

# Transitioning Software Development and Operations Laboratory to Agile

N. Ivanov<sup>1</sup>, M. L. Pack<sup>2</sup>

<sup>1</sup>Center for Advanced Transportation Technology Laboratory, Department of Civil Engineering, University of Maryland, College Park, MD 20742; ivanovn@umd.edu

<sup>2</sup>Center for Advanced Transportation Technology Laboratory, Department of Civil Engineering, University of Maryland, College Park, MD 20742; PackML@umd.edu

## ABSTRACT

Over the last five years, the CATT Lab at the University of Maryland expanded its cutting-edge software development operation from five full-time developers and 10 students to nearly 30 full-time staff and 90 students. With this growth came many organizational and software development process challenges. In response to these challenges, the CATT Lab implemented staffing and work-flow solutions, as well as a tailored agile development process loosely based on Scrum. As a result of implementation of this process, the CATT Lab is able to handle rapidly changing priorities, maintain positive work environment, and deliver high quality software. The key to success are ability to adopt and tailor processes, empowering individuals, establishing clear hierarchy, measuring performance and setting targets, and introspection and continuous self-improvement.

## BACKGROUND

**The CATT Lab.** The Center for Advanced Transportation Technology Laboratory (CATT Lab) is an applied research and development lab within the Civil and Environmental Engineering Department at the University of Maryland, College Park. The CATT Lab's mission is to develop cutting-edge software products that aid first responders, state and local Departments of Transportation, and researchers in managing and analyzing both real-time and archived data.

The CATT Lab has expanded significantly within the past five years from fewer than five full-time software developers and 10 undergraduate students to nearly 30 full-time staff and over 90 part-time students. As the Lab grew, most of the full time developers were sourced from the large pool of graduating student employees who were performing at a higher level than their peers.

Today, the CATT Lab supports a large number of software development and basic research contracts with state, local, and federal agencies across the entire country. This rapid growth exposed many deficiencies in organization, work-flow, and project management that threatened the long-term viability of the CATT Lab. The remainder of this paper discusses these growth challenges, the significant changes that were implemented, and lessons learned.

## CATT LAB STRUCTURE AND PROCESS CHALLENGES

**Early days of the CATT Lab.** The first products to emerge from the CATT Lab consisted of web-based tools to aid Maryland transportation operations staff in visualizing, exploring, and computing basic stats on their own data—which consisted of crash data, speed and volume sensor measurements, and other related data. As these products gained interest with additional state and local agencies, the CATT Lab was commissioned to build significantly larger software products including the Regional Integrated Transportation Information System (RITIS) – a data fusion system allowing many transportation agencies and first-responders to collaborate in real time during regional incidents. With this system, the CATT Lab went beyond the development of prototype applications to the development and support of 24/7 operational systems with thousands of users in multiple time zones. Additional large work contracts were initiated at the same time – further adding complexity, staffing needs, and expectations to the CATT Lab developers and management.

**Functional development challenge.** As the CATT Lab staff expanded to meet the demands of new contracts, developers organized themselves in a functional manner – creating a number of development teams focused on individual technologies and products. *Table 1* shows the team structure and responsibilities for each of the teams.

**Table 1 - CATT Lab Development Team Structure and Responsibilities**

Team	Responsibility
DBA Team	Design, implementation, and maintenance of numerous relational databases.
Java Team	Extract, transform, load (ETL) tasks required to integrate new data sources into various CATT Lab databases.
Web Team	Development of front-end components showing real-time transportation data.
Analytics Team	Development of visual analytics using archived data sets using the Flex application framework.
IT Team	Hardware infrastructure to support development and operations.
GIS Team	Geographic Information Systems (GIS) related support services.

All teams were required to work closely together to achieve the common goal of delivering software products. However, in practice this proved to be difficult for the following reasons:

- **Diversity of process.** Development processes grew organically within each team with little coordination between other teams.
- **Ad-hoc leadership.** The CATT Lab organizational structure was relatively flat without a clear chain of command. The teams were either led by the strongest developer, the strongest personality, or in some instances had no leadership at all.
- **Work-flow management.** To complete a deliverable, all teams were simultaneously required develop their individual components, perform their own

testing, and finally produce a deliverable for system testing. However, if two teams were working on one project, while a third team was working on a different project based on some other external or internal priority, the individual team components would be available at different times, which introduced delays. When the components would finally be completed, there was no clear understanding of who owned the integration process and testing, nor who was responsible for fixing issues discovered during integration. This resulted in further delays and less than desirable quality.

- **Staffing distribution.** Depending on focus of the management or incoming contracts, certain teams could grow to have seven or eight full time developers plus student developers, while other teams might shrink to only one or two developers. This created issues related to teams' ability to deliver their pieces of work in a timely manner. For example, the team with eight developers would complete their piece of the project and move on to another project, while the team with only one member would struggle to keep up.
- **Management focus-shift.** Managing a functional organization where each team behaves as an independent development group proved to be extremely challenging. The Lab Management frequently targeted individual teams to perform critical or time-sensitive work based on those teams' capabilities. When a new priority arose, these teams would be redirected, which inadvertently impacted other teams waiting on them for a deliverable.
- **Unified direction.** Frequently, project managers did not share information effectively with each other, which resulted in inconsistent directions to individual teams. As an outcome, teams were frequently attempting to satisfy different, and sometimes incompatible expectations.

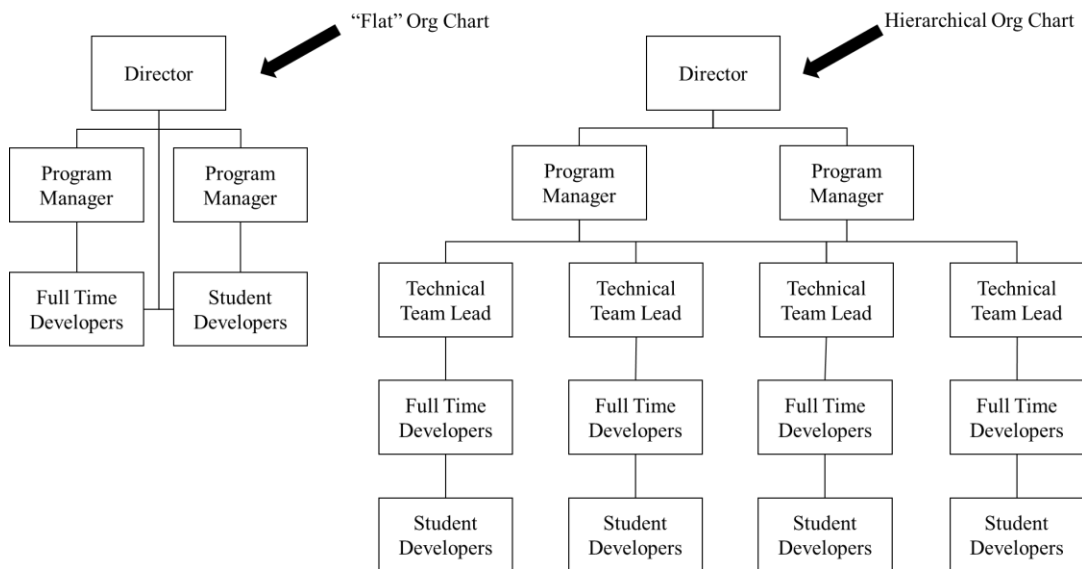
As developers and management grew frustrated by the above issues, morale began to decline. The dynamics within each team and even across teams also shifted and sometimes became toxic. Despite these challenges, the Lab continued to be successful and deliver high quality software. However, the stress that accompanied forced the CATT Lab to evaluate alternative organization and management.

## STAFFING SOLUTIONS

To remain effective, allow for growth, and to protect its employees from burn-out and frustration, the CATT Lab needed to reorganize and become more efficient. A small group of leaders and key developers in the CATT Lab were assembled to evaluate and recommend solutions. The first set of recommendations included the following:

**Introduction of hierarchy and structure.** In order to combat inefficiencies created by the existing flat functional CATT Lab organizational structure the Management Team introduced a new hierarchy as shown in *Figure 1*. The purpose of the hierarchy was to establish a clear chain of command for decision making. This new org chart decreased the number of communication channels by funneling

priorities and expectations as well as development progress through designated team leads and program managers.



**Figure 1 - CATT Lab “Flat” and Hierarchical Organizational Charts**

Initially, some developers were uncomfortable with their perception of communication restrictiveness within this model. However, as they developed a habit of utilizing technical team leads, the communication became more efficient and enabled developers to focus on tasks at hand and avoid excessive meetings. The Management Team also encouraged transparency on all levels along with “skip-level” management – where each employee is free to talk and collaborate with any other employee, including the management. Ultimately, staff came to love the new organizational structure as it reduced confusion related to expectations, roles, and communication workload.

**Staff diversity.** As the Lab structure changed, the hiring focus shifted from hiring internal graduating student candidates, to a broader search aimed at bringing outside talent. This hiring strategy led to a more diverse staff in terms of age, gender, technical expertise, problem solving capabilities, domain knowledge, and maturity. These new, externally sourced employees were able to recognize the Lab’s strengths – serving as a reality check to young employees who believed the grass was always greener on the outside. At the same time, these new employees were able to identify the Lab’s weaknesses from a more critical perspective – using their own industry experience to offer suggestions for improvement.

**Staff empowerment.** While the new structure was introduced in a top-down manner, staff were encouraged to provide suggestions for improvement of the structure and hiring process. As active contributors, the employees became more invested and more critical of the process, which ultimately helped to incrementally improve morale and provide employees with a sense of ownership unlike what they might have had before.

## WORK-FLOW SOLUTIONS

While the above changes helped resolve a good number of organizational issues with the staff, there still existed unaddressed challenges that affected the Lab's ability to meet deadlines, allocate resources efficiently, and remain flexible as an organization. These additional pain points remained:

- Constant shifts in priority that impacted schedules
- Difficulty of coordination across functional teams and leveraging resources
- Unbalanced teams
- Lack of understanding of overall CATT Lab product goals, deliverables, and schedules

Because of the newfound open communication lines, employees approached management to suggest considering adoption of the Agile software development process as a means to resolve these lingering pain points. After extensive research by management and developers, the CATT Lab formed an Agile Implementation Task Force whose focus was to provide guidance on how best to implement an Agile process that would meet the CATT Lab needs and minimize disruptions caused by the new process implementation.

One of the outcomes of this research was the realization that while the Agile process is a well-documented theoretical concept, the individual implementations of the Agile process vary wildly across industries and organization. With this understanding, the CATT Lab set out to establish a tailored version of the Agile process that would implement the relevant Agile concepts that addressed the remaining CATT Lab work-flow pain points.

## CATT LAB TAILORED AGILE PROCESS IMPLEMENTATION

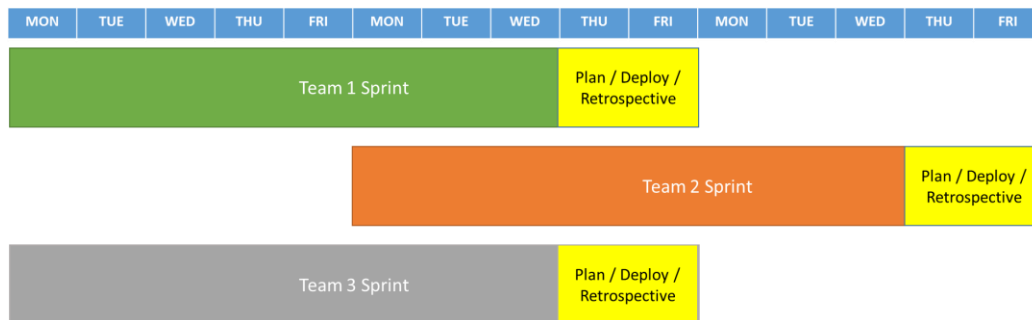
The Agile Process Task Force generated a proposal that addressed all major pain points and was implementable in a short period of time. The proposal was loosely based on Scrum and consisted of the following changes:

- *Cross-functional teams* – create three equally distributed cross-functional teams and three supporting teams
- *Two week sprints* – structure development in staggered two week long sprints
- *Sprint planning* – schedule work for sprints weekly and require management to be present to provide guidance and priorities
- *Physical team co-location*
- *Retrospective* – run retrospective meeting at the end of each team's sprint

**Cross-functional teams.** Each cross-functional team consists of subject matter experts including DBAs, back end and ETL developers, front end developers with web and analytics expertise, and quality control. The goal is to ensure that each team is capable of taking on any piece of work independently of the work being done in other teams and sprints. At the same time, exposure to multiple projects allows developers to learn other technologies and products, and to become more versatile.

Supporting teams include Information Technology (IT), User Experience (UX), and Customer Service. These teams were intended to operate outside of sprint schedule, but to be involved in sprint planning process to ensure that their priorities match the needs of development teams.

**Two week sprints.** All work is performed in two week chunks, called sprints, starting on Monday and ending on Friday of the following week. The goal is to ensure that work is broken down and scheduled in a manner that allows developers to remain focused and on schedule. As shown in **Figure 2** the start date of each sprint is staggered between each team to ensure that new priorities from management can always be scheduled quickly without disrupting the schedule and deliverables of other teams midway through their sprint. Each sprint ends on Friday, and it includes deploying completed work, running a retrospective, and breaking down and assigning the work scheduled for the following sprint. The deployments aren't always visible to the customer and could sometimes be only internal improvements. However, it is critical that each team deploys their work packages at the end of their sprint.



**Figure 2 - Sprint Staggering**

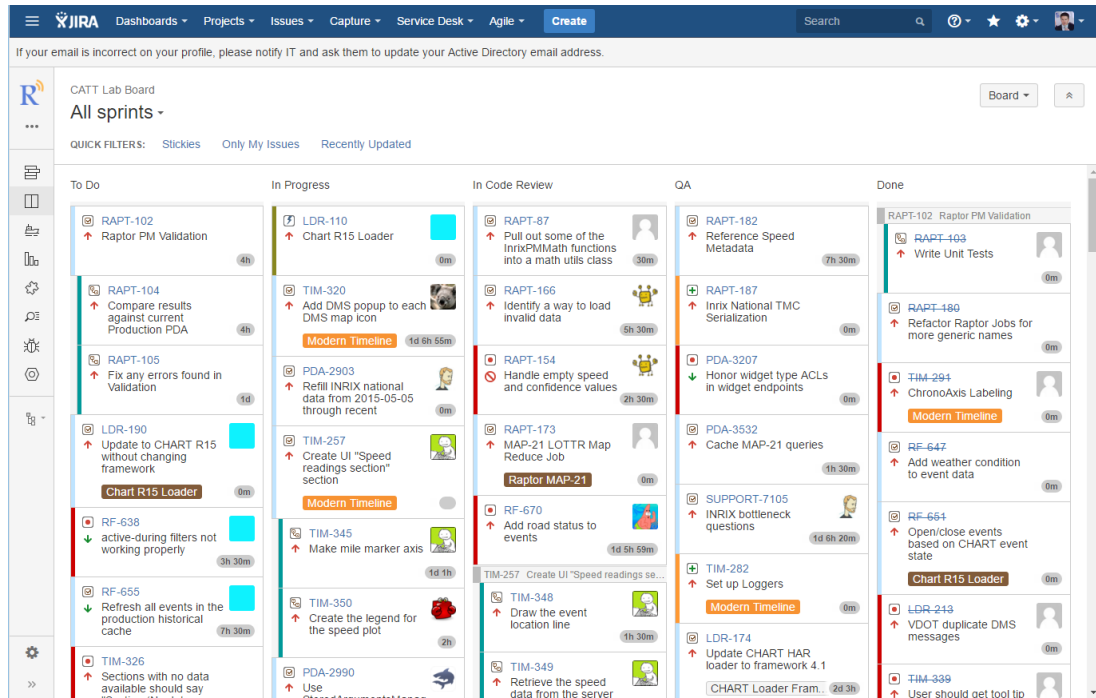
**Sprint planning.** Sprint planning meetings occur every Thursday morning from 8am to 10am in the designated “Scrum Room.” These meetings include management, all of the technical Team Leads (who fill the role of Scrum Masters), and supporting Team Leads. These sprint planning meetings represent the core of the process and are critical to success of the CATT Lab agile process. These meetings expose all known work across the Lab, and force Management to transparently prioritize and schedule work, expose how shifts in priorities by management would affect schedules, and simultaneously provide developers insights as to why and how priorities are established.

The meeting agenda for Sprint Planning is as follows:

1. **Team sprint updates** – each team lead goes through the list of their current sprint tasks and provides the following information for each:
  - a. Is the task on target to be completed, at risk, or will it need to be pushed to the next sprint?
  - b. Percent of expected effort expended on the task
  - c. Percent of task completed
2. **Team flow-up** – each Team Lead reports feedback from his or her team members regarding process, work progress, blockers, or any other significant events.

3. New work – anyone in the room is allowed to add new work to the board and explain where the work comes from and why it is important.
4. Update work priority – all newly identified work is then placed in one of the following bins:
  - a. *Ready to be scheduled* – this is work that is well defined and ready to be scheduled. This bin is further broken down into “ASAP,” “30 days,” or “60 days” according to priority.
  - b. *Blocked* – any work that is blocked internally or externally is placed into this bin to signify that it cannot be worked on until the block is removed.
  - c. *Needs requirements* – work that is not well defined is moved to this bin to signify the need for additional information.
  - d. *Needs architecture* – work that is well defined in terms of requirements, but its solution isn’t readily evident.
  - e. *Needs art* – work that requires user interface (UI) design by the UX team.
  - f. *Needs IT* – work that requires IT team resources and/or hardware.
  - g. *Proposals* – any potential work that requires estimation of effort and cost.
  - h. *Backlog* – if work is not high in priority, it is placed in the backlog which is reviewed occasionally to reevaluate priorities.
5. Plan next sprint – work that is ready to be scheduled is taken in order of priority and assigned to each team’s next sprint until each team is fully loaded with work for that sprint.
6. Plan future sprints – at least one additional future sprint is roughly outlined, but was subject to change in the following sprint planning meeting.
7. Management flow-down – the management provides any big picture changes in terms of new potential work, or anything else that may affect the development in the near future.

The development teams track all of the work in the Atlassian suite of tools including JIRA and Confluence. JIRA is an issue tracking system that allows developers to break down each task into components and assign those components to individual developers. **Figure 3** shows an example set of issues in a sprint tracked in JIRA.

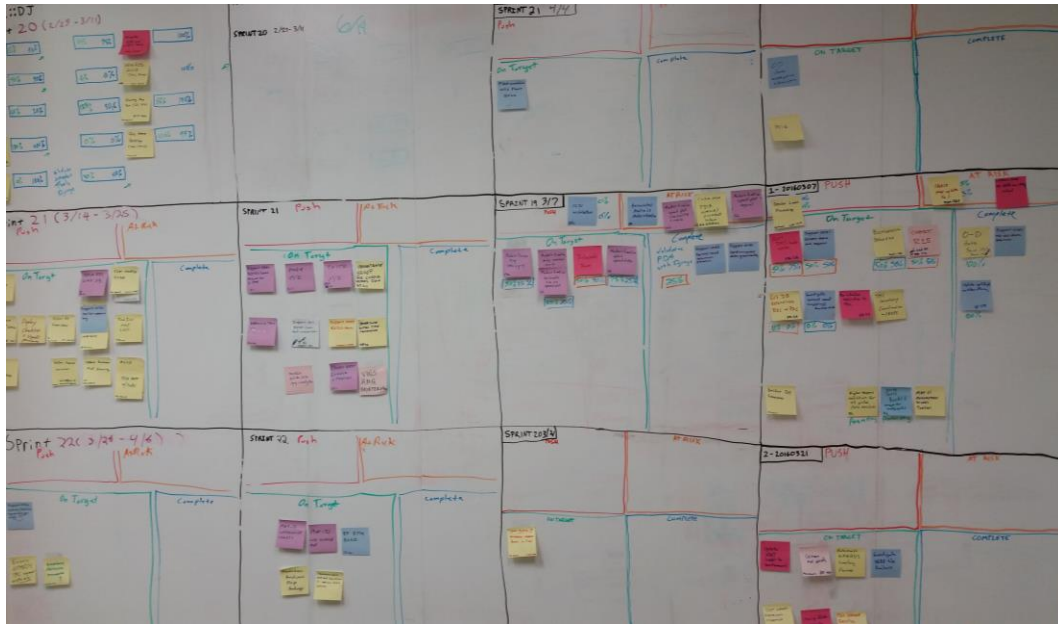


**Figure 3 - JIRA Sprint Tracking**

Even with the use of electronic tools like JIRA and Confluence, the CATT Lab desired to employ a more tactile and visual method for organizing and prioritizing work in their tailored agile process. The CATT Lab made use of nearly 400 square feet of wall space to their planning room to track, prioritize, and organize work using paper sticky notes. Each note represents high level work to be performed, and they are placed in appropriate locations on the wall: ready to be scheduled, backlog, blocked, under a given team’s sprint, etc. Each sticky note is then moved between these sections of the wall in real-time by Management and team-members. This visual management approach allows teams to see work in progress and understand complex information like processes, task relationships and risks related to a team’s ability to complete work on time. The tactile actions and the visualization help to minimize mistakes and facilitate team and management communicating during meetings.

**Figure 4** shows an example of the team sprint definitions using paper sticky notes. Each column on the board represents a team, and each row represents a sprint. Each cell is the split to classify work as “on target,” “at risk”, or “push” depending on the current progress. Also each sticky note has a percentage of effort expended and percentage of completion.





**Figure 4- Defining Team Sprints**

*Figure 5* shows an example of the work bins and sticky notes in each bin. During the meeting the team reviews each bin and moves tickets based on changes in their priority or newly acquired information.



**Figure 5 - Work Bins**

**Physical team co-location.** Agile team members are collocated in a single room to ensure team members are easily visible and accessible to each other. This approach fosters tight communication and building of relationships. As team members grow accustomed to each other, they are more likely to collaborate and ask for help when needed. They also become more knowledgeable about their team’s strengths and weaknesses and can leverage that information to perform work more efficiently.

**Retrospective meetings.** Every Friday at the end of each sprint, teams complete their deploys and meet to discuss the previous sprint and break down and assign the work for the following sprint. This meeting is specifically for teams and management attendees are there in a passive role to answer questions and provide high level feedback. However, the focus is on the team members, students, and transparency. The agenda is as follows:

1. What did we do well this sprint?
2. What did we not do well this sprint?
3. Other feedback.
4. Next sprint planning
  - a. Work breakdown
  - b. Work estimation
  - c. Work assignments

At the end of the retrospective meeting, the team members understand what made them successful in the previous sprint and attempt to replicate that in the future. They also outline their failures and take steps to prevent those from occurring in the future. Finally, they leave the meeting with a clear understanding of the upcoming sprint work and their role in the sprint.

## **LESSONS LEARNED**

**Living process and feedback loop.** Since the agile process was implemented at the CATT Lab 12 months ago, the process has evolved. The process was developed with an understanding that continuous introspection, self-assessment, and feedback loops are needed to improve the process. Each planning meeting is better than the previous planning meeting, and each retrospective meeting makes the team more effective than it was the previous week. During this 12-month implementation period the CATT Lab team further identified new challenges, and has taken steps to address each as outlined below:

**Work estimation.** Accurate work estimation is critical to effective scheduling and sprint planning. The teams consistently underestimated or overestimated their work which impacted their ability to reliably schedule work. To address this issue, the CATT Lab implemented the following changes:

1. Lunch and learn presentation – the CATT Lab staff put together a lunch and learn presentation that outlines several industry standard estimation techniques that can be used to better estimate work.
2. Planning poker – during the work breakdown portion of retrospective, the team members are asked to individually evaluate effort needed for each scheduled task by assigning it a value with one of the cards with Fibonacci sequence numbers from 0 to 55. Team members with the highest and lowest estimates are asked to explain the reasoning behind their estimates to the team – starting a discussion that leads to a better understanding of the complexities of the task, approaches to solving problems, etc. This crowd wisdom approach helps the teams reach more accurate estimates.

3. Performance measures – the CATT Lab tracks a new measure to define the amount of overestimation or underestimation that occurs in each sprint and to isolate potential issues that impact those estimates.

Despite these changes, teams still struggle with estimations and require more work to improve in this area.

**Aggressive demands from management.** In the effort to ensure progress, the management assigned more work than was feasible, and the developers felt that they were unable to effectively push back. Early sprints frequently resulted in no deployments and majority of work being pushed to the following sprint. This had a negative impact on developer morale as they felt that the “agile process” was just a disguise for the same old stream of constantly changing priority work. To resolve this issue, the CATT Lab implemented the following changes:

1. Sprints were extended to three weeks while attempting to keep the same work load as what was scheduled in two week sprints. This helped, but it also made the organization less responsive to changing priorities, and the extended work period had the unintended effect of adding more risk and uncertainty to the sprint as more work was taken on instead of an equivalent amount of work. After several sprints, the Lab reverted back to two-week sprints.
2. The planning process was changed to ensure that only the work that Team Leads were confident would get complete, actually gets scheduled. Team Leads are then held to a high standard with the expectation that *everything* scheduled in their sprint must be completed, unless it is impacted by an external blocker or another event beyond their control. This approach has had a major impact and has resulted in significantly more productive sprints.
3. The CATT Lab developed a new performance measure to track the amount of work that gets pushed to the next sprint as a percentage of overall work assigned. While the goal is to be at 0% at the end of each sprint, the percentages could be analyzed over time to determine patterns that may expose specific issues causing work to be pushed.

**Changing priorities.** The implementation of an agile process cannot affect external forces that might cause Management to shift priorities, but it has helped to make priority shifts significantly less disruptive given that the development teams are fairly protected in their sprints. The impact has mainly been absorbed by the Team Leads who personally take on any incoming work to ensure that the remainder of the team continues uninterrupted. This approach works, but as Team Leads become overloaded, they may drop or forget about certain tasks, therefore introducing risk. To address this issue, the CATT Lab is implementing the following changes:

1. Proactive communication – the management is attempting to be as proactive as possible and bring up new work in early stages (even if the new work has a minimal probability of occurring) so that placeholder tasks can be created in future sprints and expectations can be managed. However, sometimes customer demands dictate immediate priority changes, which still presents a challenge.

2. Reserve Team Lead capacity – the Team Leads reserve some of their time for unplanned work and attempt to take on that work instead of disrupting the rest of the team.
3. Reduce team size – adding a fourth team has allowed the Team Leads to work with smaller teams and open up some additional personal capacity to address unplanned work.

**Overly diverse sprints.** Cross-functional teams are capable of performing work on many different tasks. However, scheduling too much diverse work within a sprint causes excessive context switching for individual developers, thus affecting productivity. Even going from sprint to sprint, if tasks are significantly different, additional time is spent in spin-up for each family of tasks. To address this issue, the CATT Lab implemented the following changes:

1. Assign more homogeneous work for teams for a single sprint whenever possible.
2. Designate each team as responsible for a set of logically related projects so that teams operate on familiar code bases from sprint to sprint, even if they are required to work on a different set of tasks, while still maintaining cross-functional nature of the teams.

**Performance measures.** There is an old saying, “what gets measured gets done” and the same holds for agile process. Without metrics, it is difficult to discern level of success or failure. The development team may think that they work hard and accomplish amazing amounts of work, while the management keeps seeing tickets getting moved to “At Risk” or “Push” sections of the scrum board. Conversely, the management may be pushing certain work, when that particular work does not lay on the critical path. The CATT Lab developed several performance measures that the team tracks and uses to guide process and skills improvement:

1. Percent of work not carried to the next sprint – at the end of each sprint, the team tracks the number of sticky notes that did not have to be pushed to the next sprint. The goal is to achieve 100% completion rate.
2. Number of deployments – this metric measures how frequently sprint ends with a successful deployment.
3. Number of pull requests posted by Team Lead – a “pull request” is a developer generated notification that code for a particular feature is completed. Other developers are invited to review this feature and merge its source code into the main source code branch. This metric is meant to measure effectiveness of reducing team sizes with an addition of the fourth team. The theoretical expectation is that as each Team Lead manages a smaller team, they are able to perform more hands-on development which will result in an increased number of pull requests.
4. Accuracy of estimates – this metric quantifies accuracy of teams’ estimates in order to identify specific issues and address those in a meaningful manner.
5. Number of tickets resolved by developer and Team Leads – this metric allows tracking of individual technical performance.

## CONCLUSION

Organizational change can be stressful and complex. However, when organizational change comes from within as a reaction to the success of an organization and employee concerns, the change can be more palatable and even welcome. The CATT Lab's implementation of a tailored agile process along with adapting hiring processes and organizational restructuring addressed many pain points and set the Lab up for long term viability. The keys to the CATT Lab success are:

1. Adopting and tailoring a process suited to the Lab's mission and objectives
2. Empowering individuals and allowing them to own portions of the process.
3. Establishing a clear hierarchy
4. Introspection and continuous self-improvement
5. Measuring performance and setting targets

## REFERENCES

- "The Agile Manifesto for Software Development - Agile Alliance | Agile Alliance." *Agile Alliance*. Web. 16 Mar. 2016. <<https://www.agilealliance.org/agile101/the-agile-manifesto/>>.
- "Helping Teams Build What's next." *Atlassian*. Web. 16 Mar. 2016. <<https://www.atlassian.com/>>.
- "Pull Requests." *Atlassian Git Tutorial*. Web. 16 Mar. 2016. <<https://www.atlassian.com/git/tutorials/making-a-pull-request/>>.
- "CATT Lab." *CATT Lab*. Web. 16 Mar. 2016. <<http://www.cattlab.umd.edu/>>.
- "Planning Poker." *Planning Poker*. Web. 16 Mar. 2016. <<https://www.agilealliance.org/glossary/poker/>>.
- "Learn About Scrum." *What Is Scrum? An Agile Framework for Completing Complex Projects*. Web. 16 Mar. 2016. <<https://www.scrumalliance.org/why-scrum>>.
- "RITIS." » *CATT Lab*. Web. 16 Mar. 2016. <<http://www.cattlab.umd.edu/?portfolio=ritis>>.