Agile Metrics - What You Need to, Want to, and Can Measure

June 9, 2014
Agile Metrics Outline

- What do you care about?
- Why measure?
- Metrics to introduce earlier
- Metrics to introduce later
- How to prepare
- Adoption metrics
- Examples
- Challenges to agile metrics
- Best practices for agile metrics
- What the future holds for agile metrics
Agile Metrics – What Do YOU Care About?

• Exercise … let’s take 5 minutes now

• About You:
  • Your role relative to agile delivery
  • Your home organization (down to department)
  • Your years of IT experience
  • Your years of agile practitioner experience

• Metrics Perspective
  • What performance metrics do you care about the most (up to 3)
  • For each, why (as specific an answer as possible, please)
  • Are you a metrics data producer, consumer, neither, or both?

• Thank you!
Agile Metrics – Why Measure?

Are we on track?
- For planned benefits
- For planned delivery dates
- Relative to my budget
- With Agile adoption

Are you delivering value for me?

Is the work good quality?

Where are the problems we need to address?

Are we improving?

Are we working on the right things?

How do the participants and stakeholders feel about it?
Agile Metrics - Who Cares?

- **Leaders (Business and Technology)**
- **Support Organizations**
- **Program Sponsors**
- **Product Owners**
- **Coaches**
- **Scrum Masters and Team Leads**
- **Team Members**
Agile Metrics - Bottom Line Why Measure and Report?

- Answer stakeholder questions
- Provide transparency
- Manage the work
Agile Metrics – Start Sprint 1

Quality
• Story documentation quality
  • Number of times stories are rejected by delivery team
  • Number of RCAs traced to story quality
• Defect density (defects delivered per story point)

Cost and Speed
• Stories ready for consumption
• Team velocity or productivity (stories delivered per sprint)
• Work in process (number of items being worked on simultaneously)

Remember, your teams are on a learning curve!
Agile Metrics – Introduce in Later Sprints

Predictability
- Estimation variance

Quality and Value
- Prioritized backlog
- Business value delivered (accepted work)
- User satisfaction (survey)

Cost and Speed
- Team turnover
Agile Metrics

It doesn’t happen by itself!!!
Agile Metrics - Structure **Before** Teams Start

- Sprint frequency
- Release cadence (and process)
- Planning cycle cadence (and process)
- Epic and story documentation templates and guidance (tool and process)
- Initial backlog (tool)
- Documented high-level architecture choices for high-risk technologies
- Stakeholder management plan, including status reporting, planned metrics, and data capture plan
- Baseline measures for planned metrics, if possible
Agile Metrics – Consider How Your Engagement will Handle this during Planning?
What is the most-reported Agile metric?
Daily Team Burn Down Example

- Day 1 spike
- Still a problem by Day 3
- What’s going on?

- Defects fed to team?
- New technology?
- Story changes?
- Vacations?
- New team members?
Team Y added XX scope and re-baselined work projections based on Velocity to date.
Story Readiness and Estimation Variance Example

• As of MM/DD/YYYY, program-level Story Point Variance (including Estimate Variance, Scope Additions / Reductions, UI Rework) stands at 4%.

• Major features contributing to this are:
  • Feature 1
  • Feature 2
  • Etc.

Planning for next sprint team capacity, where are the stories?

<table>
<thead>
<tr>
<th></th>
<th>Admin</th>
<th>Platform (Core)</th>
<th>Athena</th>
<th>Zeus</th>
<th>Apollo</th>
<th>Hercules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story Points</td>
<td>57</td>
<td>11</td>
<td>12</td>
<td>5</td>
<td>47</td>
<td>30</td>
</tr>
<tr>
<td>What is Available</td>
<td>77</td>
<td>22</td>
<td>55</td>
<td>63</td>
<td>81</td>
<td>75</td>
</tr>
<tr>
<td>What is Needed</td>
<td>57</td>
<td>11</td>
<td>12</td>
<td>5</td>
<td>47</td>
<td>30</td>
</tr>
<tr>
<td>Business days to go</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>
Groomed Backlog and Business Value Delivered

Example

80% of value typically resides in 20% of features

65% of features provide little to no value, are rarely used and/or aren’t actually desired by the customer

The rest are OK, but not as important
Quality Metrics Example - What’s Happening with Defects?

2 of 11 agile teams have high numbers of open defects

Many open defects ready for QA and UAT (downstream teams)
Agile Metrics - Interactions, Quality vs. Work in Process

“Teams with the lowest WIP have 4 times better Quality than teams with the highest WIP.”

Agile Metrics - What to do with Work in Process findings? “It depends!”

Do you care more about productivity or time to market?

If time to market is your most important driver, push Work in Process as low as possible.

If productivity is your most important driver (number of stories and defects completed per time period), don’t focus solely on Work in Process.

Agile Metrics - What to do with Team Turnover findings? Fix it if you can!

Teams with low turnover are more stable. Stable teams are more productive and more predictable (lower variance sprint-over-sprint).

Agile Metrics – Adoption Team-Level Assessment

Nascent Teams
FOLLOW the Rules

Proficient Teams
BREAK the Rules

Hyper Performing Teams
BE the Rule

守-Shu (Stage 1)
破-Ha (Stage 2)
離-Ri (Stage 3)
## 12 Agile Adoption Quantitative and Qualitative Metrics

<table>
<thead>
<tr>
<th>Quantitative</th>
<th>Qualitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Velocity</td>
<td>Sprint Planning</td>
</tr>
<tr>
<td>Quality</td>
<td>Release Planning</td>
</tr>
<tr>
<td>Story Creation</td>
<td>Daily Scrum</td>
</tr>
<tr>
<td>Accuracy of Commitment</td>
<td>Story Estimation</td>
</tr>
<tr>
<td>Accuracy of Estimates</td>
<td>Release Frequency</td>
</tr>
<tr>
<td>Overtime</td>
<td>Retrospectives</td>
</tr>
</tbody>
</table>
Using Agile Adoption Shu Ha Ri Metrics

- For agile adoption programs
- Leverage tools
- Stage 2 (Ha) is a great achievement!
- Stage 3 (Ri) is rare
- Focus on trends over time
- Use to manage the adoption program as it scales

http://www.slideshare.net/cognizant/shuhari-measuring-agile-adoption-maturity
Agile Metrics - Must-Haves for Scrum Masters and Agile Program Managers

- Burn-down charts by team, release, program
- Defect density
- Velocity
- Sprint over sprint improvement in velocity
- Team stability
- Speed of grooming backlog
- Stability of backlog growth and inflation (scope and effort)
Agile Metrics – What Not to Measure

• People push back on per individual metrics (only down to team)

• Hard to collect, low utility metrics
Agile Metrics – Challenges

• No tool support
• No summary
• Audience education missing
• Demand for traditional metrics reporting – accept it!
  • RAG indicators for schedule, cost, quality
Agile Metrics – Best Practices

Planning for Metrics
• How will they be used?
• How will they be collected and reported?
• Are baselines in place?

Cognizant’s Provably Better Programs

Cognizant’s Shu-Ha-Ri Adoption Measurement Approach

Keep it Simple

Meaningful Metrics
Agile Metrics – Looking Forward

• Agile community: fewer metrics
• Rest of world: more metrics
• Mobile
• Self-serve
• Business-outcome focused
Thanks to Dan Fuller for his work on *Shu Ha Ri* Agile Adoption Metrics.

Thanks to our partner *Rally Software* for permission to use their metrics findings based on over 9600 teams using their ALM to deliver agile software.
Agile Metrics - What You Need to, Want to, and Can Measure

June 9, 2014