ESSENTIAL UPSTREAM KANBAN

PATRICK STEYAERT

LeanKanban
UNIVERSITY
PRESS
SEATTLE, WA

Essential Upstream Kanban

ISBN: 978-0-9845214-7-0
First digital version, 23 September, 2017. This version 23 September, 2017.

Essential Upstream Kanban can be downloaded via leankanban.com/upstream. Additional print copies of this and other Kanban publications can be purchased via shop.leankanban.com.

Visit edu.leankanban.com for a list of accredited Kanban classes, coaches, and trainers, and information about becoming a Kanban coach or trainer.

Visit services.leankanban.com or email sales@leankanban.com for information regarding private training or consulting for your organization.

Contents

Foreword

We tend to think of kanban systems and their WIP (Work in Progress) limits as offering relief from overburdening. At an individual or team level a WIP limit takes pressure off individuals and allows them to focus. The result is higher quality work produced with a greater pride of workmanship. People feel better about doing the work and customers notice an improvement in quality. A reduction in rework due to higher quality can also result in lead time and productivity improvements. However, these benefits are primarily internal

It isn't until we relieve the system of overburdening by limiting WIP across a whole workflow with a kanban system that we see dramatic improvements in economics and customer satisfaction. Relieving a workflow of overburdening dramatically reduces waiting times while work requests queue for attention. A kanban system spanning from the point of commitment to delivery dramatically reduces lead times and provides a significant improvement in predictability. Full kanban systems provide better customer service and improved customer satisfaction – these benefits are external. The observable agility of service delivery is tangibly improved.

In the early years, Kanban focused on workers providing IT services on fully committed work orders. The focus was on faster, more predictable delivery. A great deal of the improvement came from the practice of deferring commitment – because downstream work was limited by the Kanban system. The implication was that upstream work remains uncommitted and, therefore, optional. The introduction of Kanban to improve delivery has the side effect of introducing a requirement for a strong "triage" discipline. It

becomes necessary to develop capability to answer questions like: What should we work on now? What can wait until later? And if later, when? What should we discard altogether? For many professional services, making the action of triage an explicit process in the organization is challenging. Individuals have never had to face the difficult decisions of what, when, and how many to work on, and if it is even a good idea at all.

At its core, industries where the goods are intangible in nature such as advertising, market research, video editing, software development, legal, human resources, and accounting, struggle with the concept of limits. It is all too easy to say "yes", to make a commitment, to start something. The bigger challenge becomes finishing something and doing so with predictability. The more you say "yes", the more you start, the less gets finished, the longer everything takes -- and any predictability on delivery time dissolves away. But there is still relief from over-burdening when a limit is set of work requests in progress and that benefit is tangible to those doing the work.

If relieving downstream delivery workers from overburdening made sense, did it also make sense to relieve upstream creative workers, those who must discover the concepts and ideas, from similar overburdening?

We began to see "upstream kanban" systems appear in 2010 with examples emerging independently in Paris (2 separate unconnected examples) and Belgium (with Patrick Steyaert.) Patrick has continued to focus his professional life and work on Upstream Kanban and refine his ideas with Discovery Kanban and Customer Kanban in the 7 years since then. He is a recipient of the prestigious Brickell Key Award from the Lean Kanban conference series for his contribution.

Upstream Kanban, and its variants, ask us to model creative workflows, ideation and business case development processes and provide a WIP limit to relieve those workers and the upstream creative system from overburdening. However, it emerged that what was more important with upstream creative activities wasn't limiting the number of options in development but to signal the replenishment of ideas at a given stage of elaboration. It was necessary always to have enough options from which to choose, sufficient choice to make the best commitments and facilitate the triage process.

By introducing minimum limits to signal replenishment, Upstream Kanban introduces the 2nd kind of kanban signal card that exists in the physical, tangible goods, implementations of the concept. There are kanban to limit inventory and prevent overburdening of a system, and there are kanban to signal replenishment – in a physical environment, these signal the "just-in-time" supply of components to be fetched from a warehouse and delivered to the point of production, usually as a batch. So, Upstream Kanban completes the implementation and adaptation of kanban systems into professional services, intangible goods environments.

At each stage in the upstream discovery or ideation process, there is the opportunity to discard a bad idea: too expensive; too hard technically; will take too long; or limited value or generally undesirable to consumers. This concept is known as having "embedded options." Upstream Kanban isn't so much about "managing flow" as seen with downstream, Delivery Kanban systems. Instead it is about marshaling options – having enough choices at the right time, without overburdening the system and creative workers who generate those options by asking them to maintain too many of them.

Upstream Kanban is still an emerging field. It's an active area of innovation and new intellectual property development as we learn more about how to help creative, business people develop their ideas. Meanwhile, we see the rapid growth of Kanban beyond its initial roots in IT services and software development. Upstream Kanban gives us the breadth to offer a better way of working and managing work to entire enterprises. Patrick Steyaert is at the forefront of this push. Expect this guide to evolve and grow over the next few years.

Enjoy Upstream Kanban. Don't be scared to try it and apply it. Don't be put off by the rapid emergence of new ideas and guidance. Upstream Kanban is proven and tested. These aren't thought experiments you will read in these pages. These are working concepts, refined in the field and established as "fit for purpose." And if you find these ideas helpful, share them, share your story, get involved in our community and help us refine our ideas and spread our knowledge.

Best wishes marshaling your creative ideas with Upstream Kanban!

David J. Anderson
Sequim, Washington, USA, April 2017

Preface

The purpose of a business organization is to create value for its customers through meaningful work. In a fast paced world, this proves to be quite a challenge.

Ideas for fulfilling customer needs can be generated much faster than they can actually be realized. This is the source of much tension between the organization and its customers, but also, in the organization itself, between those that represent the customer and those that deliver to the customer. Business agility requires quite the opposite. It requires that those that suspect needs closely collaborate with those that fulfill needs. Central to the success is the ability to tune the fast paced idea/request/requirement generation to the slower paced idea/request/requirement fulfillment.

In the old days, decisions only needed to be taken once in a while; we all remember the annual business cycles linked to a yearly budgeting exercise or the annual release cycle. Decisions were taken infrequently and in isolation. This often resulted in customers or customer representatives that were pushing (big batches of) requests to an order-taking delivery team that was pushing (big batches of) products back to customers that were not ready to receive or do something meaningful with it. The collaboration between the end customers, the representative of those customers (such as a Product Manager, Product owner, or Key business user) and the delivery team was pretty hands-off.

Today, as the pace of business is picking up and uncertainty is rising, decisions need to be taken much faster and in joint collaboration. Many organizations already have made – or are making – the transition to (a combination of) agile development, lean kanban,

and continuous delivery, thereby increasing their speed and frequency of delivery. What they find out, however, is that faster and more frequent delivery in itself is not sufficient. Unless proper mechanisms are put in place to tune the speed of demand with the speed of delivery, a tension will remain. Customer Kanban together with Upstream Kanban are mechanisms to overcome this tension and engage the whole organization (not just the delivery team) in the move towards Business Agility.

Discover a unique approach, called Customer Kanban, and learn how to start thinking in terms of Business Agility (on top of your agile development).

Conventions

Kanban (the word) appears many times in this book, but readers will notice it is not always capitalized. The Kanban Method was so named in 2007 following presentations of the management approach that David had been using at Microsoft (Anderson, 2005) and Corbis, and the formation of a community around these and similar ideas. The Kanban Method, Kanban, or Kanban community is always capitalized in the text, when used in this sense.

However the Japanese word "kanban" (meaning "sign," "signal card," "tally," or "large visual board") has been used in the context of process definition since at least the 1960s, when Toyota named the systems they had been using to limit **work in progress** in their manufacturing plants "**kanban systems**" (Shimokawa, 2009). Such systems were just one of the many threads of inspiration behind the Kanban Method, although it is how the name for the method arose. Thus kanban is *not* capitalized in this text when referring to kanban systems, to **kanbans** (the physical cards or virtual signals that kanban systems use to control work in progress), or to **kanban boards**.

An Illustration

Suppose that you are working in an organization where you have multiple initiatives (e.g. customer projects, product development, business changes, customer requests, …) going on. Every so often work finishes and capacity becomes available. At the same time the customer is requesting new work to be performed. In the ideal world, we can perfectly match the needs of newly arrived work with the capacity that has become available. Work is completed and the required skills/capacity are available again exactly at the time when new work needs to be started. The pace of completing work is the pace of arrival of (new) work. But … we are not living in an ideal world. Work arrives when there is no free capacity or the right team members are not ready to start; or, capacity becomes available when no new work has arrived or the customer is not ready to start. Often the customer is pushing new requests while the value of old requests has not been tested. Neither the organization nor the customer in isolation can overcome the above hurdles. Both need to be engaged.

In this article we will discuss Customer and Upstream Kanban as a way to match the speed of demand with the speed of delivery. We will use the case of an IT maintenance team that has started with Kanban to improve their delivery capability. We chose an IT maintenance team as case because in the simplest way it shows why just improving the speed of delivery is not sufficient (and that there is a difference between plain agility and business agility). The fact that the team already started with Kanban (as opposed to e.g. Scrum) comes in handy as the proposed solution – Customer Kanban – builds on Kanban and the principles of flow thinking.

Mind, however, that the obstacles that motivate Customer Kanban also are encountered in organizations that are, for example, doing agile development with Scrum or that are transitioning from traditional to agile project and portfolio management.

System Kanban Success

Lead-times get shorter and more predictable and delivery rate increases.

The IT maintenance team that we use as a case study handles change requests and also delivers small projects across Enterprise Resource Planning (ERP), Business Intelligence (BI), Electronic Data Interchange (EDI), and Java applications. Historically the team has been coping with a persistently large backlog (+80 items, see Figure 1) and long, unpredictable lead-times. End-to-end customer lead-times (the time from "request" to "ready to deploy") have been varying between less than 1 week and 30+ weeks. Also system lead-times, i.e. the time from "Ready to develop" to "Ready for UAT (user acceptance testing)", have been long (+13 weeks) and variable.

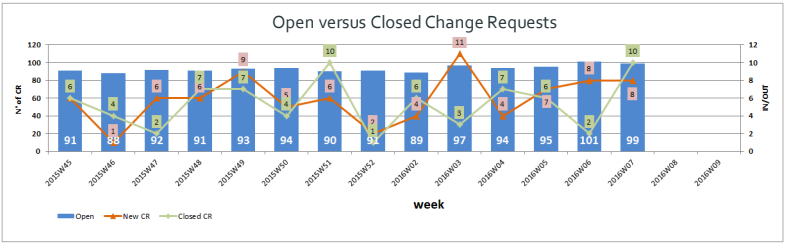

Figure 1: Demand versus Capability

To better serve their users, the IT maintenance team has started a Kanban initiative. The initial focus of the initiative was on the Downstream/System Kanban (see Figure 2). The purpose is to improve the flow of work and specifically to reduce the time that work is in progress and to increase the amount of requests that can be fulfilled. Agility of the team increases as time in progress is reduced and the team starts collaborating more.

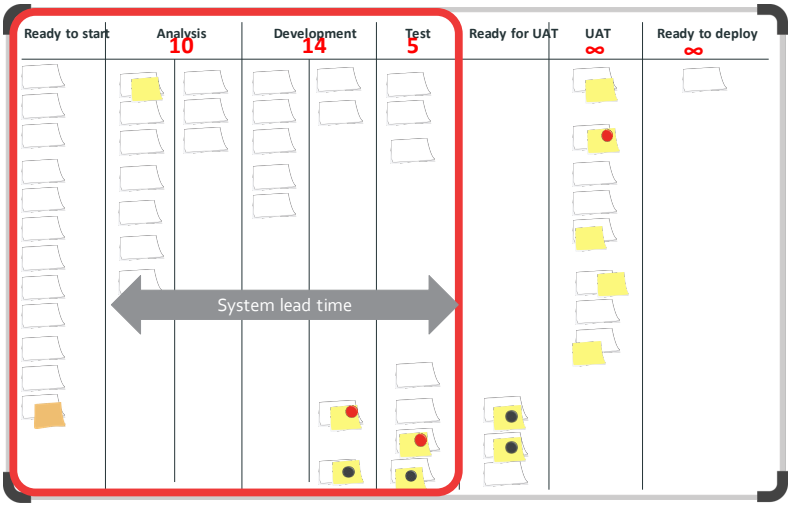

Figure 2: Downstream/System Kanban

Often Kanban is associated with visual management. Figure 2 shows the Kanban board used by the IT maintenance team. While Kanban boards by themselves can have an added value, they are only the tip of the ice-berg. For the purpose of this article we will make a distinction between visual management through a Kanban board (often also referred to as proto-kanban) and true Kanban systems.

The purpose of a true Kanban system is to improve the flow of work. Specifically the flow of work in a team – or across a group of teams – that is/are providing a service to a customer (an internal customer in our IT maintenance case). For reason that should become apparent, we will refer to these kanban systems as System Kanban.

System Kanban starts from the point of committed work (i.e. a work item that is "Ready to start"), and ends at a point just before delivery to the customer (typically a work item that is "Ready for

Acceptance"). These two points form the boundary of System Kanban in which the team can self-organize to create flow.

System Kanban implements flow by allowing workers to pull work rather than having work being pushed onto them. This is beneficial for the people that do the work in terms of creating a meaningful, collaborative work environment. It also has value to the customer as it results in a more reliable and attractive, fit-for-purpose, service. Pull is either implemented by means of a work-in-progress limit (WIP limit), or sometimes also by making use of explicit Kanban tokens.

While Figure 3 below shows both WIP limits and explicit Kanban tokens at the same time, in practice teams will choose one of both. Essentially they are equivalent, but for the purpose of this article it is worthwhile to point out that they emphasize a different aspect of creating flow. WIP limits emphasize the importance of constraints to enable the collaboration that is necessary for a steady flow. In other words, the team needs to work together to keep WIP low and lead-times short and predictable (and flow stable). Explicit Kanban tokens emphasize the importance of feedback signals. The Kanban token is a signal that new work can be started, or that no new work can be started. It signals that team members need to work together to finish work before starting new work.

System Kanban also implements flow through a cadence of planning what the team can work on and reviewing what is ready to be delivered. These cadences and other feedback loops such as regular (operations) review meetings help to sustain and evolve flow. In the case study the team implemented input queue replenishment and release planning meetings, and it collected qualitative feedback on the process through retrospective meetings (as a form of reflective observation with O-O-D-A: Observe-Orient-Decide-Act). As the WIP limits where put in place, and

the flow started to become more stable, quantitative data (about the flow) started to make sense and the team could start active experimentation with PDCA (Plan-Do-Check-Adjust).

The result of the System Kanban is that system lead-times (time from "Ready to start" to "Ready for UAT") are getting shorter and more stable and delivery rate slowly increases. The team can now start to make promises on the basis of (quasi-) stable system lead-times to the business users. Collaboration within the team is increasing and the team is starting to discuss how they can reduce knowledge bottlenecks in the team. A stable flow now emerges that can be further optimized by learning from small experiments (e.g. pair working to further reduce bottlenecks).

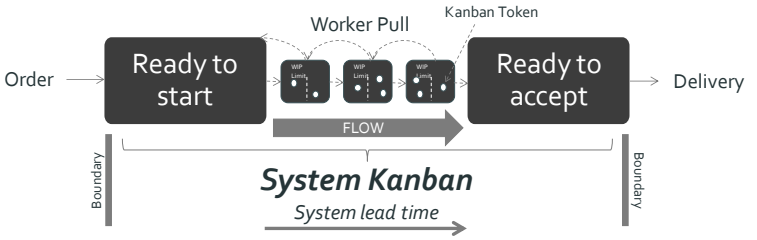

Figure 3: System Kanban

Agility, Not Agile

For the team in the case study, the introduction of System Kanban resulted in a number of characteristics that are often associated with Agile development teams even when, in the presented case, Agile development was not an explicit objective (i.e. it was not the objective to implement Scrum or iterative development). Let's have a look at two of those characteristics.

Self-organized, cross-functional team: System Kanban introduces boundaries in which the team can self-organize. It also introduces a constraint (the WIP limit) that enables novel – collaborative – behavior to emerge. In order to respect the WIP limit, team members need to collaborate. Often this collaboration is cross-functional. In the case study, collaboration between people with different competencies was actually encouraged in order to reduce bottlenecks in the flow of work. Due to this increased collaboration, the team became more cross-functional.

Inspect and adapt: System Kanban is driven by short feedback loops. The shortest feedback loop is provided by the Kanban token (or WIP limit) itself; it provides feedback on whether new work can be started or not. The Kanban cadences (such as the operations review) constitute another set of feedback loops. More importantly, the drive to reduce lead times is a drive to reduce the feedback loops between those that are performing the work and those that need to accept the work. Short lead-time is associated with increased agility irrespective of the fact that the team is doing agile development or not. Notice, by the way, that with the reduced lead-times, also comes more stable lead times. This makes the team more agile and predictable at the same time.

Limits to Success

Agility at the team level only goes so far when there is limited engagement of, and collaboration with, the business team or customer.

Due to the System Kanban initiative, the IT team is able to communicate positive results to their users as their efforts to reduce work in progress (WIP) has lead to shorter, more stable system lead-times and higher delivery rate. The business users (the "Customers" of the team) recognize the improvements that have been made. Still, they feel that more improvement is needed. For the business users the end-to-end customer lead-time (the time from issuing a request to the fulfillment of that request) feels like a much more important metric than the system lead-time. After all, for them, the clock starts ticking when they make the request, not when IT is ready to start working on their request. At least they want to get some insight on when they can expect that IT will start working on their request. They are of the opinion that IT could be much more than an order taker and be more pro-active in working with them on anticipating upcoming requests.

The manager of the team also has his thoughts about the situation. He is convinced that more collaboration with the business is required. He feels that the goal should not just be to deliver better against the requests that have been submitted but that they should collaborate with the business users to increase the business value of the requests. They need to better understand the problems that the business users are trying to solve and be more innovative in providing solutions.

The team itself feels that while they may still (substantially) improve their capability to deliver, the low hanging fruit in the development flow has been picked and most of the turbulence and wait times now come from the incoming work (from the users) and

outgoing work (back to the users). They feel that the business users should collaborate more with them to finish work.

It is clear that agility at the team level only goes so far. In the case, there is a need to start collaborating not just within the team but also between the team and its customers. They need to start collaborating not just about work that is being performed but also on incoming and outgoing work. In Kanban terms, they need to start looking at the end-to-end flow. In the next sections we will explain why and how.

End-to-End Flow

According to the English dictionary, flow is the action, or fact, of moving along in a steady, continuous stream.

In business terms, flow is when value is created for the customer through meaningful work. Explicit in this definition of flow is that both the customer, that has a need or demand, as well as the workers, that fulfill the need or demand, are important. Implicit in this definition is the focus on **short lead-time.**

In general, this means that the time between starting something and completing it should be short. In the end-to-end flow, in a business context, it means that there should be a short time between suspecting a need and satisfying that need (or, at least, learn from the attempt to do so).

System Kanban focuses on the flow of work within an established service. As illustrated in **Figure 4** this is only a relatively small (albeit important) part of the end-to-end flow. Given that the boundaries of a pull system are (by definition) **push-pull boundaries,** and System Kanban is no exception to this, the implication is that while workers may be pulling work; the customer may well still be in the mindset of pushing orders (i.e. pushing requests, ideas, requirements, …).

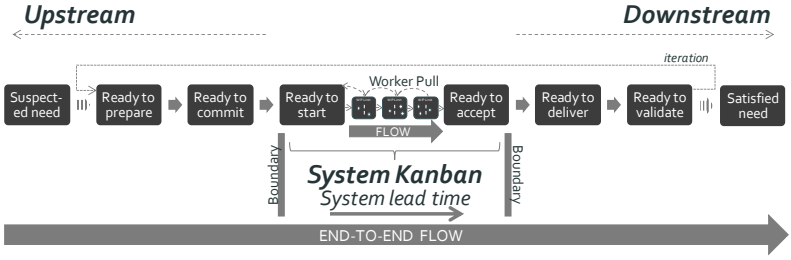

Figure 4: System Kanban in the context of the end-to-end flow

Order pushing by the customer typically manifests itself Upstream as a (large) inventory of orders (backlog) in front of the System Kanban. Downstream it manifests itself as work that has been delivered but not accepted by the customer. A situation that is all too common in many organizations, at the frustration of both the customer as well as the team. Additionally, too **large fluctuations in the demand** may lead to temporary starvation of the System Kanban (a common phenomenon in e.g. project organizations) or a System Kanban that is not presented with the right variation of orders (e.g. variation of orders that does not match the diversity of competencies in a team). The team is starved from work or is working on low quality or low value work while high value work is stuck in the upstream process. This often results in higher costs (e.g. a supplier team that is starved from work) or a loss of potential gains (loss of opportunities because of delay).

All in all, a steady flow of work can only be sustained to the extent that there is a steady **flow of demand** where **customers pull** value rather than just pushing orders.

In the remainder of the article we will discuss **Customer and Upstream Kanban** as a way to create a steady end-to-end flow that covers both the steady flow of demand, and the steady flow of work that leads into a steady delivery.

A Deeper/Wider Look

The time has come to look at the end-to-end flow from request to delivery.

Shortly after implementing System Kanban, the IT maintenance team also started implementing an Upstream Kanban (see **Figure 5**). The purpose of the Upstream Kanban was to manage the stream of incoming requests before being able to commit the work for execution downstream. The Upstream Kanban was modeled according to the change-request (CR) and project life-cycle that was agreed with the change advisory board (CAB). In this life-cycle, it is required that for (smaller) change requests, a score card is filled in to assess the priority of the CR. Possibly a clarification from the business user is required afterwards when the CR is not clear. Projects require a, so-called, blueprint to agree on the business requirements.

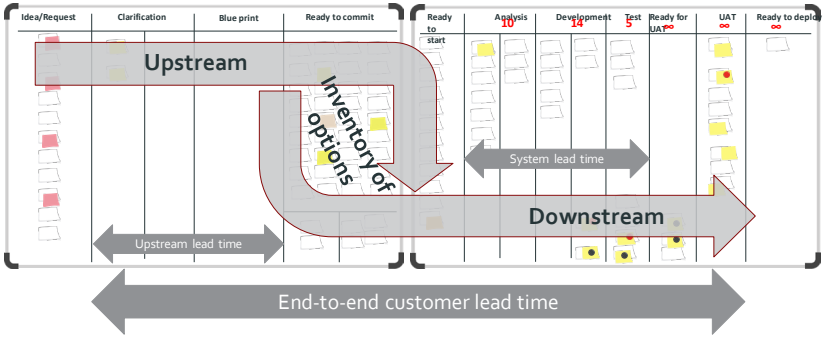

Figure 5: Upstream + Downstream Kanban

The board that the team uses visualizes the end-to-end flow starting from the capture of requests and ending in work items that are ready to be deployed. The flow consists of upstream and downstream with an inventory of options[1] – i.e. work items that are waiting to be committed – in between. The commitment point is right after this inventory. On the board, system lead-time – the stable part of the lead-time – only represents a small part of the end-to-end customer lead-time that the business users are interested in. The only reliable promise that can be made, however, is the promise that is based on system lead-time. This is due to two reasons:

1. Looking downstream, the fact that the "Ready for UAT" and "UAT" columns on the board do not have a WIP limit means that there is no control over the amount of time that is spent in these workflow steps. Some business users respond rapidly when an item is ready for them to accept, and some business users never seem ready to accept work items.

2. Looking upstream, the team cannot make any reliable lead-time promises earlier than the commitment point because of the large inventory of options right before the commitment point.

    a. Larger requests encounter quite some friction in the upstream. Often these are small, but highly visible projects that have high but uncertain value. Because of the friction (i.e. differences of opinion, availability of stakeholders, etc.) they tend to get stuck until the point that they need to be expedited. As the downstream team cannot refute

1 In plain English, an option is a thing that is, or maybe chosen. At his point we use the term option in this sense. In later sections we will use a more technical definition.

these expedited projects this often leads to too much work downstream.

b. Many other items, such as low priority changes, do not encounter this friction. They quickly move through the upstream. These items however get stuck in the inventory as they keep being bypassed by higher priority (and expedited) requests.

The result is a highly variable end-to-end customer lead-time as shown in Figure 6 (x-axis = customer lead-time in weeks; y-axis = frequency of occurrence). Worth noting is the inverse correlation between the upstream lead-time and the customer lead-time. Requests with a short upstream lead-time (e.g. small changes) tend to have a longer than average customer lead-time; and vice versa. I leave it to the reader to think about what this implies.

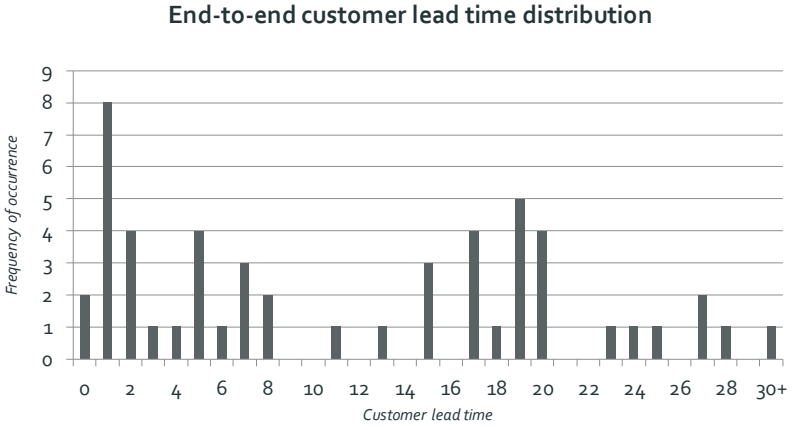

Figure 6: Distribution of Customer Lead-times

Creating Value

Part of the upstream process was designed to make an optimal choice among the incoming requests. The underlying mindset is one in where a separation is made between decision-making and the actual execution: the business decides on priorities and the IT maintenance team executes according to priorities. This reinforces the role of the IT maintenance team as an order taker and the business as the order giver. Furthermore, decision-making relies on a quantitative assessment of requests. All requests are ranked on the basis of a business scorecard.

A lot of effort is put in analyzing and assessing requests. This effort, however, does not seem to result in optimal value. High-value/higher risk requests encounter a lot of friction in the upstream process (delays, rework, confusion, …). They require the most effort to assess but are also the most likely ones to get postponed, modified or even rejected. This leads to quite some wasted effort. On the other end of the spectrum, low value/low risk requests that encounter little friction in the upstream tend to rush through the upstream, but then do not get executed. Again this leads to quite some wasted effort and frustration. For the downstream team it appears that there is a short supply of high value/higher risk items (unless they are expedited) and a high supply of low value/low risk items. The high decision making effort in the end seems to contradict the purpose of the decision-making (optimal choice). When the decision-making effort is high, the number of alternatives that can be assessed is limited as there is limited capacity to do the assessment. When only a limited number of requests can be assessed, there is quite some pressure to pick the right alternative to assess before the assessment even takes place.

In the end not all value is alike; and should not be treated alike. When uncertainty is low (as in low value/low risk requests) there is no risk in expending the full analysis and assessment effort in one go. When uncertainty is high (as in high value/high risk requests) it is advisable to stage the assessment effort. In more technical terms this means that for low uncertainty requests, we can commit the full analysis and assessment effort once we decide to assess the request. For high uncertainty items, we first create an option (i.e. an initial limited effort assessment) before committing the full analysis and assessment effort. In the next section we will see how this plays out in practice.

Not all value is alike.

From the above analysis it is clear that there is room for improvement. While the team refers to their board as the "Kanban board", except for the area from "Ready to start" to "Test" (System Kanban), the rest of the board is visualization without any real underlying Kanban system (also sometimes referred to as "proto-kanban"). The visualization does, however, surface the problems in the end-to-end flow. On the upstream side of the System Kanban,

demand for work (aka "options") is pushed into the inventory in front of the commitment point from which it is pulled into the input queue of the team. At the other end, on the downstream side, of the System Kanban, work is pushed back to the customer for acceptance. In the next two sections we will explore how the scope of the Kanban system can be enlarged to cover a broader section of the end-to-end flow by revisiting the Upstream Kanban and introducing a Customer Kanban.

Pushing the Boundaries

The scope of the Kanban system can be enlarged to cover a broader section of the end-to-end flow.

In order to get a more reliable Customer lead-time and better collaboration with the customer, the team in our case study needed to push the upstream push-pull boundary further upstream and the downstream boundary further downstream. First of all, the team needed to revisit their Upstream Kanban. The purpose of the Upstream Kanban is to evaluate the different options and prepare work items so that they are ready to be committed. The objective is that the team can execute work items without undue delays.

The System Kanban board and the cumulative flow diagram (CFD, see **Figure 7**) indicated to the team that they were not entirely succeeding in properly preparing the work items in the upstream process. The team noticed the large WIP in the "Analysis

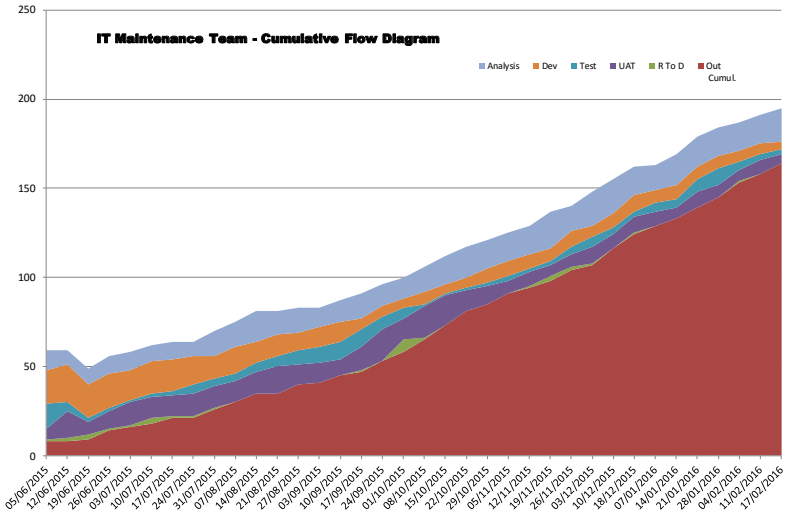

Figure 7: Cumulative Flow Diagram (CFD) for the downstream flow.

ongoing" column after the "Ready to start" column. In the CFD it can be noticed that the large WIP in "Analysis ongoing" is not accidental. It persists as a large area in the CFD (the light-blue area at the top of the CFD). Further investigation revealed that items tended to get blocked because the work item was not properly prepared. Typically, it was not entirely clear what needed to be done, further clarification of the business user was required, and/or the item needed further analysis. Other problems such as the fact that sometimes-conflicting viewpoints between users only became apparent once the work had already started, provided a strong indication that the upstream process needed to be revisited.

The team decided that CRs and (small) projects needed to go through a similar but different upstream process. They found it important to make a clear distinction in the upstream process between synthesis and analysis (see **Figure 8**). The first step – synthesis – seeks to find consensus between the different users and their fragmented and sometimes conflicting demand by building a shared coherent concept. Only after that, the concept could be further detailed, analyzed and, if needed, broken down into work items that can be allocated to the team for execution. The team felt that a proper synthesis was essential to avoid misunderstandings for high value/high risk items. For low value/low risk items the synthesis step could be skipped. See **Figure 10** for how the upstream board was redesigned.

Upstream Kanban

The team also recognized that the upstream process is essentially a triage process: each request needs to go through several steps of selection (see also **Figure 9**) but not all request need to go through the same steps and not all of them with the same "urgency". In the past, selection was mainly based on "readiness": once a request, like a small, low priority change, was ready for the next step it actually got selected for that next step. Because these small requests didn't require a lot of synthesis nor analysis, they tended to flood the inventory before the commitment point (they almost immediately moved from "Idea/Request" to "Ready to commit" on the upstream kanban board). Higher value projects that did require more elaborate upstream work (synthesis and/or analysis), on the other hand, got stuck until suddenly becoming urgent.

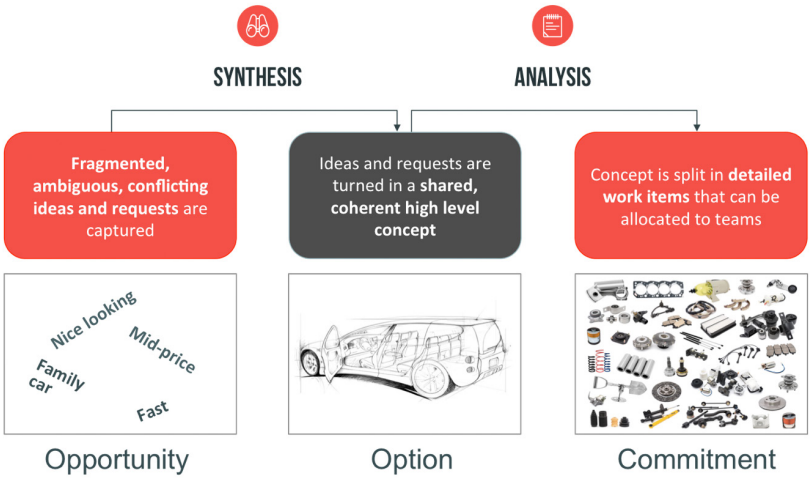

Figure 8: Upstream process for high value / high risk requests.

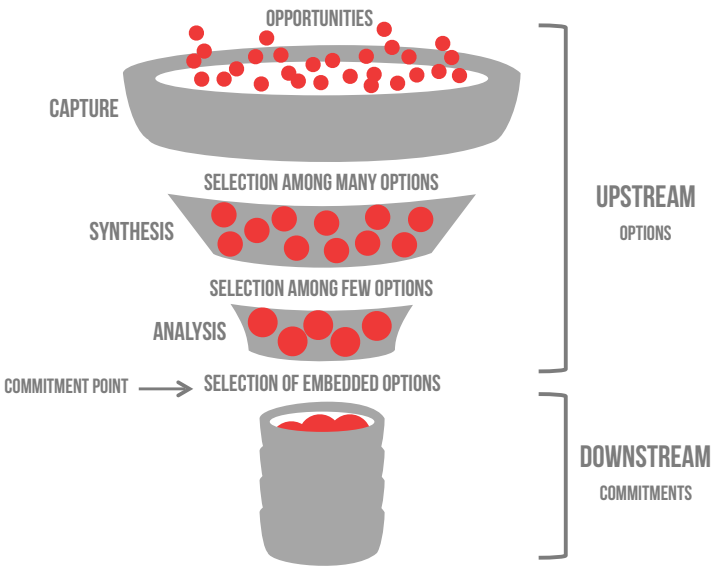

Figure 9: Option selection in the upstream process.

Rather than selection based on "readiness" the team needed to have a way of making selections that would ensure a proper diversity of options at each step in the process. Specifically it needed to ensure that at each step sufficient attention was given on projects and that not too many small requests got selected.

A triage framework was put in place that allowed to triage requests at the moment of capture. In the triage framework a critical distinction is made (amongst others) between "yellow" items that need to follow a longer path through the upstream and that require a more preemptive approach (e.g. preempting stakeholders) versus "green" items that only follow a short path through the upstream and that can be pulled on demand (see **Figure 10**).

Next to the triage framework, the concept of "minimum options" was put in place to avoid downstream starvation (i.e. avoid a lack of options to choose from for the downstream team). Rather than

WIP limits that impose a maximum limit on the work in progress, the minimum option limits impose a minimum on the number of options that are on the board. The minimum option limits work like an order point. It requires the attention of the team and the business users when a limit is not reached. Once the limit is reached the attention can be directed elsewhere again (e.g. back to delivery downstream for the team members, and the day-to-day business for the business users). Proper policies can guide the team and the business users on which priority should be given when minimum limits are not reached.

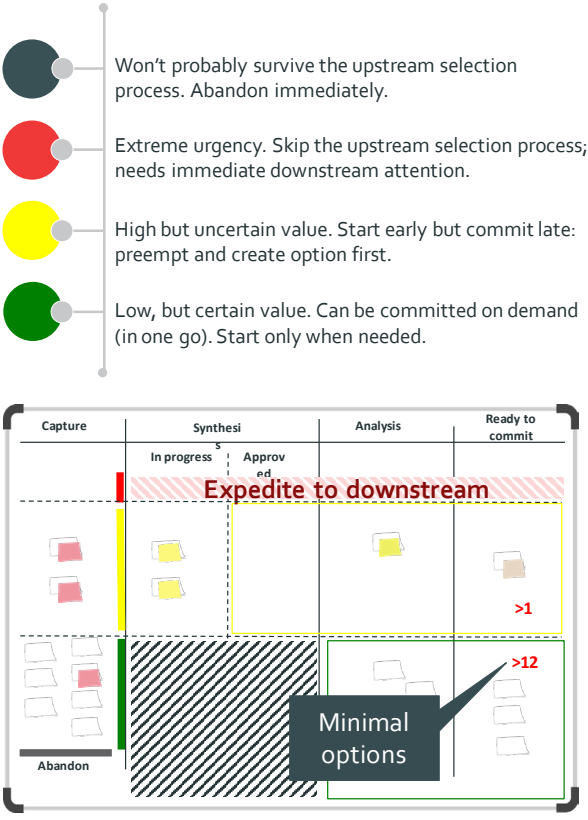

Figure 10: Upstream Kanban with triage and minimum options limits.

Customer Kanban

From customer push to customer pull.

Upstream Kanban alone is not sufficient. While Upstream Kanban allows to make sure that the team is presented with sufficient options of the right variety at all times, it does not prevent the customer (or business user in the case study) from pushing orders onto the team. Often this takes the form of business users that issue requests and use all their power to get the work on those requests started, but are not available to help the team when their input is needed. Often this is very manifest when the work is made "ready for acceptance by the business users", but the business users are not ready to accept.

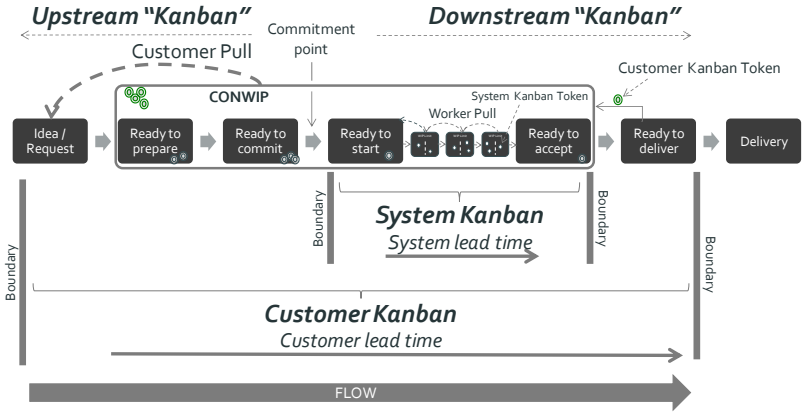

Figure 11: Customer Kanban.

"In the ideal world the rate of demand matches the rate of finishing work; in the real world this is seldom the case. ***Customer Kanban*** *engages the team and the customer to address this problem."*

The team needed a creative solution to deal with this situation. The idea of a Customer Kanban emerged. As shown in **Figure 11** Customer Kanban introduces a one-stage kanban **CONWIP** (CONstant WIP) limit on top of the System Kanban WIP limits and the Upstream Kanban minimum options limits. In practice, Customers receive a number of Customer Kanban tokens[2]. Each customer can at all times issue new requests (in the "Idea/Request" column), but is only allowed to pull a new request for further processing (preparation of the work) when a Customer Kanban token is available. The Customer Kanban token is then attached to the request (when it is pulled). The token is recuperated when the customer accepts the result of the work that was done to fulfill the request (i.e. when the item is "Ready to deliver"). The overall idea is that the customer should not push new orders into the system when older orders are still being processed or waiting for acceptance.

While System Kanban WIP limits are essential to short and stable system lead-times (the lead-time that is important to the team), the Customer Kanban CONWIP limit is essential to short and stable **customer lead-time** (the time between suspected need and satisfied requirement, and the lead-time that is important to the customer). Business users and the team need to collaborate to remain within the constraint of the CONWIP limit. The CONWIP limit is essential to achieve customer pull on top of the worker pull that is enabled by WIP limits.

All in all, Upstream and Customer Kanban push the boundaries of Kanban. They introduce two new constraints/feedback signals. The first is a minimal options limit and the second is a CONWIP limit. While, in practice, they may be introduced separately, both need

2 In the case, the Customer Kanban tokens were actually assigned to groups of users (e.g. HR, Production, Supply-chain, …). The repartition of the tokens was based on an analysis of the historical demand.

to be present to create a steady flow of demand and customer pull. While the CONWIP limit has a more global character (ensuring that the total number of requests in progress are kept in check), the minimal options limit has a more local character (ensuring sufficient options at each step in the upstream flow). Because they are so connected, often we use Customer Kanban as the umbrella term for both Customer and Upstream Kanban at the same time.

To end this section, note that much more can be said about Customer and Upstream Kanban. Quite a few questions are left open in the above explanation: how to set the CONWIP limit; how do you allocate the CONWIP limit over different types of customers; how to determine minimum options limits; do you need maximal options limits on top of the minimum options limits (I suggest you don't, especially if a CONWIP is already in place); which feedback loops and cadences need to be put in place; etc. This is left to further publications and presentations.

Customer Kanban, More Than System Kanban

System Kanban focuses on the flow of work, and collaboration in a team that is delivering a service. It implements pull through WIP limits. But as with any pull system, it has a (push-pull) boundary. In the case study we saw that this push-pull boundary formed the boundary between the order taking team and their order pushing customer(s). In a fast moving world, however, it is not sufficient to just serve a customer; nor is it sufficient for a customer to just push work orders to an order taking team (even if that team is "pulling" work). More than order taking, it requires teams of (knowledge) workers to intimately collaborate with the customers that they are serving; more than order pushing, it requires customers to closely collaborate with the teams they require a service of.

Customer and Upstream Kanban cover a larger part of the end-to-end flow than System Kanban does. Upstream Kanban introduces the notion of minimal options as a way to create a steady flow of demand. Customer Kanban introduces a one stage Kanban CONWIP as a way to create customer pull. Together they foster the collaboration between teams and their customers resulting in shorter and more stable customer lead-times (the lead-time that the customer is interested in) and a more stable overall flow. **Figure 12** summarizes how Customer Kanban (including Upstream Kanban) compares to System Kanban.

While not entirely there yet, Customer and Upstream Kanban are a stepping-stone to the ultimate end-to-end flow. The reader might be aware of an even more encompassing flow that takes learning from the customer feedback into account (as depicted in the end-to-end flow in **Figure 4**). This flow is addressed by

an even larger set of Kanban systems that is called **Discovery Kanban**. Additionally to the flow of demand and customer pull with Upstream and Customer Kanban, Discovery Kanban includes Kanban systems for innovation and change. Particularly it includes Kanban systems that support learning through reflective observation and active experimentation.

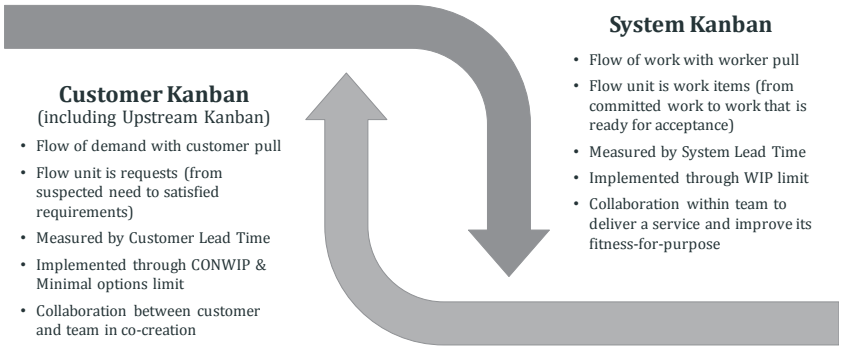

Figure 12: Customer Kanban vs System Kanban.

A Case for Business Agility

A key characteristic of Agility is the ability to rapidly respond to change without loosing momentum. Agile development teams exhibit this characteristic by allowing the user to revisit and change requirements as the user's needs are discovered. Business Agility, according to what we have discussed in this article, must go further than this. It requires the ability to cope with a fluctuating demand created by the diverse, fragmented, and often conflicting requirements when multiple customers or users are involved. Without proper mechanisms in place, even when teams are doing agile development, customers may well keep on pushing requests without the proper collaboration in place to fulfill those requests and teams run the risk of alternating between periods of starvation and periods of overburdening. Most likely they will not be able to deliver what the customer wants when the customer wants it, if we take into account that they need to do this not just for one customer but for all customers.

Customers, or at least the representatives of the customers, experience the mirror image of the problem described above. In order to get the maximum value out of a team, they need to create a steady flow of demand and support the team when they need their help to fulfill their demand. What is truly needed is the capability to anticipate and collaborate to fulfill demand and extract the highest possible value out of that collaboration. This is called business agility.

Conclusion

Today's fast paced business requires Business Agility. Customer Kanban pushes the boundaries of Kanban to deliver this.

The purpose of an organization is to deliver customer value through meaningful work. As the pace, uncertainty and complexity of business is picking up, this proves to be quite a challenge. More and more, organizations need to cope with a fluctuating, fragmented, and often, conflicting demand that outpaces their (agile) delivery. This often creates a tension in the organization between those that represent the demand and those that fulfill the demand.

Agile development only goes so far when there is limited collaboration to tune the pace of demand to the pace of delivery. It is not sufficient anymore to respond to the changing requirements of a single customer. Nor is it sufficient for a customer to push requests to an order taking team. What is needed is a close collaboration to identify the best possible options that can be worked on to deliver most value for the customer and the organization; to ensure that at all times a good diversity of options is available; and to collaborate to deliver against those options. What is needed is a customer that pulls value rather than pushing requests.

Customer Kanban, together with Upstream Kanban, is a way to foster that collaboration. It builds on the foundations of Kanban. System Kanban (the team level Kanban that organizations are familiar with today) develops team Agility (the Agility that is akin to Agile development) through the use of WIP limits; Customer Kanban develops business agility through the use of minimum options limits and CONWIP. Rather than just focusing on the flow of work, it focuses on the end-to-end flow (from suspected need to satisfied need).

While Customer Kanban builds on the foundations of Kanban it is can also help teams that are doing Agile development. In general, organizations that can profit from Customer and Upstream Kanban include: Businesses with a separate IT function can profit from Customer Kanban to align business with IT; Product development companies can use Customer and Upstream Kanban to align product management and product development (and also create alignment among product managers); Project organizations or outsourcing companies can make use of Customer Kanban to align with their customer. While this list is not complete, it does give a flavor of when and where Customer and Upstream Kanban are applicable.

On the next page we end with some pointers on how to get started with Customer Kanban.

Getting Started

The first step towards Business Agility is to experience flow – not just at the team level but true end-to-end flow.

The concept of flow, as discussed in this article, is not an easy concept to master for people that have not experienced it. Rational explanations of flow only go so far. Often they are not sufficient to mobilize a team or organization into action. Flow needs to be experienced. This is a bootstrap problem: in order to mobilize a team, flow must be experienced; in order to be able to experience flow, the team must be mobilized. In the case that was discussed (and many other cases), we made use of extensive flow simulation – in what we call a **flow lab** – to solve this bootstrap problem. In the flow lab, teams are immersed into flow thinking, reflective observation and active experimentation through simulations that cover team flow (at the level of System Kanban) as well as end-to-end flow (at the level of Upstream and Customer Kanban). Other simulations may include cross-team flows (dependencies between teams) and weaving experiments through the flow (Discovery Kanban).

*More information on the **Okaloa Flowlab** and/or **Discovery Kanban** in addition to **Upstream Kanban** and **Customer Kanban**, can be obtained through the author (patrick.steyaert@okaloa.com) or through the Okaloa website (www.okaloa.com).*

Acknowledgements

First and foremost, I would like to thank my partner Arlette for supporting me and helping with writing this article.

Furthermore I would like to thank the people that have reviewed this article and helped me shape my thoughts: Arno Korpershoek, Johan Vanwelkenhuysen, Thomas De Vries, Vincent Vanderheeren and Wim Bollen.

The chapter, An Illustration, was inspired by a blog post of Pawel Brodzinski.

Patrick

27 Sept 2016

About the Author

Patrick Steyaert
patrick.steyaert@okaloa.com

Patrick Steyaert is an integrative thinker and change agent, working with organizations to establish flow in knowledge work to improve agility and predictability in the context of innovation and change. He is a Lean Kanban trainer and coach and regular speaker at international conferences. Patrick is also the winner of the 2015 Brickell-Key award for his work on Discovery Kanban, a recognition of an exceptional contribution to the Kanban community.

Together with Arlette Vercammen, Patrick Steyaert has founded Okaloa to help customers with bringing more "flow" in their organization using an integrative approach with Kanban at its foundation. Okaloa is currently working with customers ranging from high-tech SME's to large incumbent companies applying Kanban in their transformation and strategy implementation.

LeanKanban INCORPORATED

leankanban.com

About Lean Kanban, Inc.

Lean Kanban, Inc. (LKI) is dedicated to developing and promoting the principles and practices of the Kanban Method to achieve the highest quality delivery of professional services through using Kanban. LKI programs include professional development training, a certified training curriculum, events, and published materials.

Certified Kanban Training

Lean Kanban University (LKU) offers a complete curriculum of certified Kanban classes ranging from introductory to advanced, as well as enterprise services. Visit **edu.leankanban.com** to find certified Kanban training or a knowledgeable and experienced coach or trainer in your area. Additionally, Lean Kanban Services offers private training, coaching, and consulting worldwide.

Credentialing Programs

LKU provides leadership training and credentials for managers, coaches, and trainers. Professional designations include Team Kanban Practitioner, Kanban Management Professional, Kanban Coaching Professional, and Accredited Kanban Trainer. Visit **www.leankanban.com** for details.

Global Conference Series

Join the global Kanban community with Lean Kanban events. The Lean Kanban events series focuses on providing pragmatic, actionable guidance for improving business agility and managing risk with Kanban and related methods. Visit **conf.leankanban.com** for a calendar of upcoming conferences and events.