Sample Agile Assessment

Prepared by:

Bob Hartman

June 10, 2008

Introduction

Purpose

The purpose of this assessment is to compare the various practices used at Sample to the software industry as a whole. This information was then used to create the basics of an action plan for an agile transition throughout the organization. The comparative results are broadly characterized under six different agile components. Although only a small portion of the organization have been involved in agile projects this grouping was chosen because it also gives an indicator of agile readiness in various areas. The resulting agile adoption outline makes extensive use of this information.

Understanding the Results

It is important to understand that these results are based only on interviews conducted over the course of two days with no actual observation of teams in action. As such it is a small snapshot and may be skewed by the beliefs of any of the individuals that were interviewed. The answers to most questions were confirmed with answers by multiple people; however that is not a guarantee that all results are 100% valid.

It should also be noted that this assessment, like all assessments of this type, is not precise. It is based on the experience and knowledge of the interviewer. The results are estimates that are used as a baseline for recommending a course of action. As such, absolute precision is not a requirement.

The results are broken into 6 different areas: Product Ownership, Release Planning and Tracking, Iteration Planning and Tracking, Development Practices/Infrastructure, Testing Practices and Team. In each area a percentage estimate is recorded with 0% representing “extremely poor”, 30% representing “poor”, 50% representing “average”, 70% representing “good”, and 100% representing “excellent.” A short description of the findings is included for each area of the assessment.

Assessment Results

Overall Impressions

Several very good things stand out from the various interviews conducted. As expected, there are also some deficiencies, including some that need immediate attention. It should be noted that the interviewees were not told the purpose of the interviews until the meetings actually were occurring. Once they were told the reason for the meeting all were extremely cooperative, open and thoughtful. The people interviewed acted very professionally, while also holding nothing back. This is noted because it clearly backs up the assertion by management that the culture is one of open collaboration. The items that were noted on the plus and minus side of the ledger are quite varied in nature with only a few mentioned in this section. Other pluses and minuses are mentioned in context within the actual assessment results sections.

The most obvious plus noted was the very near unanimous support for the agile process. Almost every single person interviewed knew some projects were being done in an agile manner and had heard good things about the results. Only two individuals were not extremely enthusiastic about the potential of switching to agile. However, those individuals were not against the plan, they simply didn't know enough of the details to have an opinion one way or another.

In every interview people agreed that all of Sample's employees are extremely loyal to the company and dedicated to its success. This is an important plus because transition is often difficult. With everyone committed to success it makes transition a bit easier for all of them to handle.

It was also noted that there was an extremely high correlation of answers from different people. In many assessments different areas of the organization give vastly different answers to the same question. In the interviews conducted for this assessment the high degree of nearly duplicate answers was much higher than normal. This is not to say there were no discrepancies, because there were, but those were generally in the context of some very specific projects.

Unfortunately, there are some negatives that are important to mention in this section as well. The primary negative being what appears to be a strong disconnect within the XYZ project. In fact, the only thing the members of that team agreed upon was that they had problems. The XYZ project is being done in an agile manner, but it appears that it was done with little to no training and the team is having significant issues because of that. The team is also quite large which adds to the problem.

The second negative was the use of an alternative agile methodology (FDD) within one group. While not harmful by itself, mixing two different methodologies within the same organization will often cause harmful division between teams. They compete for resources in different ways which causes them to become competitive in a negative manner.

Product Ownership – 30% (below average)

This low score is based on interview answers that consistently pointed out a lack of knowledge about whether product management, business analysis, or the project management groups were in charge of the features to be built. One respondent went so far as to say "Product Management has no say because they are always overruled by the developers." There is significant concern in this area, especially for an agile transition since this is a critical practice. The number one cause of agile failure is a non-functioning product owner role.

This lack of clarity is concerning and will need to be addressed. This will need to be a baseline practice that each team follows as much as possible. Based on results with prior clients, having the product management role be the product owner, supported by the business analyst and technical leads can work well. This sounds like a recipe for disaster (decision by committee), but it is not that way in practice. The group works to refine the requirements so they are digestible by the team. The product manager and business analyst work together to define the business value of each requirement and prioritize the list. In the end, the product manager has final say on priorities because it is vitally important that business value (which accounts for customer value, internal and external risk, size, etc.) is the driving force.

Release Planning and Tracking – 40% (slightly below average)

The concerns in this area are mostly in regard to the agile teams. It appears that some shortcuts were taken when projects were started and as a result some of those teams do not have a properly structured product backlog. This lack of structure shows up as ill-defined requirements, or even no requirements in one case! Agile requires a well formed prioritized product backlog which is usually finalized (in structure and initial list) at the release planning meeting.

It was also not clear whether there is a well defined release vision for all projects. Conflicting answers were received on this point. Having a well defined vision helps limit scope creep for agile projects. Studies have shown that most scope creep happens outside of the original vision of a project. Having a vision statement also gives the project team a definition of their goal that helps orient them when they feel lost.

Iteration Planning and Tracking – 60% (slightly above average)

Overall, Sample has an excellent group of project managers and tracking systems are in place. There are some anomalies which keep this score from being significantly higher. In particular, the Enterprise Reporting team struggles mightily with iteration planning. Other projects are having similar struggles, just not as pronounced. It is also not clear that iteration tracking is communicated well on the agile teams. Traditional teams seem to be doing the equivalent tracking much better than the agile teams.

For the agile teams it was not clear how many were successfully using the web based tools, nor was it clear if standard burndown or other charts were visible. Visible status indicators (often called information radiators, or big visible charts) are a usual agile practice. They enable the team or anyone

else to see at a glance how the team is doing. Without these status indicators it is unclear who knows how a team is doing at any given point in time.

Development Practices/Infrastructure – 50% (average)

Sample has many excellent practices in place. The use of staged environments seems to be a good practice, however there were many interviews where significant concern was expressed about the environments. Most concerns were about how long it takes to change environments, but some were about how the final promotion to production sometimes changed things at the last minute. Taking too long to move from one environment to another may be a significant concern for agile projects. Having production code that differs from the final developer release code is a concern regardless of development methodology.

While there are many good things Sample does, there is also a lack of formal development practices such as the use of test-driven development and the use of design patterns. In terms of agile adoption, the use of test-driven development will significantly increase code quality and shorten development time. Design patterns help with architecture, which is exceedingly complex at Sample and in an agile environment some concern was expressed about how that would work.

It was noted that there is a formal practice of code review for every checked in module. While this is an excellent practice, it is likely that the way in which it is done will need to change slightly in order to support agile projects. Concern was expressed that many developers check in huge chunks of code at once, causing backups in the code review process. Agile projects will need code reviewed in near real-time in order to be effective.

Testing Practices – 60% (slightly above average)

Testing is done in more or less industry standard fashion. For the agile projects there is an attempt being made to do testing within the iterations. There also is some testing done early via prototyping. All of these bump the score in this section to above average. A negative is the lack of an automated testing framework that is used consistently. In particular a key success factor for agile is the ability to continue to expand the automated unit and acceptance test suite so that during the nightly build regression errors are found. The use of Mercury tools can help in this regard, but there may be a need to look at other testing frameworks to enable testing below the GUI layer (the business logic layer). This is necessary so that agile projects don't have to build out to the GUI layer prior to testing.

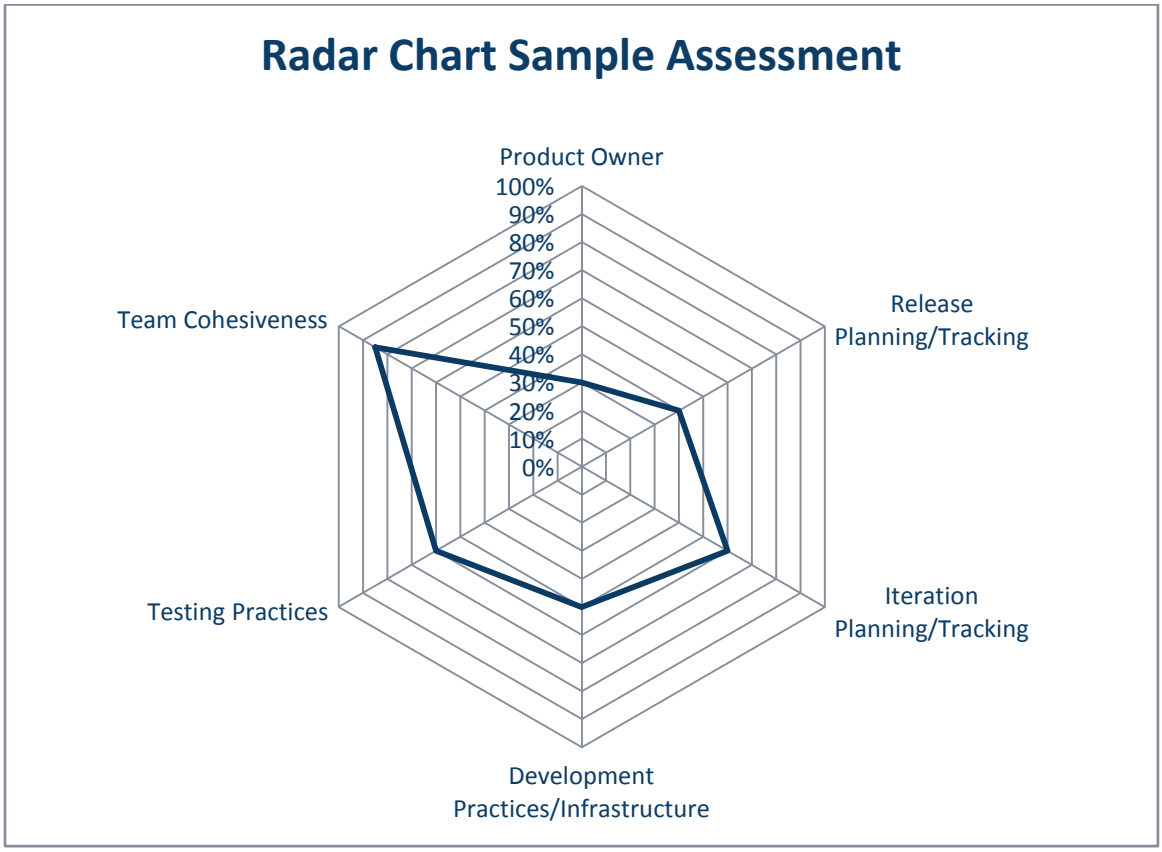
While testing is slightly above average in general, the XYZ project is again an exception. Multiple comments were made about insufficient or no requirements documentation being present. This lack of information caused testing to suffer significantly. Without sufficient documentation the testers simply did not know what to test or what to expect. It is allegedly taking them significant time finding enough information to perform their job.

Team Cohesiveness – 85% (significantly above average)

Sample has a fantastic culture and environment. While there are some areas of friction that need to be smoothed out (product manager, business analyst, project manager interactions for example), the general consensus is everyone believing the organization is a great place to work. Collaboration is generally on the very high side. Different groups seem to work well together in most cases. There is a strong camaraderie between almost everyone interviewed.

The only negative noted is the perception that the long range vision seems to change too frequently. A common comment was the direction of the company keeps changing every time a big customer says something new. Most of those interviewed feel this is a detriment, but also believe the company is strong enough and motivated enough to succeed in spite of the changes in direction. It was also mentioned that this is being worked on at higher levels, but it is unclear whether that information is correct.

Graphical View



Recommendations

Immediate

This section is rarely included in an action plan, however some of the results make it necessary for immediate action to be taken.

The first area of concern is the XYZ project. This project needs to have some coaching for individual roles (product manager, project manager, business analyst) in order to help them better understand how to do things in an agile way. It also needs the entire team to have more understanding of how agile works, particularly in the areas of iteration planning, feature breakdown and estimation. This can be done through coaching, but it is much easier to do through classroom training. In order to reinforce the knowledge transfer there needs to be follow-up coaching done to help the entire team through an iteration planning session. Finally, it may be necessary to discuss with management how the team could be logically split into two smaller teams working on the same project. The single large team is being too overwhelming for people knew to using an agile process. The recommendation is to do all of this as soon as possible because this project seems to be in serious need of some corrective action.

Management needs to make some decisions for the sake of consistency. In agile there are many different choices that can be made, and each team appears to be making them a bit differently. Normally that would not be a concern, however for transition purposes it is better to train for a particular baseline of practices that are non-negotiable while also making it clear that other areas can be modified to best meet the needs of the team. The primary concern in this area is that the agile teams are not being consistent in some ways that are so fundamental they should be non-negotiable. In particular the agile teams are not being consistent about what “done” means at the end of an iteration. It also seems that not all teams are using the same iteration length, which is not a horrible practice, but it is definitely not a best practice. There are other examples of inconsistencies, but the point has been made. The recommendation is to come to agreement on the baseline of practices that will be taught so that the teams all have a common starting point rather than making it up as they go along.

Finally, someone should look into the expressed concern of production code varying from the final developer released code. If this is actually occurring it seems that it is a support nightmare in the making. This concern was raised by several developers in interviews.

Short-term

As was expected, there is a significant need for a common baseline among agile teams. They are using a variety of practices including experimenting with different ways of tracking iterations (some use Rally, some use TargetProcess, others use the basic cards on a wall approach). It will be necessary to do a baseline agile training for all team members on agile teams so they all will be using the same basic cookbook as a starting point. In addition, all teams should get basic training on release and iteration planning as well as estimation. Specific roles should get training on practices specific to their role. In

particular these should include project managers, product managers, business analysts and technical team leaders (both QA and development). Each of these trainings will have some things in common, but they will be significantly different in other areas therefore it is not possible for someone to get their role based training while attending a session for another group.

As teams have been trained and begin using an agile approach it will be necessary for coaching to take place to reinforce the training. This will primarily revolve around the normal agile meetings (release planning, iteration planning, iteration demo, iteration retrospective, daily standup, etc.). The goal is to have each team fully self-sustaining after no more than three iterations.

As discussed in the initial proposal, an executive session designed to explain the concepts behind agile development as well as guidance on what to expect is necessary. The transition will not succeed without full support from the very highest levels of the organization. The session will have significant time to get questions answered so that everyone is comfortable with the approach that will be used.

Longer-term

Everyone in the organization needs agile training. As teams are getting ready to transition to agile projects the team can be trained and coached as described in the section above. In addition, people that are supporting agile projects in some fashion (release management, documentation, etc.) should receive basic agile training so they know what to expect from the agile teams.

Development teams could use some additional agile practices which will enable them to work more efficiently. In particular, test-driven development and the use of design patterns could make a big difference in coding quality and speed. Both of these courses are taught by Net Objectives with excellent results. Most organizations find that starting with test-driven development gives the fastest return on investment and it seems the same would be true at Sample. Use of design patterns is a good practice that would significantly improve the ability to have an emergent architecture that fits better within the agile model.

Overall

In general, these results are very close to what was expected based upon initial conversations with various people within Sample. As a result, the overall recommendation is to closely follow the original proposal in terms of scope and types of training and coaching. The only exception to this is the immediate need for intervention with the Enterprise Reporting project.

Contact Information

For further information on this assessment please contact:

Bob Hartman
President, Sr. Trainer and Coach
Agile For All
bob.hartman@agileforall.com
Office: 303-766-0970
Fax: 866-620-8792