WHITE PAPER

ASSESSING KANBAN FITMENT IN THE FLUID AND FAST-PACED WORLD OF SOFTWARE DEVELOPMENT

- Vikram Abrol, Ketan Shah.

Abstract

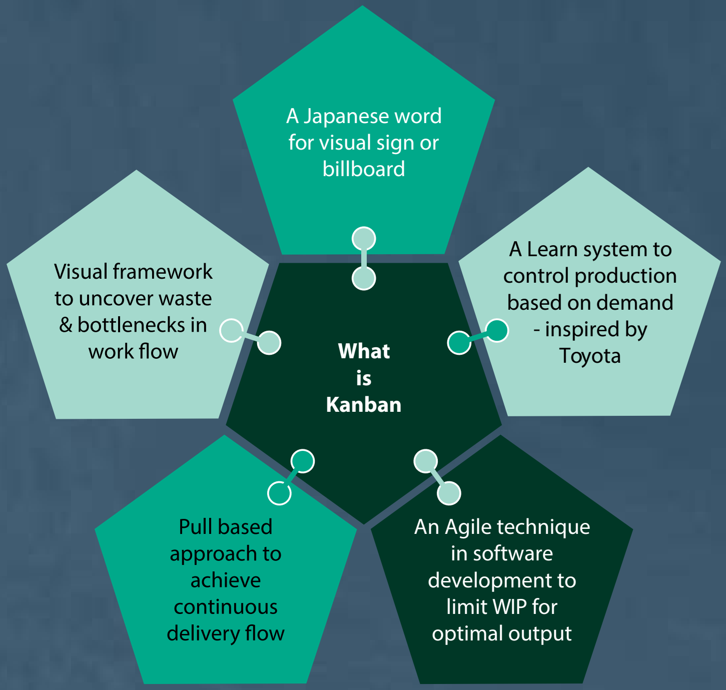
Operating in a business environment governed by speed and agility, IT companies are under constant and immense pressure to reduce time-to-market and enhance product quality. The birth of the Agile approach and models like Scrum owe their existence to this need driving managers to find better solutions. Looking to achieve a faster and more efficient software development cycle (SDC), IT companies have adopted certain methodologies, such as the Lean approach, from the manufacturing industry – another business where speed and efficiency hold the key to profitability. The concept of Kanban also originated in the manufacturing space and has filtered into the IT industry several years ago as an effective approach to manage SDC.

Infosys®
Navigate your next

The terminology related to Kanban in manufacturing context comes mostly from Toyota Motor Corporation in Japan where the system was invented. The Japanese term Kanban literally means a visual card or a signboard. Hence the Kanban system of work management essentially focuses on visualizing the workflow in order to reduce constraints and minimize the work-in-progress (WIP).

Kanban in the context of software development

The term Kanban can take on different nuances in the contexts of manufacturing process and IT software development. Toyota production line staff used a Kanban – an actual card – as an inventory control cue in their manufacturing process and implemented Just in Time (JIT) production methodology to reduce idle inventory and WIP stretches. Kanban in software development, especially in the context of Lean Agile adoption, is about creating a visual process framework which provides information about the status of work: the progress of the software development and stages where the work is obstructed or in a 'waiting' state.



What is Kanban

A Japanese word for visual sign or billboard

A Learn system to control production based on demand - inspired by Toyota

An Agile technique in software development to limit WIP for optimal output

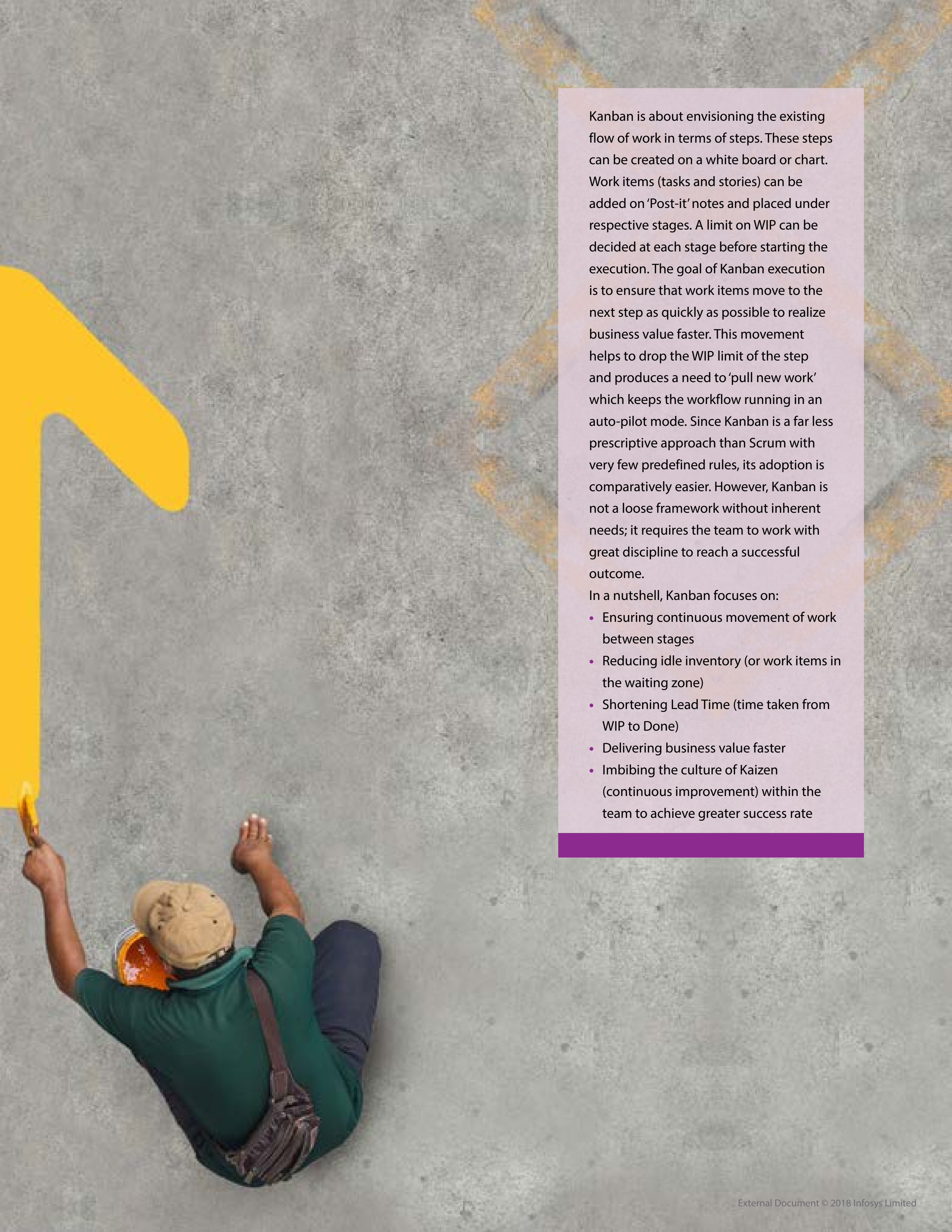
Pull based approach to achieve continuous delivery flow

Visual framework to uncover waste & bottlenecks in work flow

The key principles and essence of Kanban

Let us understand the core principles of Kanban and their interpretation in simple terms. In the table below we have summarized six key principles.

Principles	How to implement
Visualize Work-stages (Flow)	• Divide the work into small tasks, write them on cards or 'Post-it' notes and stick them in columns as work stages based on the system workflow. The workflow can be depicted either physically with cards etc. or with digital boards using popular Agile lifecycle management tools like Leankit Kanban , Jira , SwiftKanban. For a typical development project the work stages can be Design, Development, Test, Production, Deployment, and Done. • Split each stage further into 'Ready' and 'In Progress' to identify workflow bottlenecks, wastages or 'Muda' (in Japanese) .
Limit Work-in-progress (WIP)	• Set an upper limit for each stage that denotes the number of work items (tasks or cards) that can be accommodated in that stage at any point in time.
Measure and Manage Flow	• Measure, manage and optimize the Lead Time (time taken for one item to move from the first stage to the last). The shorter the Lead Time the greater the workflow efficiency and predictability.
Set Up Explicit Process and Policies	• Establish process and policies clearly to handle exceptional situations around work items such as blockers, changing WIP, conditions to break WIP ceiling. This enables the team to implement any rational improvement or change in the workflow.
Implement Feedback Loop	• Provide the team a daily opportunity to learn from the flow, constraints at each stage, throttling, etc. • Add the feedback in the cycle and support the culture of continuous improvement. A fast feedback loop is an integral part of any Agile adoption.
Improve Collaboratively and Evolve Experimentally	• Encourage the team to collaborate increasingly and reduce multi-tasking. This ensures smooth flow and reduces the Lead Time. • Promote the spirit of experimentation with stiff WIP and support efforts to evolve the execution framework. This helps the team understand the bottleneck areas. Collaboration, inspection and adoption are the fundamentals of Agile.



Kanban is about envisioning the existing flow of work in terms of steps. These steps can be created on a white board or chart. Work items (tasks and stories) can be added on 'Post-it' notes and placed under respective stages. A limit on WIP can be decided at each stage before starting the execution. The goal of Kanban execution is to ensure that work items move to the next step as quickly as possible to realize business value faster. This movement helps to drop the WIP limit of the step and produces a need to 'pull new work' which keeps the workflow running in an auto-pilot mode. Since Kanban is a far less prescriptive approach than Scrum with very few predefined rules, its adoption is comparatively easier. However, Kanban is not a loose framework without inherent needs; it requires the team to work with great discipline to reach a successful outcome.

In a nutshell, Kanban focuses on:

- Ensuring continuous movement of work between stages
- Reducing idle inventory (or work items in the waiting zone)
- Shortening Lead Time (time taken from WIP to Done)
- Delivering business value faster
- Imbibing the culture of Kaizen (continuous improvement) within the team to achieve greater success rate

Application and use-cases

Kanban is best suited in a business scenario with a regular inflow of work where priorities are certain, scope is well defined and requirements cannot be moved to a backlog to be addressed in time-boxed (2-4 weeks) iterations. This is one of the reasons why more Kanban implementations can be found in ticket or incident-based projects, production or operation support teams and in business scenarios where stringent service level agreements (SLAs) need to be adhered to by teams. In such cases, any workflow bottlenecks must be quickly detected and made visible to stakeholders in real-time to improve turnaround time. This is where the powerful framework of Kanban delivers results.

However, what are the options , if program has multiple teams some having knowledge of specific area of the whole system like ticket handling / bug-fixing for certain levels while others having support experts who have expertise on a set of modules of the overall product or application. In such a scenario there can be multiple Kanban teams work in parallel. To ensure there is proper sync-up and work distribution team can go for a ceremony what we term as 'Kanban of Kanban' on the similar lines of Scrum of Scrum'. In this a representative from each Kanban team can have a syncup meeting to flag up any blocker that other team may face and decide how the work in backlog can be optimally distributed.

In fact, Kanban can be adopted in any function of an organization: Marketing, Sales, Human Resources (HR) or software development. Such adoption can help the company identify workflow inefficiencies and bottlenecks on a daily basis, learn from such situations and improve operations.

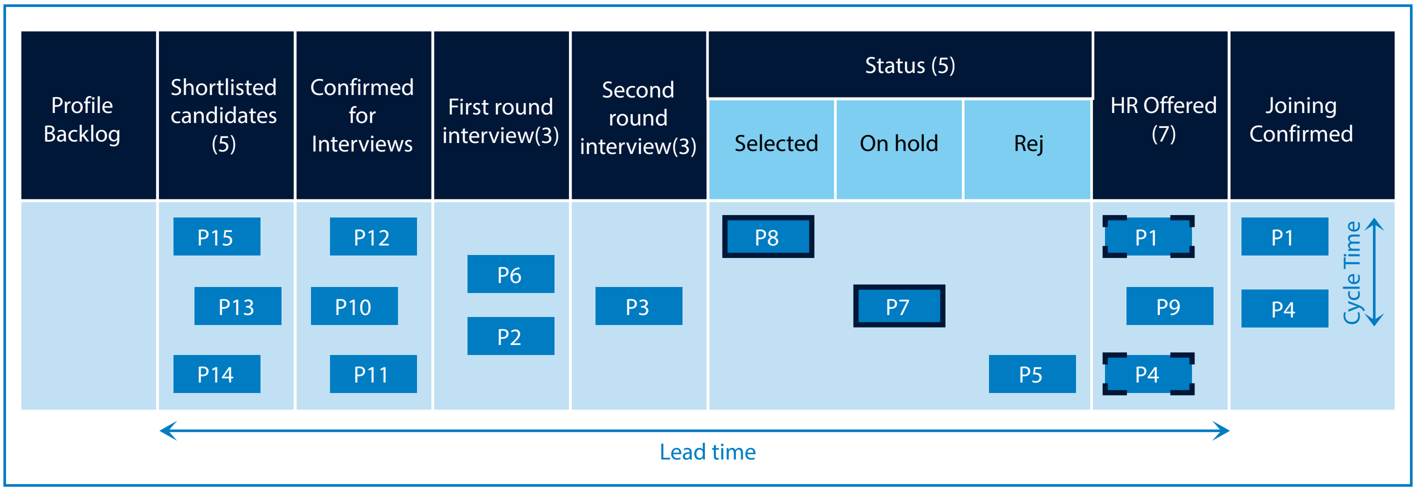
To understand the application of Kanban better, let us take a look at a use case in the HR department where the HR team is recruiting talent for a critical project. Even in a small firm looking to hire a few professionals, the process of managing the entire workflow back and forth along with reporting the status and progress of the process to all stakeholders can be quite chaotic. Stakeholders, obviously, want to know the progress of the hiring process at every stage.

Using the Kanban system, the HR team can visualize the flow and put down all the steps involved in the hiring process on a board:

- Obtaining candidate profiles and backlog
- Short-listing candidates based on roles and needs
- Communicating with candidates
- Confirming interview schedules
- Scheduling and confirming first rounds of interview (telephonic)
- Scheduling and confirming second rounds of interview (in person)
- Updating the outcome of the screening process (selection/rejection/on-hold status of candidates)
- Sending offer letters to selected candidates
- Confirming the joining date

Now the HR team allocates the WIP upper limit to each step and the visual workflow starts disseminating information on the progress of the workflow: between which stages the workflow is smooth and where it is constrained. Using this information, when the team applies process improvement measures, they are able to see the difference between executing the hiring process in a disorganized ad-hoc manner as opposed to the disciplined execution using Kanban principles. Kanban makes the progress visible to everyone in real time without increasing communication overheads and eliminating back-and-forth emails or long complicated spreadsheets. One can imagine the benefits of efficiency and visibility Kanban can bring to a large organization implementing this framework and applying it to each function in the organization. Apart from identifying dysfunctions it also helps in improving the effectiveness at each step, thus directly improving the throughput.

This is how the Kanban board in the HR use case may look like:



The Kanban board for the HR recruitment function radiates key information about the progress and provides visibility to all stakeholders even without necessarily talking with the HR team at length or going through excessive email communication. The key points reflected on the board are:

- A clear lack of candidates for the interviewer even when they are scheduled for interviews. Hence the capacity of interviewers to handle WIP is not fully utilized and needs to be re-planned.
- The 'short-listing' and 'candidate confirmation for interview' stages show signs of dysfunction throttling the outflow towards interviewers.
- Lead Time can be reduced by filling the 'interview' stages to full WIP planned and thus removing obstacles for the stage helping the work item move faster to next stage.

If Kanban is the silver bullet, where does Scrum stand

Scrum, the most popular framework for agile in past many years, is about building and delivering highest priority business requirements incrementally in short sprints – in a time span of 2-4 weeks.

Scrum is more suitable for projects with a large number of requirements that can be prioritized based on business value and spread across sprints. The ability to prioritize and identify work to be completed in short sprints allows the team to build cadence, deliver the work and fulfill their commitment. Not all Scrum teams go to production after every sprint, although the teams are expected to build Potentially Shippable Increments (PSI).

Let us compare Scrum with Kanban on a few key parameters. The comparison can help us determine the aspects a team should consider while choosing a framework for Agile development.

Parameters	Scrum	Kanban	Considerations
Delivery Flow	Every sprint (iterative)	Continuous flow	In case of projects where the flow of incoming work is continuous Kanban is beneficial.
Work Batches	Scrum works on 'push' mechanism in short sprints (1-4 weeks batches).	Kanban does not work in batches. New work is picked up as soon as WIP limit falls.	Kanban can respond to change in needs faster than Scrum, but if time-boxed prioritization is important, Scrum is more suitable.
WIP	WIP limit is applied by way of sprint time-boxing.	WIP limit is applied to each workflow stage and changes on a daily basis.	In Kanban work items in progress at each stage are capped unlike in Scrum. Hence WIP in Kanban shows the dysfunction at the earliest.
Monitoring and Tracking	Burn-down , burn-up charts	Consolidated Flow Diagram (CFD)	In Scrum, teams get more time to address deviations whereas in Kanban any impediment can throttle the pipeline.
Team Roles	Pre-defined and prescriptive roles	No constraints All the existing project roles are allowed	Kanban is accommodating. However, prescriptive roles like Product Owner, Scrum Master would be required in some form.
Team	Cross-functional team	Team specializing in particular skills is allowed	Flexibility in execution while focusing on constantly moving the work to next step is the key benefit of Kanban.
Ceremonies & Framework	Sprint Planning, Review , Retrospective and Daily Scrum	Lightweight Planning is based on WIP continuously. Daily stand-up is not mandatory but is often adopted by Kanban teams.	Kanban provides a framework to reduce overheads but some planning certainly needs to be put in place at milestones.
Metrics	Velocity(Story Point* completed in a sprint) * Size of work	Lead Time Cycle Time, Throughput	Key aspect of Kanban is to measure Lead Time which is a direct measure of the team's efficiency to complete a work item
Estimate	Story Points	Work estimated is based on requirement, usually in effort hours.	Estimation generally varies and hence Kanban emphasizes daily movement rather than planned estimates.
Change	No change within sprints	Change allowed as soon as WIP limit drops	Continuous flexibility to decide the work makes Kanban effective in addressing dynamic prioritization which is a key aspect of ticket-based or SLA-driven projects. Here the shelf-life of a ticket or an incident is expected to be a few hours.
Scaling	Scrum of Scrum	No defined mechanism	Scrum of Scrum can help in collaboration between multiple Scrum teams for a large program and ensure that any cross-team impediment is addressed swiftly. Though "Kanban of Kanban" does not exist, it is conceivable.

Where Kanban may not be the best fit

The Kanban framework is based on the concept of balanced work flow where the tasks move from the left extreme of the board (Ready Queue) to the right extreme (Done) smoothly without any constraints or bottlenecks causing delays and without backlog piling up. The basis of Kanban is the ability to complete one task after another incrementally to deliver value. Hence projects involving tasks that are more prone to fall into or remain in the 'wait' state in the pipeline are not ideal candidates for Kanban application. Predictability is of essence in the Kanban environment. If the size of a task and the time needed for executing it is unknown, the completion of the task can be unclear.

To summarize, Kanban may not be effective in situations where:

- Tasks are prone to spike or are research oriented – where research may span weeks and hence, with all the dynamics involved, it is difficult to predict the movement of tasks and the amount of effort needed
- Tasks evolve and the scope is not defined; the next stage may take indefinite time or go in the 'wait' state till answers are obtained from stakeholders
- Work items are dependent on each other and one blocked work item can obstruct the entire pipeline
- The functionality or business need is such that the extensive amount of work in totality needs to be delivered together or can only be put into production in one go, that is, when small portion of work unit is not valuable till the Minimum Marketable Feature (MMF) is built fully
- A team is in the initial stage of building a product (a green-field project) and the requirements may evolve, drop or be unclear; the task can be stuck at a stage and may block the flow. There may be pieces held up due to tight coupling, waiting for full integration. For example, common framework, data caching, globalization and security are still evolving and an entire product or application cannot be launched in a fully integrated form until all the dependent pieces are devised and tested.

Challenges of adoption

1. Kanban requires a major mindset change in the way team has to focus on getting work unblocked because a single task that is stuck may not allow WIP to fall and deliver work to next stage thus potentially blocking the entire pipeline.
2. Less or no tolerance to any delay in task movement between stages
3. Continuous ups and down in team size based on the WIP (which may change as per backlog inflow)
4. Difficult to predict throughput or achieve consistent lead time when WIP fluctuates regularly

Conclusion

Kanban provides a mechanism to manage the dynamics of frequent change in work that needs to be accomplished at a fast pace. It provides unparalleled visibility in terms of work progress and bottlenecks impeding the flow. Many production support projects based on Information Technology Infrastructure Library (ITIL) face challenges in these areas and this is where, we think, the future of Kanban lies. Kanban can be scaled up to handle a large number of inflow tickets and to prioritize them with the help of Application Lifecycle Management (ALM) tools like SwiftKanban, Leankit Kanban and Jira Agile. This can help us achieve functionality similar to ITIL tools such as Remedy or LotusNotes.

About the Authors

Vikram Abrol has worked in the IT space for more than 15 years and holds PMP and CSM certifications. He is affiliated with the leadership program of the Indian Institute of Management, Calcutta. A Process Consultant and Quality Manager with over 6.5 years' experience at client sites in the US, Vikram has been an Agile Coach for a large retail e-commerce program.

Ketan Shah is Principal Consultant, Agile. He holds CSM and CSPO certifications from Scrum Alliance as well as Six Sigma Green Belt certifications. He has over 15 years of IT experience including more than 7 years in E2E Agile execution from inception to implementation for several large enterprise engagements in various roles like Agile Coach, Scrum Master / Mentor, Delivery Manager and Agile Consultant. He likes to spread Agile awareness via different media such as interactive trainings, mailers, community forum discussions, and Agile COE initiatives.

Watch this space

In a follow-up paper on the application of Kanban at the enterprise level, we will unveil details on how WIP limits in Kanban can be experimentally evolved. We also plan to explain and interpret the Consolidated Flow Diagram (CFD) and other metrics and tools in the next piece.

For more information, contact askus@infosys.com

Navigate your next