VERSIONONE

KANBAN FOR AGILE TEAMS

Mike Cottmeyer and Dennis Stevens

EXECUTIVE SUMMARY

Kanban is an emerging set of management practices for software development teams that was derived from Lean Manufacturing, the Toyota Production System (TPS) and Goldratt's Theory of Constraints. Kanban has recently become popular with many project teams because of its ease of implementation, use of visual controls, ability to accommodate a wide variety of organizational design patterns, integration of stakeholders and relentless focus on the continuous delivery of value. Many development teams have found success with Kanban when more mainstream agile practices did not yield acceptable outcomes.

> While there are meaningful differences between agile and Kanban, many teams will find that blending the two approaches can create tremendous value for their organization.

Organizations that are interested in adopting or improving agile methods should evaluate the underlying principles behind Kanban and how the principles can work together with more traditional agile methodologies. While there are meaningful differences between agile and Kanban, many teams will find that blending the two approaches can create tremendous value for their organization.

This paper explores the common tools, practices and philosophies of Kanban and examines how an agile team can leverage Kanban to create a more effective product delivery organization.

CONTENTS

INTRODUCTION

Kanban is a Japanese term meaning "visual record" or "card." Kanban derives its name from the use of a card system to manage the flow of work through the development organization. In Lean speak, Kanban means signal cards or a signaling mechanism used to represent a piece of work in process. The number of cards in circulation is equivalent to the capacity of the system. In this way a Kanban system is designed to pull work when there is capacity to process it. This mechanism is known as a pull system because new work is pulled into the system when there is capacity to handle it, rather than being pushed into the system based on demand. A pull system cannot be overloaded, providing the capacity as determined by the number of signal cards in circulation has been set appropriately.

Kanban is straightforward to implement. The three steps are to visualize the process, limit work in process and then manage the Lead Time. Lead Time is the average time it takes for one unit to go through the entire process – from start to finish – including the time waiting between sub-processes. The Lead Time is managed by addressing constraints in the process, focusing people on creating flow, planning for variability and establishing context - specific process policies. This paper will discuss each of these practices and explain how they apply to a software development team.

THE KANBAN BOARD

The Kanban Board is a tool that helps the team visualize the process, and shows the flow of value through the development process. If you have seen a Scrum task board you are familiar with the general concept behind the Kanban Board. There are sequenced columns that represent the various states a work item can exist in during the development process. As work progresses through the development lifecycle, the card is moved from one state to the other, until it has been accepted by the customer.

A simple Kanban Board might have columns titled Next, Analysis, Development, Acceptance and Production. At first glance, the Kanban Board resembles a waterfall process. But in practice, the Kanban Board explicitly avoids the problems with a waterfall process by enforcing small batch sizes and incremental development. Each column will have a Work in Progress (WIP) limit established with cards that are associated with user stories. As a card is completed in one column, it moves to the next. As a card moves, it creates an open space on the board in its current column representing available capacity. The team will then pull a completed card from a previous column. As spaces in the Next column become available, the product owner will prioritize a card for that column.

VISUAL CONTROL SYSTEM

A Kanban Board is a Visual Control System. A Visual Control System uses visual signals to control an activity or process. One example of a visual control system is the silhouettes in a tool rack to indicate where a tool should be stored in the workshop. Looking at the work area, a person can quickly identify where a specific tool is and audit the workshop for any missing tools. Another type of Visual Control is an assignment board that may use a magnet that is red on one side and green on the other to show if someone is in the office and which item they are currently working on. Visual controls can also be used to manage or control a process.

Visual Control Systems are valuable in changing behavior because they display status in an easy-to-see format. This ensures that everyone has a shared understanding of work status and process constraints. Any problems are visible to everyone, providing immediate feedback so that corrective action can be taken. A Kanban Board is a Visual Control System that delivers transparency into work status and process constraints. This transparency is a key to achieving organizational change. It also enables the team to self-organize - assigning their own tasks and moving user stories across the board without direction from a project manager.

WORK IN PROCESS LIMITS

Kanban is more than just a way of visually tracking progress at the user story level. In Kanban, the team sets work in process limits for each column of the Kanban Board. Why is limiting WIP important? Little's Law explains this. Little's Law is an equation that relates Lead Time, WIP and Cycle Time (the average time between deliveries of a unit). In its simplest form it states that Lead Time = WIP / Cycle Time. This means the quickest way to reduce the Lead Time is to reduce WIP. Explicitly limiting WIP is a key difference between the Scrum task board and the Kanban Board.

One common challenge in Scrum is late delivery within the sprint. Late delivery means that all the backlog items were in process during the sprint, but were only fully done and accepted in the day or two prior to sprint closeout. Late delivery introduces risk, tends to destabilize velocity and results in delayed value delivery to the customer. Scrum addresses this by encouraging team members to work together on only a few backlog items to completion before starting new work.

Kanban takes this implicit guidance and makes it explicit by setting a limit on the number of work items that can be active at any one time. By limiting the amount of work that is in process, the team is forced to focus on getting backlog items fully done before new work can start. Adding WIP limits implements pull-along with the benefit of making sure that demand never exceeds capacity. If there is no explicit limit to WIP and no signaling to pull new work through the system, then it is not a pull system.

WIP limits should be determined by the team in collaboration with management. This helps optimize the flow of value and reduces the number of intermediate work products. This collaboration also gets management aligned with protecting the WIP limits.

CONSTRAINTS

WIP limits not only help reduce Lead Time; they also highlight the constraints in the system and effectively restrict the amount of work that can be in any one state at any one time. When work cannot move forward on the board because the WIP limit has been reached in the next state, it makes the current constraint on the system highly visible.

The Kanban Board forces the team to stop taking on more work until the problem with the constraint is fixed. By focusing all our attention on the constraint, the team works together until the impediment is removed. WIP limits encourage everyone to work as a team and prevent any one individual from getting too far ahead of everyone else. Kanban elevates awareness of constraints and forces the team to address them before they can bring additional work into the queue. While Scrum has retrospectives at the end of an iteration to address these items, Kanban explicitly points out constraints in real time and encourages the team to address them as they arise.

Because WIP limits also provide visibility into quality problems and make explicit the impact of defects on Lead Time, Kanban encourages higher quality. The combination of improved flow and better quality helps to shorten Lead Times and improve predictability and due date performance. Establishing a regular release cadence and delivering against it consistently helps build trust with customers and across the value stream with other departments, suppliers and dependent downstream partners.

FLOW, NOT UTILIZATION

One typical conflict in organizations is the drive to ensure full utilization of "resources" – people that actually do work in the organization. The columns on the Kanban Board do not represent strict hand-offs between team members, but the feature's state during that time in its development lifecycle. While Kanban teams can accommodate a reasonable division of labor, Kanban teams are still typically made up of specializing generalists. Having specializing generalists on the team

reduces skill-set dependencies and levels the flow of work across the team. Exposing the intermediate states highlights opportunities for improving the skills or makeup of the team.

Kanban, like Scrum, leaves it to the team to decide how work moves through the system and doesn't explicitly assign work to an individual with a specific skill-set. The Kanban Board does acknowledge the reality of intermediate development states explicitly controlling the amount of work in each state. In Scrum, lack of clarity into the team processes can encourage a focus on team utilization. With Kanban, the focus of management, and the team, becomes the flow of work, not the utilization of the team.

PLANNING FOR VARIABILITY

User stories moving through the development process will not all be the same. The level of effort for Analysis, Development and Acceptance will not be uniform. This can lead to idle resources while waiting for an upstream activity to complete. Kanban provides three useful practices to address the impact of variability on the workflow.

The first practice involves queues between columns. In this case, there needs to be visibility into an amount of work that has completed onstate and is ready for the next state. For example, you may want to know how many items are through analysis and ready for development. That way, when the developer completes a task they can pick up the next item for development. A column showing work that is in the ready queue helps maintain flow while also pointing out a downstream constraint before it has led to a delay.

The second practice is to use generalizing specialists effectively. If a developer completes an analysis item, they can either take it into development or start another analysis item. This decision can be made based on the best use of the developer's time and the state of work against the WIP limits in different columns. Daily planning meetings in Scrum facilitate mature teams having conversations regarding queues and prioritization of the next work item. Again, Kanban makes these decisions explicit and visible to the entire team – including management.

The third practice addresses variability in demand. If a new demand comes into the development team from some market requirement or from a defect that has escaped into production, it gets added to the Kanban Next column as a high priority item. In Scrum, this would either interrupt the sprint or have to wait until the next iteration. In Kanban, the defect can get pulled into development at whatever column is appropriate as soon as a spot opens. This helps minimize the impact on work in progress from production defects and expedited user stories.

NO ESTIMATES OR ITERATIONS?

While agile is fundamentally designed to accommodate change, the idea of an iteration implies some degree of certainty about what it is you plan to build, even if that certainty is only two to four weeks into the future. Traditional agile teams establish iterations, typically of 1 – 4 weeks, and then spend effort estimating and prioritizing the backlog to plan the iteration. In some environments customers can't wait until the iteration boundary to introduce changes, or requirements cannot be neatly fit into a consistently sized window of time.

This understanding has led some teams to abandon the iteration alltogether and now consider planning at iteration boundaries to be a wasteful activity. These teams have opted toward working directly from the prioritized product backlog and only bringing new work to the team when the WIP limits allow a new feature to begin. Agile introduced the idea of small batch sizes by limiting the amount of work that could be pulled into the iteration. Kanban teams take this approach to the extreme and reduce the batch size at any one time to that of a single user story.

Some Kanban teams have also suggested that estimating is a wasteful activity that provides no value to the customer. They have established a perspective that if work is worth doing then the work should be done. In some extremes, teams had been spending 30-40% of their time providing estimates to the business for iteration planning, internal cost transfers or performance tracking that provided no external benefit. These teams have benefited from transferring this effort in estimating to productive effort. Kanban does not call for the elimination of iterations or estimating. With Kanban, iterations and estimating are optional. Unlike Scrum, Kanban based teams should do these only if there is a need for it in that team's context for that specific type of work.

The teams that have eliminated iterations no longer have iteration based retrospectives. They conduct periodic operational reviews to assess team performance, communicate with management and evaluate the overall health of the system. Teams choose appropriate boundaries for traditional retrospectives but they are not necessarily tied to the operational review or any predetermined time-box.

VELOCITY VERSUS LEAD TIME

Agile teams become predictable by stabilizing the size of the product backlog and establishing a consistent Velocity over time. We can assess the delivery date of a fixed scope product backlog by dividing its total size by the Velocity of the team. Likewise, we can use Velocity to make trade-off decisions when trying to manage to a fixed delivery date release. Without the iteration, there is no longer a fixed period of time from which the team can measure team Velocity.

Kanban teams replace Velocity with the notion of Lead Time. Lead Time measures the total elapsed time from when the feature was started until it is marked complete and accepted by the customer. Lead Time can be useful for predicting delivery and making both short-term and long-term customer commitments. Because features are independent and not allocated to a fixed-scope sprint, the team can always work on the highest priorities of the business. The team is able to set service level policies regarding the Lead Time for an item from the time it enters the development process.

Many Kanban teams track the Lead Time of similarly sized stories, making a distinction between the lead-time of a one-point story versus the Lead Time of the other possible story point values. If the team becomes proficient at breaking down user stories into similarly sized increments of work, estimation is no longer even necessary and Lead Time becomes the only metric needed to measure the delivery capability of the team.

TIME TO USE KANBAN?

Kanban is just about managing workflow. Initially, it doesn't replace anything the organization is doing. What it does do, however, is drive change. In Kanban you start with whatever process you have, visualize it, introduce WIP limits and then evolve from there. Kanban is not a development methodology. It is not a replacement for Scrum, XP, AUP or DSDM. Kanban is a powerful set of practices that can be used to complement your current agile practices or when your current processes stop making sense.

Consider these examples: Have you ever tried to talk about planning in two week iterations with a team that is responsible for production support? The idea that you can't add anything to the sprint once it is started is going to fail - you'd probably abort every sprint. How about a mature product team that is iterating an existing product?

Does it necessarily make sense to stop every two weeks for a few hours to plan? To retrospect? Probably not. If requirements are generally understood, the team talks over lunch and they just know how to get things done.

How about a team that suffers from late delivery in the sprint? A Kanban Board can keep them focused on continuously delivering value, even if they don't do away with the iteration boundaries. What about a new team that isn't ready to self-organize or self-manage? A Kanban Board can give the manager a tool to help the team learn how to self-organize and self-manage. Kanban can be used as a replacement for processes that aren't working or as an addition when a little bit more control is required.

KANBAN IN THE ENTERPRISE

Just like Scrum and most agile practices, Kanban is useful at the small team level. But most teams that are struggling with agile development are struggling with environmental problems. Kanban becomes really interesting at this higher level. Unlike Scrum's Chicken and Pigs mentality, Kanban involves management in setting the WIP levels and service level policies and provides constant transparency into the status of the work. Kanban scales from small teams doing user stories to multiple teams doing projects to the enterprise portfolio level to the Enterprise Value Stream. What if you could have an enterprise Kanban? A portfolio Kanban? A project Kanban? A team Kanban? As you go deeper into the organizational hierarchy, the WIP limits at the enterprise level force value from the portfolio. Portfolio limits force value from the projects. Projects force value from the teams. Imagine a system with a predictable enterprise Lead Time all the way down to team level Lead Time.

Think about the enterprise as a series of capabilities that together form a system, which can be managed and subordinated to the constraint. Now that is exciting! Now you can focus your enterprise agile adoption dollars at the constrained teams and resource groups to get maximum value from your investment.

SUMMARY

Many software teams will come to use Kanban from agile. Some teams will adopt Kanban outright, never having passed through Scrum. Many teams will use Kanban at the project, portfolio or Enterprise Value Stream level while continuing to use Scrum at the small team level. What is important is that the organization finds ways to do what makes the most sense for its context. Visualizing the work flow, setting WIP limits and focusing on managing flow and reducing lead times allows organizations to drive effective organizational change that results in improved economic outcomes. Extending Kanban beyond the small team to the project, portfolio and Enterprise Value Stream level is an exciting frontier that is being explored by leading enterprises to align efforts and gain a competitive advantage in their markets.

AUTHORS' BIOGRAPHIES

Mike Cottmeyer is an independent agile coach that provides agile training, agile coaching, and agile transformation services designed to help pragmatically, incrementally, and safely introduce agile methods into any sized organization.

Prior to becoming an independent coach, Mike was the Regional Vice President of Pillar Technology Southeast. He was responsible for new client acquisition and engagement design. He stayed involved as a hands-on agile coach and trainer for larger and more complex agile initiatives. His specialty was and is large programs in complex multi-methodology environments.

Prior to joining Pillar, Mike was an agile coach and trainer with VersionOne where he helped customers lead large-scale agile transformations and adopt agile project management best practices. Before joining VersionOne, Mike was a Senior Project Manager for CheckFree Corporation where he led a portfolio of agile projects for their online banking and bill payment business unit.

Mike is a certified PMP project manager and a certified ScrumMaster. Mike was involved with the creation of the DSDM Agile Project Leader certification, and holds this certification at the Foundation, Practitioner, and Examiner levels. Mike was named an honorary member of the DSDM consortium and served on the board of APLN and the Lean Software and Systems Consortium. He currently co-leads the PMI Agile Community of Practice.

Dennis Stevens leads Synaptus, an organization committed to enabling the agile enterprise. For over 25 years, Dennis has been passionate about helping organizations deliver technology that makes a difference for their business. For 15+ years, he has served in enterprise project management and leadership roles. He advocates blending responsible Lean and Agile Software Project Management with traditional project management practices in the Enterprise. He attended FSU on a violin scholarship, served in the USMC, helped coach the 2008 12U Girl's AAU National Champions, and has a degree in Organizational Psychology and Development.

Dennis has been frequently published, including: "The Next Revolution in Productivity" Harvard Business Review, June 2008; "Re-thinking the Agile Enterprise," Cutter Consortium Executive Report, July 2009, "Cloud Computing: Where to Start When you have Already Started", Microsoft Technet, June 2010; and, "Introduction to the Agile BABOK", IIBA, June 2010. Follow Dennis on twitter at @dennisstevens or his blog at http://www.dennisstevens.com.