehouse</td>
<td>62.59</td>
</tr>
<tr>
<td>Offshoring</td>
<td>42.68</td>
</tr>
</tbody>
</table>

Source: 2007 Project Success Survey, *Dr. Dobbs Journal*
Additional Resources on Using Agile Techniques

- Sign up for the free monthly AgilePM Newsletter: www.AgilePM.com
- Kevin Aguanno (your speaker) is available via aguanno@ca.ibm.com. He is the author of several books, audiobooks, and CD-ROMs related to this subject matter: