How can my organization know if our Agile transformation is successful?

October 2016

Scope of Report

It is commonly accepted that most organizations today have moved, are moving, or are evaluating a move toward the use of the Agile methodology. This report considers: (a) why the move to Agile; (b) what it means to adopt the Agile methodology to incur a transformation; (c) how to measure to know if your transformation is successful; and (d) how to insure that the effects of the transformation are continued.

Why the move to Agile?

An IT organization has certain responsibilities that relate directly to their business client and the rest of the organization. From a business perspective, there are five (5) core goals for any IT team.

1. Effectively manage workflow
2. Proactively manage end user expectations
3. Accurately plan, budget and forecast deliveries
4. Accurately estimate deliverables
5. Show value to the organization and the client

Agile, when properly adopted, has been shown to be an effective development method that addresses each of these five goals. As with any new business strategy, the move to Agile would be an attempt to optimize business efficiencies that affect the bottom line and the client-supplier relationship.

What is Agile transformation?

Tom Cagley has suggested that a transformation is a "complete or major change in someone's or something's appearance, form, etc."; in other words, a changeover, metamorphosis, transfiguration, or conversion. Transformation "evokes a long-term change program that will result in a large-scale, strategic change impacting a whole organization (or at least a significant part)". For Agile, it means fostering an environment of teamwork, trust, and open communication to facilitate continuous or frequent delivery of working software.

When an organization embraces such a change, it typically has gone through several stages. First, *discovery* -- a realization of organization needs and how you will attempt to fulfill the needs through a process solution. This is also characterized by knowledge gathering and process analysis. Secondly, *proof-of-concept* – coordination through the organization to solicit sponsors and stakeholders, and assign participants to test the solution. This is executed through a pilot program, or a sampling of teams to use Agile, to generate interest and enthusiasm. Using the lessons learned, and positive and negative feedback, the organization then moves to *definition*, a more structured approach to implementing Agile. The last phase is *institutionalization*, in which the transformation is complete, and Agile is used throughout the organizational IT community. This is exemplified as not just a practice, but a 'core foundation' based upon innovation and business value.

Do we only start to measure when institutionalization occurs, or do we measure through all the process steps to realize when we have arrived at transformation? Obviously, the answer is that we implement metrics as the

270 West Lancaster Ave., Liberty Square B-2, Malvern, PA 19355 • 1-610-644-2856
info@softwarevalue.com • www.softwarevalue.com

process evolves to be able to measure process outcomes, adjust the implementation as necessary, continuing to progress until the goal is reached.

What then do we measure to gauge transformation?

Scrum is a common approach to implement Agile project management. Other Agile and Lean frameworks include Extreme Programing (XP), Crystal, and Scaled Agile Framework Enterprise to name a few. The measures and metrics mentioned in this paper can be applied to most if not all.

There are several key metrics that are used to measure the Scrum environment. To review the terms and the process, the following is the framework which is being measured.

- A product owner creates a prioritized requirement list called a product backlog.
- During sprint planning, the team pulls a subset from the product backlog to accomplish in a single sprint.
- The team decides how to implement the features that are represented in the subset.
- The team has to complete the work in a 1-4 (2 weeks being typical) week sprint.
- The team meets each day to assess its progress (daily Scrum or Stand-up).
- During the sprint, the Scrum Master facilitates delivery of value.
- By the end of the sprint, the features (work performed) meet the definition of done and are ready for delivery.
- At the end of the sprint, the team engages in a sprint review and retrospective.
- For the next sprint, the team chooses another subset of the product backlog and the cycle begins again.

The following are the recommended metrics based upon process measurement within that framework. All of them imply that there are organizational targets that once met would support the transformation.

1. Velocity and Productivity

According to the Scrum Alliance: "Velocity is how much product backlog effort a team can handle in one sprint. This can be estimated by using the historical data generated in previous sprints, assuming the team composition and sprint duration are kept constant. Once established, velocity can be used to plan projects and forecast releases."

Velocity is a measure of throughput - an indication of how much, on average, a particular team can accomplish within a time box. Velocity can be gauged by the number of user stories delivered in a sprint, by the number of story points delivered in a sprint, or by the number of function points delivered in a sprint. Since user stories are not generally considered equal in complexity or time to develop, they have too much variability to be a reliable measure. Story points are subjective and are generally only consistent within a stable team. Again there may be too much variability to measure at an organization level, or across teams.

While story points provide the micro view within teams, we need some way to measure the macro view across multiple teams. Function points can be used at the inception of the project to size the backlog, to determine the deliverability of the minimum viable product and to capture actual size at completion. This allows a quantitative view of volatility. In addition, function points are a rules based measure of size therefore can be applied consistently and are useful for standardizing velocity or productivity. Productivity is size/effort, expressed as

function points delivered per FTE or team member. Using function points as a basis for size, an organization can compare performance within dynamic teams and to the industry through the use of agile benchmark data.

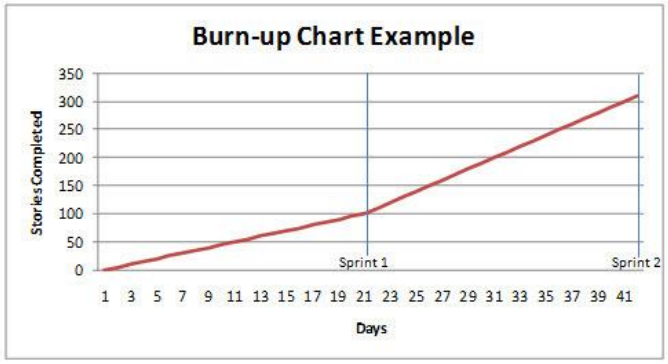
2. Running Tested Features (RTF)

In general terms, the Running Tested Features (RTF) metric reflects "how many high-risk and high-business-value working features were delivered for deployment. RTF, counts the features delivered for deployment denominated per dollar of investment. The idea is to measure, at every moment in the project, how many features/stories pass all their (automated) acceptance tests and are known to be working". The two components are time (daily) and the number of running, tested features ready for delivery to the business client. This metric is often used in environments where operations or production environments are "owned" by separate organizations (often true in DoD and Government environments).

3. Burn down/Burn up charts

According to Wikipedia, "A burn down chart is a graphical representation of work left to do versus time. The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal. That is, it is a run chart of outstanding work. It is useful for predicting when all of the work will be completed."

A burn up chart tracks progress towards a projects completion. In the simplest form, there are two lines on the chart. The vertical axis is amount of work, and is measured in units customized to your own project. Some common units are number of tasks, estimated hours, user stories or story points. The horizontal axis is time, usually measured in days.



These charts can allow you to identify issues (e.g. scope creep) so adjustments can be made early in the cycle. They are also effective tools for communicating with clients and management. The advantage of a burn up chart over a burn down chart is the inclusion of the scope line. It also allows you to visualize a more realistic completion date for the project, by extending a trend line from the scope as well as the completion line. Where the two trend lines meet is the estimated time of completion.

4. Technical Debt

Technical debt is a measure of the corners cut when developing a functionality (e.g. to prove that the functionality can be implemented and is desirable) the code may be written without full error trapping. As technical debt increases, it can become harder to add new features because of constraints imposed by previous poor coding. The measurement of technical debt was introduced in parallel with Extreme Programming (XP) which introduced the concept of "refactoring" or regularly revisiting inefficient or hard to maintain code to implement improvements. XP builds in refactoring, restructuring and improving the code as part of the development process. Technical debt is typically measured using code scanners which use proprietary algorithms to generate a metric based on the number of best practice rules that a particular piece of code infringes.

5. Defect Removal Effectiveness (DRE) and Defect Escape Rate (DER)

Measuring quality has always been a key metric, regardless of the life cycle methodology. The two key metrics in this area measure the ability to remove defects prior to release where:

$$\textbf{DRE} = \frac{\textbf{Defects found pre-release}}{\textbf{Defects found in production + Defects found in pre-release}}$$

The inverse is called the Defect Escape Rate.

The question usually arises over the time frame for a 'release'. Quite simply, it depends on your delivery schedule – if you do a real release every 2 weeks, then that may be your measure of time. It is important to be consistent. As with any defect measurement, you will have to decide what priority defects are considered and are they all treated equally in the equation.

6. Work Allocation

There are three team metrics which can be used to support the outcomes of other metrics (cause and effect). The organization makes a sizable investment in building a solid cross-functional team with the right expertise for the product development. To protect the investment there is a key focus on building core product teams with deep product and technology knowledge. Rotating team members reduces the team scalability as continuity is constantly broken between releases. The following metrics are mainly targeted to gauge impact of team assignments, team changes between releases, and how the time is actually used – all which can affect delivery and costs:

1) Team utilization is quantified by the Team Utilization Quotient (TUQ).

TUQ = Average time spent by team on the project

Example: Utilization is 10 resources for 5 months project.
- 4 resources joined in the beginning
- 2 resources joined after 2.5 months (50% project left)
- 4 resources joined in the last month of the project (25% project left)

TUQ = {(4*1)+(2*.5)+(4*.25)}/10 = .60 = 60%

2) Team scalability is quantified by the Team Scalability Quotient (TSQ):

TSQ = % of the team retained from the previous release

In a TUQ example, we built a team of 10 people. The team had low utilization because of team assignments. Assuming the team is ready to take on next the version of the product, if you replace half of the team members with newer members to work on the new product release it reduces team scalability by 50%.

The third team metric is Work Allocation. This is a simple chart showing what percentage of available time was spent across all work categories for the sprint. Time activities should not only consider development activities but must include the time spent with clients, customers and stakeholders. In Agile, which fosters a cooperative environment, time needed for communication and feedback is as important as the time to code and test.

The use of these metrics should encourage resource managers, Scrum masters and Scrum coaches, to carefully consider how time and resource allocation impacts team efficiency and scalability. The transformation of the organization is from hero building to team building, and if you want to gain a fair ROI, you will invest in developing cross-functional teams. Obviously, disrupting teams will not generate the delivery responses you seek. Conversely, as team dynamics are fostered and improve, so will velocity.

7. Customer Satisfaction and Team Satisfaction

Last but certainly not least, one of the measures which is highly revealing of performance is customer satisfaction. Customer satisfaction answers the question of whether the client is happy with the delivery, quality and costs of the functionality being delivered. Satisfaction provides a view into how the team is perceived by the clients.

Team satisfaction measures how the team is affected by agile adoption. Agile transformation provides an environment that values technical innovation, collaboration, teamwork, and open and honest communication which yields higher team satisfaction. Team satisfaction is positively correlated to productivity. Team satisfaction can be an indicator of how well the organization has implemented Agile.

How do you know that the effects of the transformation will continue?

The most common answer is “you don’t know for sure”. As a matter of record, experience has shown us that without continued measurement and adequate coaching, teams fall into entropy and lose efficiencies. A measurement feedback model should be in place to monitor performance levels, to know when to get coaching and how to address process improvements as needed.

At any point in the transformation, an independent assessment may be in order to determine where you are in comparison to where you want to be. Feedback from an assessment is critical for developing a fact based plan for improvement.

Conclusion

The journey to transformation involves a cultural organizational change which can be thoroughly measured using common Agile metrics. The efficiencies of the new Agile environment can be quantified, maintained and improved through the use of a continuous measurement framework and periodic independent assessments.

References

SPAMCAST, Tom Cagley. Nov 2015. So You Want A Transformation! https://tcagley.wordpress.com/2015/11/10/so-you-want-a-transformation/

Agile Metrics: Running Tested Features, 9 June 2014, https://www.scrumalliance.org/community/articles/2014/june/agile-metrics-(1).

Wikipedia Burn down chart – https://en.wikipedia.org/wiki/Burn_down_chart.
Metrics Minute: Burn-Up Chart, Tom Cagley. https://tcagley.wordpress.com/?s=burn+up

Metrics Minute: Burn-Down Chart, Tom Cagley. https://tcagley.wordpress.com/?s=burn+down
Clarios Technology: What is a burnup chart?,http://www.clariostechnology.com/productivity/blog/whatisaburnupchart
Technopedia: Technical Debt, https://www.techopedia.com/definition/27913/technical-debt
XBOSOFT: Defect Removal Effectiveness – Agile Testing Webinar Q&A, https://xbosoft.com/defect-removal-effectiveness-agile-testing-qa/.
Agile Helpline: Agile Team's Efficiency in Various Resource Allocation Models. http://www.agilehelpline.com/2011/05/agile-team-efficiency-in-various.html
DCG Software Value. Webinar: Agile Metrics – What, When, How, David Herron. Nov. 2015.