

Sample Agile Assessment For Acme Corp

Prepared by:
AccuRev



Introduction

Over the last two days, Agile Evangelists Jack X. and Jill Y. from AccuRev had the pleasure of working with the development, QA and management teams at Acme. The team did a great job describing their Agile processes, and areas for improvement. While we agreed that there were many things they were doing well, the team did identify some areas they thought needed to be improved upon.

Areas for Improvement

Getting to done

During some retrospectives, product owners noticed that some of the user stories could not be included in the demo for their clients. Some of the user stories were only half completed, and more often coded than tested. Jack X. expressed concern that they were not delivering value to clients that they had promised at the beginning of the iteration.

Improve Delivery Rate

We heard several concerns from members of the management team, as well as from the development teams, about the rate at which software was released. Many people pointed out that although they were doing iterative development, software was not reaching the customer as soon as it should. The team believes we need to find ways to release software faster after each iteration is complete.

Create Metrics

The teams are looking for ways to improve over time, and are looking for ways to measure improvement ideas, especially across development teams. Concern was also raised that the metrics shouldn't distract from the key driver of delivering customer value. There are some recommendations in this proposal for good ways to use metrics to monitor the process which can be used process improvement in the future.

Agile Success Workshop Process

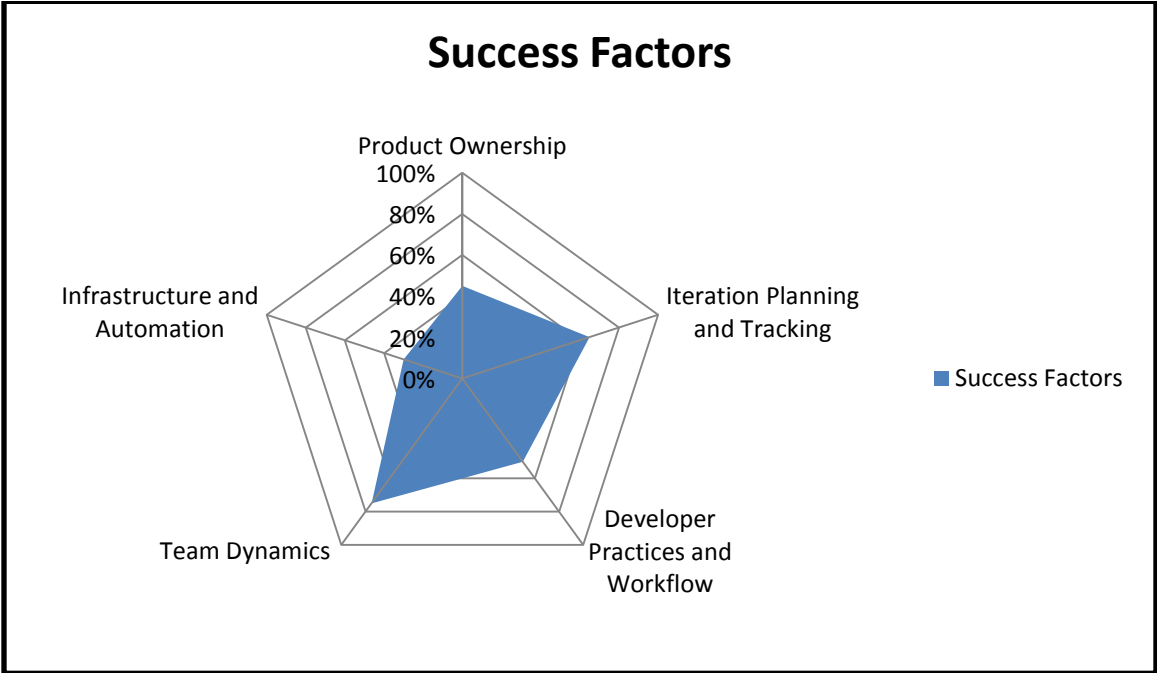
Our consultants worked with the team to understand and analyze the Acme development processes, and then compare them with the best practices used in the industry as a whole. Five Agile success factors were chosen to best represent the current opportunities and challenges for Acme, and for how they relate to the organization as a whole:

1. Product Ownership
2. Iteration Planning and Tracking
3. Developer Practices and Workflow
4. Team Dynamics
5. Infrastructure and Automation

Within each factor for success, the management team graded itself on how much room there was for improvement. This grade was also subjectively modified based on Acme's process relative to industry best practices.

The recommendations in each area are designed to be testable. Metrics have been identified to show self-improvement in each area. One of the goals of this action plan is to help, over time, the organization adapt and improve, both with Agile and, even more importantly, as a development organization.

Assessment Results



Part 1: Product Ownership 45%

Acme had a product owner but needed to improve in several areas

- Regular grooming of the backlog
- Define the acceptance criteria for stories so the development team can understand when the user story is “done”

In some cases, there seemed to be a lack of visibility into what the “big picture” was for the product vision. It’s essential that a product owner can articulate that vision to the rest of the team so that the resulting product can be fully featured and developers know the common goal.

Product Ownership Action Plan

- Make sure user stories were well written in typical “user story” form
- Product owner should participate in planning sessions and product demos
- Product owner can manage different inputs from sales, marketing and other stakeholders well
- Plan a grooming meeting where the backlog can be prioritized



Target: Product Ownership to 65%

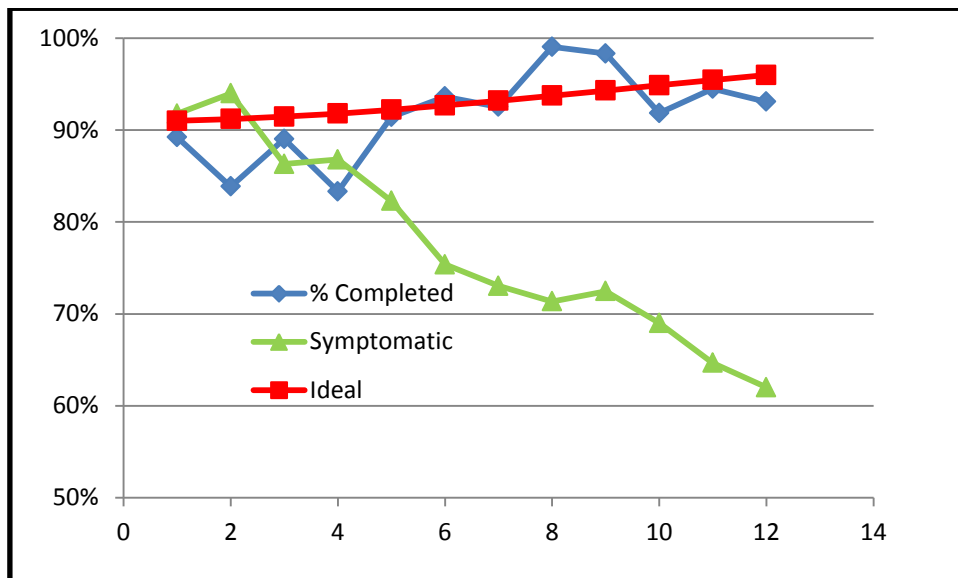
Part 2: Iteration Planning and Tracking 65%

Impressions from the interview suggested that Acme struggled with sizing and planning user stories for several iterations. Multiple times teams were not able to complete a large amount of tasks and stories planned for that sprint.

Acme suffered from inconsistent planning meetings. In some cases, the meetings occurred without all of the necessary members present. In other cases, the planning meeting hadn't happened at all and the product owner sized the stories himself.

To improve on this, we suggest collecting the following metric from your Agile Project Management (APM) tool.

Number of Stories Completed Per Iteration Vs Plan



This metric should trend up and stabilize for the team to correct itself. To improve this trend, we suggest these actions:

Iteration Planning and Tracking Action Plan

- Make sure user stories are sized correctly
- Include all members of the scrum team in the planning session to build consensus on story points
- Examine if impediments are being cleared by the Scrummaster
- Look for automation techniques such as automated testing, automatic merging and continuous integration / deployment



Target: Iteration Planning and Tracking to 75%

NOTE: After these improvements are made, look for the team to become more adept at estimating stories however, there should be some room for variance in the number of stories completed during the iteration; it will not be an exact science. The trend should remain stable within a small range.

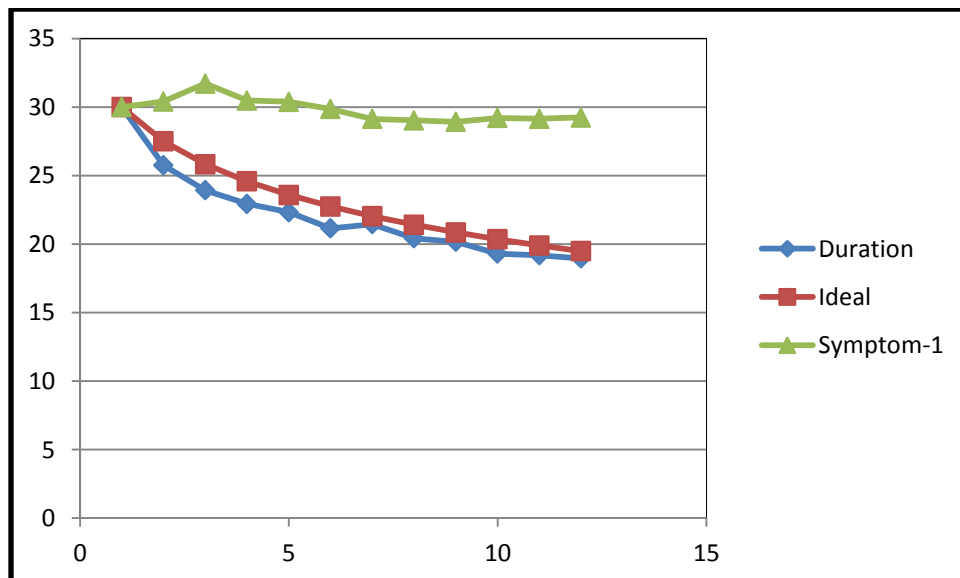
Part 3: Team Dynamics 60%

Team Dynamics play one of the most important rolls when it comes to having a high functioning Agile implementation. We interviewed several members of Acme's scrum teams and found that the team was successfully communicating with each other, but we did find that feedback was not getting to some of the appropriate members of the team in order to receive rapid feedback and remove impediments.

Some of the team members mentioned that the standup meetings were running around 45 minutes in length. We believe that some of the team members were getting bogged down by giving status updates to the team rather than participating in feedback for the other members.

We recommend the following metric which can be gathered by the Scrummaster.

Length of Standup Time



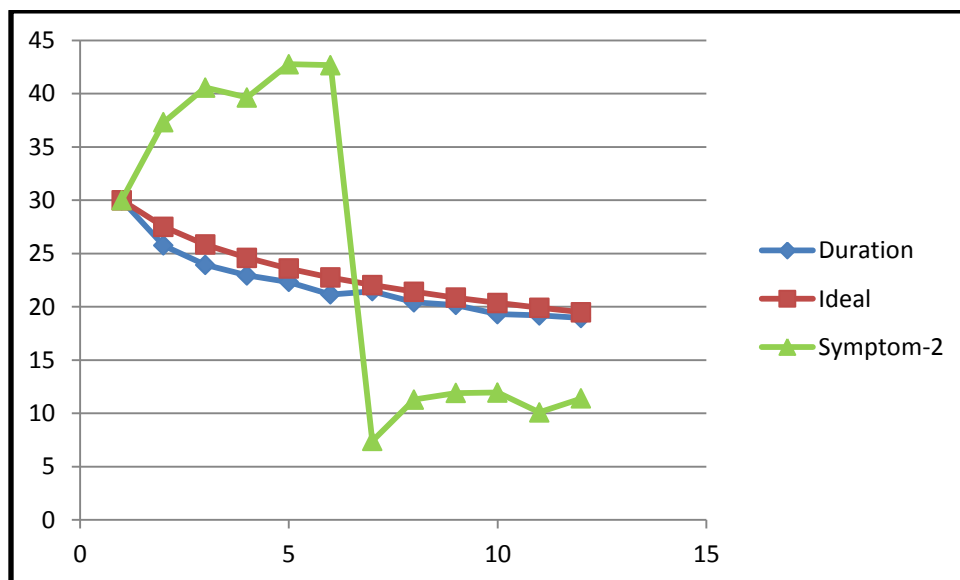
Trend down- Do nothing, team is working out their issues

Trend up- Look carefully at the burn down chart at the same time to see if there is a correlation between impediments that can't be cleared and work accomplished.

Action Plan

- Teams may be getting bogged down with a status update vs giving feedback to the team. Discuss with Scrummaster that team is aware of the reasoning for the meeting (feedback)
- Look for impediments that are not being cleared by the scrum master and instead being resolved in the standup
- Put a limit on how long the members of the standup can talk via various facilitation techniques

Time To Clear Impediments

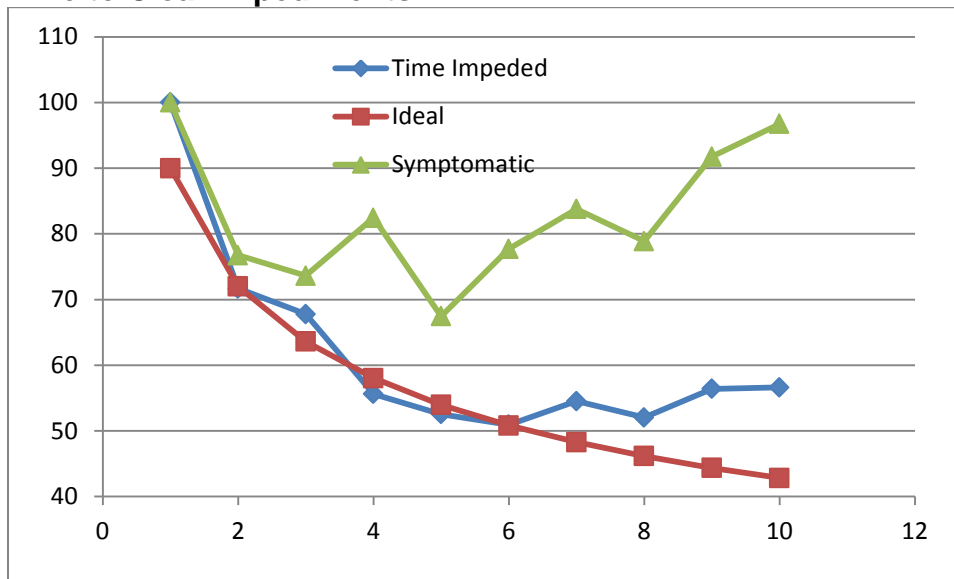


One of the other areas of concerns that came out of the interview process was that impediments sometimes were not resolved in a quick enough time for teams to recover. This would often slow the progress of the sprint and lead to incomplete tasks during the course of the iteration.

Often times the Scrummaster was not empowered to actually resolve some of the issues that came out of the standup meetings. Resources to resolve conflicts across multiple teams were an issue, especially when dealing with other dependent teams that were not Agile.

We suggesting gathering the following metric from your APM tool:

Time to Clear impediments



Team Dynamics Action Plan

- Look for methods to increase visibility to the impediments
- Give the Scrummaster power to make impediments visible to multiple teams so that they can resolve quickly
- Make sure the Scrummaster does not have a conflict of interest with the Agile team, for example the Scrummaster cannot also be the Product Owner or Manager of that team or they will lack the motivation to clear impediments



Target: Improve Team Dynamics to 65%

Part 4: Infrastructure and Agile Automation 30%

One major area of concern for us was Acmes investment in infrastructure and automation for their agile teams.

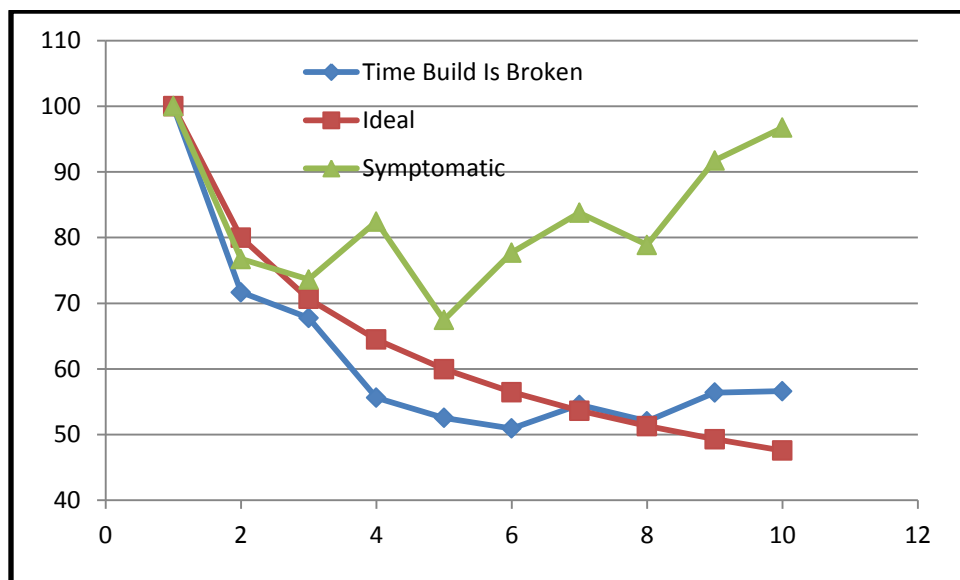
From the quality assurance (QA) team we heard many problems about the length of test time. There were automated tests, but they were not running in parallel environments. Most of the QA team was waiting on automated tests to finish before they could proceed with exploratory testing.

In addition, the build system often produced broken builds. While not a problem in itself, the recovery time for this was too long. Teams would often wait for several days to recover from a broken build and receive a new one for the development team.

We suggest gathering the following metrics:

Time Between Broken Builds

This metric will show you how long it takes for the team to recover from a coding mistake that is caused by either a failed compilation or failed tests. You can gather this metric from your build management system.



Infrastructure and Agile Automation Action Plan

- Look for infrastructure problems that don't allow developers to quickly fix broken code. For example; being able to quickly switch to a new stream/branch or revert a change.
- Inquire about build server strength and compile times, possibly invest in a build farm
- Look at communication and visibility into the broken build. Better visibility will incite developers to fix the build faster.
- Look to invest in multiple test environments so QA engineers don't have to share redundant environments
- Examine Branching and Labeling Structures in the SCM system to improve build performance and promotion



Target: Infrastructure and Agile Automation to 60%

Part 5: Development Practices and Workflow 50%

The development team suffered from some long feedback cycles from the QA team. Many of the user stories were moved from each stage of development process but the QA team was still testing incomplete code mixed into the *Trunk* release. This resulted in incomplete tests until towards the end of the iteration when the code was more complete.

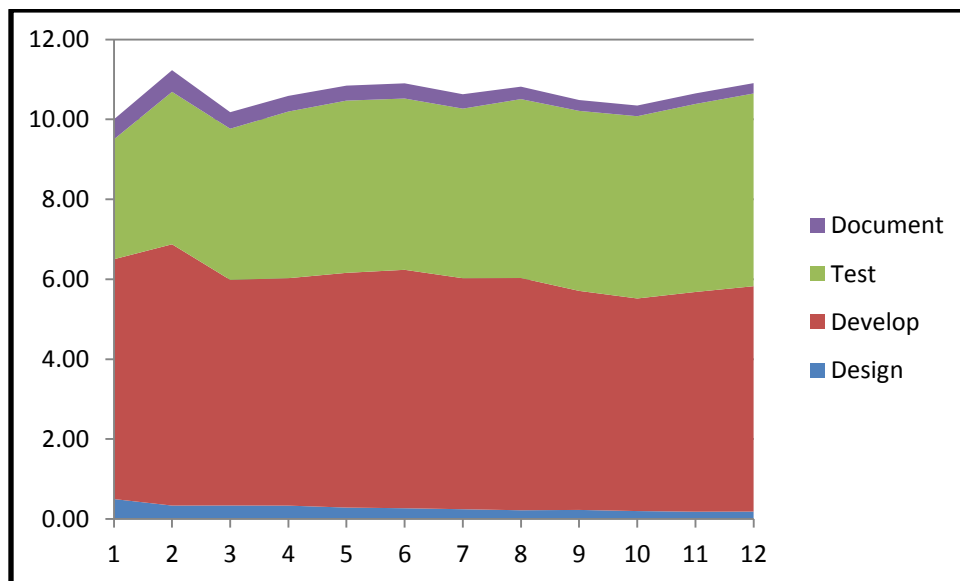
The development team also would wait to deliver changes for fear of producing a broken release to the QA team. Developers would wait to check in changes until the last minute causing a very high length of time between development (DEV) and QA before a release would make it out the door.

Dependant teams also would not resolve their issues between them. Often each team did not know what issues depended on their code changes, neither team knew of what user stories were complete at the end of the build cycle.

We suggest gathering the Metric “Time Spent Per Development Cycle Per Story”

Time Spent Per Development Cycle Per Story:

This will measure how long each step of the development cycle takes per user story as it works through the development process. It will also show you work in progress for the QA team so bottlenecks are visible throughout the process. This metric can be gathered by a combination of software configuration management (SCM) and APM.



Action Plan

DEV- Engage with development managers to see if these problems exist:

- Story estimation size is off
- Dependencies are not being resolved between teams
- Developers are having trouble accounting for tasks which are not driven by value (i.e. merging etc.)
- Implement a branching hierarchy to make sure all user stories can move through each stage of the process in parallel
- Create a multistage continuous integration system which will execute different tests/builds per each stage of the process and merge code automatically to teams where the tests/builds have completed

QA- Engage with QA managers to see if these problems exist:

- Dev team is delivering builds with features “undone” or “half-done”
- Examine if automated tests are taking too long or are a manual process



Target: Development Practices and Workflow to 60%

Action Plan Checklist:

The leadership team and AccuRev consultants generally agree that overall progress has been good. There are a few pockets within the organization that have had productivity problems, largely relating to infrastructure, personality conflicts, and resource requirements, which can be addressed in the near term. Across the organization, coordination against cross-group plans, across geographical locations, and on sharing best practices has been challenging. The team recommends stepwise improvements starting with documenting and tracking cross group dependencies.

***Green items can be purchased as an AccuRev services/product offering*

Product Ownership

- Make sure user stories where well written in typical “user story” form
- Product owner should participate in planning sessions and product demos
- Product owner can manage different inputs from sales, marketing and other stakeholders well
- Plan a grooming meeting where the backlog can be prioritized

Iteration Planning and Tracking

- Make sure user stories are sized correctly
- Include all members of the scrum team in the planning session to build consensus on story points
- Examine if impediments are being cleared by the Scrummaster
- Look for automation techniques such as automated testing, automatic merging and continuous integration / deployment

Team Dynamics

- Teams may be getting bogged down with a status update vs giving feedback to the team. Discuss with Scrummaster that the team is aware of the reasoning for the meeting (feedback)
- Look for impediments that are not being cleared by the scrum master and instead being resolved in the standup
- Put a limit on how long the members of the standup can talk via various facilitation techniques
- Look for methods to increase visibility to the impediments
- Give the Scrummaster power to make impediments visible to multiple teams so that they can resolve quickly
- Make sure the Scrummaster does not have a conflict of interest with the Agile team, for example the Scrummaster cannot also be the Product Owner or Manager of that team or they will lack the motivation to clear impediments

Infrastructure and Agile Automation

- ❑ Look for infrastructure problems that don't allow developers to quickly fix broken code. For example; Being able to quickly switch to a new stream/branch or revert a change.
- ❑ Inquire about build server strength and compile times, possibly invest in a build farm
- ❑ Look at communication and visibility into the broken build, better visibility will incite developers to fix the build faster.
- ❑ Look to invest in multiple test environments so QA engineers don't have to share redundant environments
- ❑ Examine Branching and Labeling Structures in the SCM system to improve build performance and promotion

Development Practices and Workflow

- ❑ Story estimation size is off
- ❑ Dependencies are not being resolved between teams
- ❑ Developers are having trouble accounting for tasks which are not driven by value (i.e. merging etc.)
- ❑ Implement a branching hierarchy to make sure all user stories can move through each stage of the process in parallel
- ❑ Create a multistage continuous integration system which will execute different tests/builds per each stage of the process and merge code automatically to teams where the tests/builds have completed
- ❑ Examine if automated tests are taking too long or are a manual process
- ❑ Dev team is delivering builds with features "undone" or "half-done"

Contact Information:
info@agilecycle.com

Lexington Corporate Center
AccuRev Inc.
10 Maguire Road, Bldg. 1, Suite 112
Lexington, MA 02421
781-861-8700