2008

PACIFIC NW

SOFTWARE QUALITY CONFERENCE

Collaborative Quality

October 13-15, 2008

Conference Paper excerpt from the

CONFERENCE PROCEEDINGS

Collaborative Change

Debra Lavell
Intel, Corporation

Abstract

At Intel, we have implemented a more effective approach for capturing and sharing lessons learned through retrospectives. A retrospective is a ritual where a team comes together several times during the lifecycle of the program to discuss what is working well and uncover opportunities for improvement. Retrospectives are a powerful way to help teams apply the wisdom learned with the intent of long-term behavior change.

As with any organization or business process change undertaking, one of the most difficult challenges to overcome is getting an entire company to change their culture and modify the way they work and behave. Introducing the retrospectives methodology into Intel has been no different. In this paper, we explore the people side of bringing retrospectives into an organization. The tips and tricks are from a facilitator's perspective. This paper will explore how one handles the human aspects of introducing collaborative change.

Bio

Debra has over 10 years experience in quality engineering. She currently works as a Program Manager in the Corporate Platform Office at Intel Corporation focusing on Retrospectives and Organizational Learning. Since January 2003, Debra has delivered over 200 Project and Milestone Retrospectives for Intel worldwide. Prior to her work in quality, Debra spent 8 years managing an IT department responsible for a 500+-node network for ADC Telecommunications. Debra is a member of the Rose City Software Process Improvement Network Steering Committee. For many years, she was responsible for securing the monthly speakers. She currently is the President of the Pacific Northwest Software Quality Conference, Portland, Oregon. She holds a Bachelor's of Arts degree in Management with an emphasis on Industrial Relations. To contact please email: Debra.S.Lavell@intel.com

Introduction

Due to the increased complexity of our product and service development processes at Intel, as well as the expanded use of globally dispersed development teams, we have implemented a more effective method for capturing and applying our program and project lessons learned. This approach is known as the retrospectives methodology. A retrospective is a ritual where a team comes together several times, (we recommended three strategically placed) during the lifecycle of the program so the team can pause and reflect on what is working well (so they can keep doing it) and discover collaboratively where there are opportunities for doing things better (start or stop doing it). Retrospectives are a powerful way to help a team apply the wisdom learned with the goal of long-term behavior change.

As with any process change, one of the most difficult challenges to overcome is inspiring an entire organization to change their culture by collaboratively changing the way they work, and how they behave. Introducing the retrospectives methodology into Intel has been no different. This paper will explore the people side of bringing retrospectives into an organization. The tips and tricks are from a facilitator's perspective focusing how to handle the human aspects of introducing collaborative change.

Changing human behavior

One reason organizational change is so difficult, is that it requires sustainable changes in how people do their work and how they behave while performing that work. By working collaboratively, the team members identify what needs to change and what the team is willing to commit to do (rather than management dictating what they want to see change). This "wisdom of the crowd" deciding what is the most important to the success of the team dramatically increases the likelihood the change will happen. Introducing an effective retrospective methodology within an organization requires the management of many facets of change:

Adoption of a new way of doing things. Many program and project teams have never experienced continuous process improvement efforts through multiple retrospectives across the lifecycle. They normally wait until the end (typically called a Post-Mortem or Post Project Review) to discuss what happened, which is too late to affect the current program. Most team members move quickly on to other teams and are not interested in spending a day reviewing what happened on the program. The retrospective methodology requires a series of events where team members collectively meet face-to-face (if possible), at strategic points during the development lifecycle to discuss what is working and what needs to be improved. Collaborative discussions and activities are the basis of a retrospective.

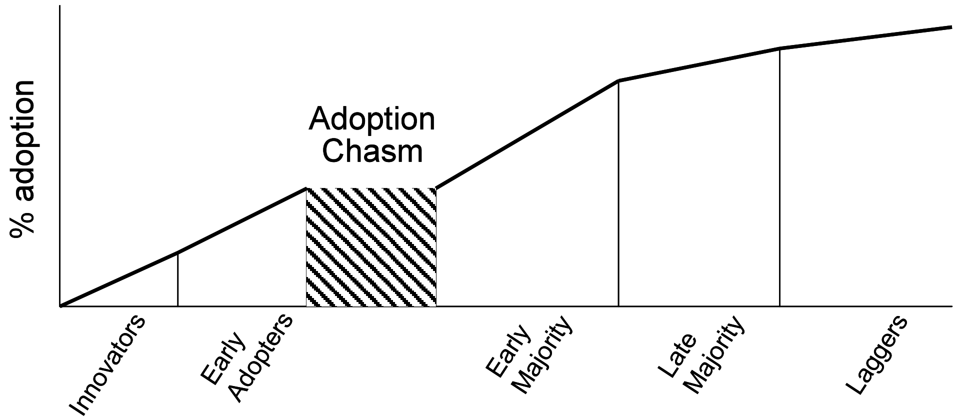
Management of a new deliverables. After a post mortem, some team's output is a bulleted list of items, which is never reviewed or discussed again. The deliverable from a retrospective is an action plan, written collaboratively by those who have a passion to fix the problem. Every action plan has next steps broken down into tasks with an owner, who is passionate about ensuring the change is implemented. The team manages the completion of the action plan by adding it to their weekly product development meetings, monthly goals, and possibly roll up to a group level as IMBOs (Intel Management by Objective) to ensure the changes desired are visible and progress tracked.

Updating existing systems. To ensure retrospectives and the lessons learned that came from them become institutionalized, work systems need to be developed or existing work systems must be modified to incorporate organizational learning as part of their normal processes. For example, we recommend three milestone retrospectives within the Product Life Cycle, which is the common development lifecycle framework that most teams utilize to manage their programs and projects. Before a team can exit a particular phase or milestone, a retrospective is held to capture learnings. Change happens when collaborative discussions facilitate a culture of "this is how we work to constantly improve."

Replacing old behaviors. This is the most challenging aspect of change. If you want to get the most out of retrospectives, it will require a change the way people work and associated business practices. When the team collectively decides what behaviors need to change, it becomes self managing and self policing. Nevertheless, not without an occasional reminder to help spark forward progress. At three, six, nine, and twelve months, reminders are sent via email and if no response, a 30 min meeting is scheduled to discuss what progress has been made in the implementation of the new behavior. This means that the human side of change has to be managed in order for it to become institutionalized.

Why does institutional change need to be managed?

In order for retrospectives to become a broadly accepted and utilized method for capturing program and project key learnings within an organization, and to see measurable change, the majority of the organization must acknowledge and embrace the change, as something they believe will benefit them. It is not uncommon for new ideas and ways of doing things to be tried, and if successful, accepted by a percentage or slice of any organization known as the "Innovators" and "Early Adopters." However, gaining widespread support for an idea can be a much tougher problem to solve. Below is Figure 1, where Geoffrey Moore and his book titled "Crossing the Chasm" describes the challenges gaining institutional acceptance of retrospectives within Intel. 1



Source: Crossing the Chasm (author modified)

Figure 1: Modified Version of the Technology Adoption Life Cycle

As described by Moore's model, a chasm exists between people and project/program teams willing to experiment with retrospectives (innovators and early adopters), and the remainder of the organization that may not be willing to implement it as a standard method of improving their organizational learning capability (early majority, late majority, and laggers).

Collaborative discussions among the team members of what needs to change and securing advice and support from an influential sponsor, is a critical aspect for crossing the adoption chasm. In addition, fundamental to the process is having an understanding that the value proposition for method changes such as retrospectives is different on each side of the chasm. To the left of the chasm, the value proposition needs to be sufficient to convince the innovators and early adopters to experiment with the method. However, if you want to achieve broad adoption by the remainder of your organization, the value proposition on the right side of the chasm must be sufficient to convince the majority to change the way they collect and disseminate program learnings – a much more difficult task.

As part of our retrospective process, we schedule time to share the results of a retrospective with senior management so they can provide support to the team and remove any barriers to implementing change. It is always helpful to look at an example to better understand the impact of sharing the outputs from a retrospective with management. During the execution phase of a major platform, the Program Manager decided to "try out the new process." After the retrospective, we held a management report out, the purpose of the presentation was to share the recommendations from the team (action plans) that resulted from the collaborative discussions, and to gain senior management support for broad adoption of the methodology across the business unit.

It quickly became apparent the senior manager was satisfied with the current method of performing a single post-program review at the end of the program, after launch of the product. The retrospective output was good, but it did not satisfy the value proposition for broad deployment. The significance of the methodology became obvious during the meeting when the collective voice of the team and the business unit champion successfully demonstrated the mid-execution phase retrospective output applied to the follow-on program that had just entered the planning phase was invaluable "early warning" learning. The business unit now had the means to quickly apply key program learnings to improve two programs in progress the execution improvements on the current program, and planning improvements on the follow-on program.

If the senior manager continued to stay with the old methodology of a single post-mortem, the planning phase learnings would continue to come in too late to affect the next program. At best, the changes may improve the third program on the roadmap – too late for effective change. As a result, the senior manager now requires all programs within his business unit to utilize the retrospectives methodology as part of the standard program management practices multiple times during the life of a project or program.

A Case Study

Over the past three years, we have collected a mound of data from our early adopters. To illustrate the power of collaborative change, meet Team Dolphin (not the real team name. The real name has been removed to ensure confidentiality of the product being developed). The Dolphin program team is responsible for effectively and systematically defining the platform planning deliverables such as platform requirements, value proposition, and scope for the product. The platform planner requested a four-hour retrospective prior to the hand-off to the execution team to:

- Improve platform planning efficiency through repeatability
- Align with platform planning team(s) during the key scope, feasibility and commit milestones
- Ensure all planning documents are achieving a specific work state to drive effective technical analysis

The sponsor decided to hold a six-hour retrospective (broken into two three-hour meetings) so both the planning and the execution teams could participate. In addition, the current platform planner invited the *NEXT* platform planner to sit in and listen to the discussions during the retrospective. For example, the Dolphin program team is just exiting the planning phase and is preparing to hand off to the execution team. The Dolphin planner invites the Penguin planner, who is next generation platform, to attend the retrospective so the Penguin team will have an early warning to issues and concerns. One of the most powerful aspects of their participation is for them to hear the emotion and passion from the Dolphin team when they share significant problems that plagued their planning efforts. Since the Penguin planner was just entering their planning phase, the learnings are immediately applied and impact to the program is instantaneous.

We are using a model at Intel where we locate a "recommender" and a "receiver" who commits to making changes on the next program. In the example above, the Dolphin team is the recommender and the Penguin team is the receiver. During the Dolphin platform retrospective, the team prioritized five areas for improvement. All of the five learnings (that came out of the current planning efforts) were shared (they are the recommender) with the subsequent planner (who has "received" the learnings) and has committed to implement the recommendations by saying "I will stop the repetitive mistakes and implement change."

What did the "receiving" team decide to do differently next time? The top three items are:

1. Ensure a documented plan is achievable. The next planning team has committed to creating and communicating a timeline that includes a much narrower scope with a better understanding of constraints.
2. Add a new position and staff it within the planning effort to focus on identifying opportunities, gaps, and constraints. The next planning team has been involved in the hiring of this key position to ensure the planners have one picture of the program.
3. Improve the use of tools to document and manage requirements. The next planning team is piloting an enterprise-wide Product Lifecycle Management (PLM) tool to integrate platform data and connect teams. The team is

committed to producing better requirements and managing changes as the project progresses.

So what happened?

It is too early to tell. What we do know is that a third program is just wrapping up their launch for the current product and two previous teams have captured learnings they are willing to share. Imagine five generations are sharing and comparing learnings to improve all aspects of the program. We believe collaborative discussions, focused on multiple generation learnings provide a more comprehensive learning opportunity as comparisons can uncover gaps which initiate conversations with experienced planners.

In conclusion, we have captured over 30 learnings within this business group, spanning five generations. All key learnings are captured in a common repository for planners, architects, and engineers to query, browse, and search for learnings, which allow various solutions and experts to talk with about specific problems they are trying to solve. The planner wants to document key practices in the repository so other divisions can benefit from their wisdom. This is a way for all planners to give back to other teams by sharing best practices and innovative solutions. For the repository to be useful, the planners tell us they want a robust search engine that returns logical and ranked learnings so they get to what they are really interested in, they want a web-based solution, and prioritized learnings with data from a trusted source.

Keys to managing collaborative change

Once the value proposition for broad deployment is established and influential sponsors are behind you, the real work begins to ensure retrospective methodology achieves full adoption. The five primary aspects for managing the transitional change management activities are:

1. Determine if your company is a "Learning Organization." The first step is to build the case for organizational learning as a means for continuous improvement, and retrospectives as the method for collaboratively capturing and applying the improvements is to do an environmental scan to find out what is current state. Ask questions such as:

- What is your process to gather lessons learned?
- How often do you require teams to stop and reflect?
- Do you document the lessons learned in a consistent format and in one location?
- Can you effectively search, monitor, and update the documented learnings?

Look for a consistent process to capture lessons learned, multiple touch points during the lifecycle, and a centralized location so the entire organization can query, search, and monitor the learnings. You are well on your way to being a learning organization if these three items are in place.

2. Tailor the methodology. Every organization is unique; therefore, the retrospective methodology must be flexible and scalable to meet the full needs of the team. Start by asking simple questions to understand the current program review process, and then negotiate how much change the team can absorb. For example, instead of starting with performing three retrospectives at strategic points across the program lifecycle, negotiate for one additional retrospective (in addition to the end) as a place to start.

3. Ensure someone is accountable for results. Action plans collectively created in a retrospective, have committed owners, and established due dates, but it becomes easy for the owner to procrastinate on the completion of an action plan when it is in addition to their normal program-related work. Below are three suggestions to prevent apathy from taking over and increasing the accountability of action plan owners.

- Select one opportunity the team whole-heartedly agrees must improve. During the brainstorming of issues, it is easy to get over zealous and want to "solve world peace." Resist the temptation!
- Document the one opportunity in the form of an action plan (working as a team) to fully understand the problem you are trying to solve. It is human nature to want to jump to solution space and fail to clearly state the problem in the form of a well-written problem statement. A good problem statement reads: Lack of X, resulted in Y. Brainstorm together various obstacles to making the improvement, identify who needs to support the change, and the next steps in short, two-week intervals.
- Ensure the completed action plan is accessible to the team. Many times the output from a retrospective is stored on someone's computer hard drive. If possible, create a centralized repository for all learnings so those who need to get to the data can do so quickly and easily. In addition, capturing and documenting the team member names associated with the recommended action improves accountability of the action.

4. Make changes to existing work systems. Change is easier for humans to embrace when it is embedded into well-established work systems. Companies who use a development "lifecycle" or a "software development life cycle" where a program team passes through various stages such as exploration, planning, development, and product launch can link lessons to the entry and exit criteria of the phases and milestones to help reinforce change.

5. Change reward systems: Establish a reward system to recognize and reward teams that make use of the retrospectives methodology and implement the necessary behavioral changes. We have found an effective reward is something desirable, timely, and provides an opportunity for reinforcement of the change. Small rewards, delivered for meeting smaller goals, are many times more effective than bigger rewards. These can be as simple as a hand written thank-you, gift certificate from a good restaurant, or movie passes. Bigger recognition efforts such as monetary awards such as $500 or $1000, extra vacation days off, or annual "extreme performer" incentive award trip to some exotic location. Whatever the reward, invest time and energy to create a reward system that reinforces the behavior changes you desire.

Conclusion

As with any change in the way people perform their work, one of the most difficult challenges to overcome is getting people en mass to modify the way they behave in order for the change to become part of the fabric of the organization. Change is hard; however, working collaboratively to identify what needs to change and then working together to help make the change(s) stick will be easier utilizing the retrospectives methodology.

The first challenge is to define and communicate the value proposition for broad adoption of the methodology and cross the chasm between the early adopters and the mass majority of the organization. Once this is accomplished, the five key steps in making collaborative change are:

1. Determine the starting point based upon an organization's current learning capability
2. Tailor the methodology to ensure problems are collaboratively identified by the team
3. Ensure someone is accountable for implementing the documented action plans
4. Make necessary changes to existing work systems
5. Change the reward systems to reinforce positive behavior

Working together, to collaboratively solve problems is the most effective way for significant change to happen. Going it alone is more difficult and isn't as much fun.

References

1. Moore, Geoffrey A. Crossing the Chasm. New York, NY: HarperCollins Publishing, 1991.