

CONTENTS INCLUDE:

- About Kanban
- The Kanban Method
- Introducing Kanban
- Coordination Practices
- Emergent Behaviors
- Prioritization and more...

Getting Started with Kanban for Software Development

By David J. Anderson and Janice Linden-Reed

ABOUT KANBAN

Software organizations want development to be predictable: to accurately state what work will be done and when it will be finished. To make such predictions, mechanisms must be in place to determine prioritization, workflow and lead time to delivery.

Even if all these mechanisms are in place, there is the risk that unfolding events can throw off predictability. What if someone is not available for a handoff? What if the conditions of the business change, forcing re-prioritization? A business needs to be able to see their situation clearly and make needed corrections quickly.

Kanban takes an organization's current development process and provides greater visibility into the status of the work and how it is proceeding. Kanban provides a method to continually adapt in order to smooth out kinks in the arrival of new development work. In this way, it allows the organization to avoid crises and respond more quickly and easily to issues that do arise.

Kanban also gives more precise direction on how to invest development energy into only the most valuable work. The end result is a development pipeline that is predictably and efficiently delivering high value work.

The Kanban Method reduces risk and increases flexibility, resulting in a more resilient development cycle.

Why Kanban?

Why make any change in your development process? Consider your organization's unique situation and goals before starting. Here are some benefits that can be delivered by the Kanban Method.

Goal 1: Optimize Existing Processes – Introduction of visualization and the limiting of work-in-progress (WIP) will catalyze change with minimal disruption.

Goal 2: Deliver with Higher Quality – Limiting work-in-progress and defining policies for work prioritization will bring greater focus on quality. Policies can also address quality criteria directly.

Goal 3: Improve Lead Time Predictability – There is a correlation between the amount of work-in-progress, lead time and defect rates. Limiting WIP makes lead times dependable and keeps defect rates low.

Goal 4: Improve Employee Satisfaction – Kanban reduces context switching and pulls work at the rate the team can complete it. Working at a more even, predictable pace, means employees are never overloaded.

Goal 5: Provide Slack to Enable Improvement – Creating slack in the value chain improves responsiveness to urgent requests and bandwidth to enable process improvement and quality improvement.

Goal 6: Simplify Prioritization – Kanban enables fast reprioritization to accommodate changes in the market.

Goal 7: Provide a Transparency on the System Design and Operation – Improved visibility builds trust with customers and managers. It also shows the effects of actions or inactions. As a result, collaboration improves.

Goal 8: Enables Emergence of a "High-Maturity" Organization – As improvements are implemented, organizational maturity improves leading to better decision making and improved risk management. Risk, managed appropriately, brings predictable results.

THE KANBAN METHOD

The Kanban Method is defined by a set of key principles and practices. Five core principles enable The Kanban Method.

1. Visualize the Workflow	Represent the work items and the workflow on a card wall or electronic board.
2. Limit Work-in-Progress (WIP)	Set agreed upon limits to how many work items are in progress at a time.



Agile Team Management Made Easier

30,000 Agile Teams
170 Countries

Try It Free

See for yourself why VersionOne is the most utilized agile management tool on the planet

VersionOne.com **VERSIONONE**
Simplifying Software Delivery

3. Measure & Manage Flow	Track work items to see if they are proceeding at a steady, even pace.
4. Make Process Policies Explicit	Agree upon and post policies about how work will be handled.
5. Use Models to Evaluate Improvement Opportunities	Adapt the process using ideas from Systems Thinking, W. E. Deming, etc

When these 5 conditions are present, a Lean software development method will emerge over time.

Attributes of Lean Software development

Lean software development has these principles: Respect people, eliminate waste, defer commitment, create knowledge, deliver fast, build quality in and optimize the whole.

Think of the development process as a pipeline. Anything that slows down the pipeline is waste. Look for ways to increase the value while improving the efficiency of the process. The improvements will happen when you can see what is going on and reduce waste.

Lean software development looks to the system, the way work is managed or handed off, for the source of errors. When an issue is identified, the development group looks to how they can change the system for permanent improvement.

Lean Thinking asks us to focus on the system – the process and its policies – and not the performance of individuals. The focus is on a better economic outcome rather than on utilization of individual workers.

VISUALIZE THE WORKFLOW

The Kanban Method takes an existing software development process and brings it into sharper focus. Rather than re-defining the development process, it reveals details about the current process providing visibility that allows the organization to better understand and better communicate about needed changes.

Kanban brings greater visibility into what work is being done and the issues that arise with work items or the workflow. Understanding the current capability of the system of development gives more accurate predictability.

Mapping the Value Creation Network

The value creation network is the complex sequence of information discovery and creativity that takes place from initial request to delivery of software. Visualizing such a network can be challenging and it typical degenerates into a rough sequence of steps: common handoffs are shown; and time spent queuing is shown. Optimization often occurs from analysis of the value creation network.

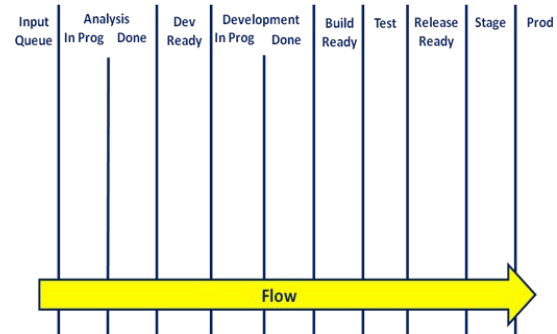
Different work item types may flow differently and require a separate analysis and visualization.

The card wall will be created for the portion of the value creation network that is under political control of managers supportive of the Kanban initiative.

Creating the Card Wall

It is typical to draw card walls to show the activities that happen to the work rather than specific functions or job descriptions.

Draw columns on the board that represent the activities performed, in the order first performed. Model in-progress and completed work by splitting the columns. Add the initial input queue and any downstream delivery steps that you wish to visualize. Finally, add any buffers or queues (Ready or Done states) between activities.



A card wall showing workflow

It is important to create a card wall that reflects the current development process, with the intent to improve it over time.

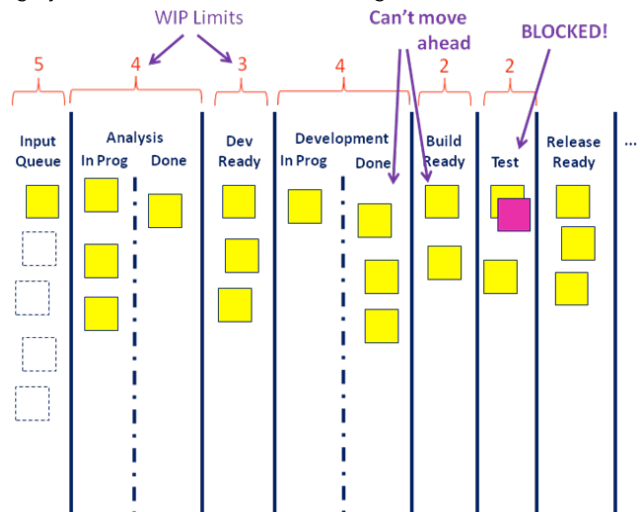
The cards themselves could be sticky notes or index cards. They may represent features, stories or tasks. Critical information for a card would be title, reference number, and date it entered the system.

LIMIT WORK-IN-PROGRESS (WIP)

Leadership is the secret ingredient - all 5 core principles of the Kanban Method will not enable a significant improvement to happen without it. The WIP limit provokes the conversations that lead to improvements.

A limit on work-in-progress constrains how many work items can be in each workflow step at a time. When a work item progresses, a slot opens and a new work item can flow into the development step.

A key result of a WIP limit is that blocked work “holds up the line”. A blocked work item still counts against the limit, so a situation may arise in which no new work can progress until the block is resolved. This drives collaboration as the team is highly motivated to clear the blockage.



Blocked item in Test must clear to open up a space

If a conflict with the limit arises, the team has a choice to break the WIP limit and ignore the issues, or face up to the issues and address them using models from lean thinking.

How big a limit? - WIP limits for workflow steps should be set as an average number of items per person, developer pair or small, collaborative team. Limits should be in the range of one to three items per person, pair or team.

Excessive time should not be wasted trying to determine the perfect WIP limit; simply pick a number that is close enough, and make progress. Empirically adjust if necessary.

Agreed upon limits - WIP limits should be agreed upon through consensus with up- and downstream stakeholders and senior function management as well as the development team. Not every step must have a limit.

Limit the queue - A work-in-progress (WIP) limit on the input queue of upcoming work focuses attention on what to start next.

It provokes a focus on value. A queue between the backlog and development can be used for interim prioritization.

Queue limits should be kept small, typically only large enough to absorb the natural variation in size of items and task duration, and the period between queue replenishment meetings.

Handling bottlenecks - Bottlenecks, once identified, should be buffered. Add a lane on the card wall as a holding area for stories around the bottleneck area. This introduces some slack into the process.

Buffer sizes should be as small as possible but large enough to ensure the bottleneck is never idle.

No WIP limit? - Care should be taken that establishing any unlimited (no WIP limit) workflow steps does not introduce bottlenecks or cause excessive transaction- or coordination costs when deliveries (batch transfers downstream) are made.

Collaboration - Limited WIP means that team members might have to “swarm” on a work item to complete it. This cross-functional collaboration is a strong way to finish work faster and with higher quality.

MEASURE AND MANAGE FLOW

Flow is the visible progression of the work items through the system. Flow should proceed at a consistent pace. If a work item seems to languish, a conversation should happen to resolve how the team can get it “flowing” again. If all work items within a particular step of the workflow seem to stall, this may suggest a bottleneck in the system. A conversation and action toward resolution can happen with the team and management based on such visible evidence of an issue.

Flow is not something that can be manipulated directly. It is an indicator that notifies when some aspect of the process needs adjusting or some area needs attention. The goal with a

kanban system is to create the conditions for the work to flow at a dependable, measurable pace. The systemic variables that can be directly influenced are the team members, the typical work item size, the workflow steps, and the WIP limit. Flow of an individual item can be affected by issues such as ambiguous requirements, dependencies, or approvals.

EXPLICIT PROCESS POLICIES

Having agreed upon policies regarding: workflow; WIP limits; approvals; prioritization; and other process topics; results in greater understanding, greater compliance, and less contentious discussions when exceptions do occur. If a suggestion to violate a policy is made, a discussion can happen reviewing the current policy and posing the question of whether the group feels the policy should be changed, or if it is worthwhile to allow a one-time exception. Focus on policies depersonalizes issues and exceptional circumstances. This leads to a healthier, more collaborative workplace. It is often best to post explicit process policies in a public team area such as next to the card wall.

USE MODELS TO EVALUATE IMPROVEMENT

W. Edwards Deming suggested that we should study the system and its capabilities. The Kanban Method embraces this idea. Gathering data on system performance: flow time; WIP; delivery throughput; will provide scientific insight into opportunities for improvements. It is common (and encouraged) for kanban teams to frequently modify their processes and policies during their daily check-in meeting and at formal retrospectives and operations review meetings.

Useful models to learn and use may include work from:

- **Womack & Jones** on *Waste Reduction*
- **Eliyahu M. Goldratt** on *Theory of Constraints*
- **John Seddon and Peter Senge** on *Systems Thinking*
- **W. Edwards Deming's** *System of Profound Knowledge* and specifically his work on variability
- **Taiichi Ohno (Toyota Production System)** on *muda, muri and mura*.

INTRODUCING KANBAN

The best way to introduce Kanban into an organization is incrementally rather than through a planned transition initiative and prescribed training program. It may also be useful to start with a pilot team who can “try it out” and demonstrate it for the rest of the organization. The most important thing is to gain consensus around the introduction of Kanban and just start using it, changing as little of the existing culture and process as possible. Here are 12 steps to getting started with Kanban.

1. Agree on a set of goals for introducing Kanban. What benefit do you want to see for your organization?
2. Determine a set of unique types of customer valued work

that flow through the organization, examples include bug, change request, or feature.

3. Map the value creation network for different types of work. This will represent the underlying model of your system (see step 5).
4. Begin to visualize the flow through the system and study the system performance and capabilities. The purpose is to understand the system operation not affect performance or implement control.
5. Define some starting point from where you want to control system performance. Identify what is upstream of that point and the upstream stakeholders. Define some exit point beyond which you do not intend to control system performance. Identify what is downstream of that point and the downstream stakeholders.
6. If there are different urgency criteria for the different work item types, define a class of service for each with relative priorities.
7. Study the demand for each work item type. Is the arrival seasonal or event-driven? What is the tolerance for late or unreliable delivery of each? Create a risk profile that allocates a percentage of resources to each work item type.
8. Meet with team members, upstream and downstream stakeholders to discuss and agree on policies around WIP limits, prioritization meetings, and release mechanisms. If you choose to introduce the concept of class of service, discuss and agree target delivery policy for each class of service.
9. Create a board/card wall to visualize the workflow. This will cover the workflow that exists between the input and exit points defined in step 5.
10. Optionally, create an electronic system to track and report the same.
11. Agree to have a standup meeting every day in front of the card wall/board, as a daily opportunity to drive improvements. Agree to have a regular operations review meeting for longer term retrospective analysis of process performance.
12. Educate the team on the new board, WIP limits, and pull system operation.

COORDINATION PRACTICES

Holding regular meetings reduces the coordination cost for those meetings and improves attendance.

Input queue replenishment and delivery planning should be done independently and have their own independent cadence.

Daily standup meetings should be used to discuss issues, impediments and flow. Such standup meetings do not typically

follow the established pattern of other Agile development methods.

Daily standups are an essential part of a culture of continuous improvement. Because the standup brings the whole team together briefly each day, it provides an opportunity for all stakeholders to suggest and discuss process performance problems. The period immediately after the standup often develops into informal process-improvement discussions.

Grooming the backlog with regular triage to reduce its size improves the effectiveness and efficiency of input queue replenishment meetings.

Issue management, escalation and resolution is a core discipline in improving the performance of the system. It is important that teams develop a core capability at issue management, escalation and resolution.

For all types of work items escalation paths and policies should be clearly defined.

EMERGENT BEHAVIOR

Through its 5 core principles, the Kanban Method enables incremental process improvement through repeated discovery of issues affecting process performance. It supports gradual, continual improvement toward higher performance and greater quality. It drives process evolution, rather than a revolution.

Once in place, Kanban has been observed to stimulate emergent behavior in many organizations. Such emergent behaviors may include:

- Process uniquely tailored to each project/value stream
- Decoupled cadences (or "iterationless" development)
- Work scheduled by (opportunity) cost of delay
- Value optimized with classes of service
- Risk managed with capacity allocation
- Tolerance for process experimentation
- Quantitative Management
- Viral spread (of Kanban) across the organization
- Small teams merged for more liquid labor pools

PRIORITIZATION

In a limited WIP pull system, prioritization practices will guide which work items should be pulled next to optimize value. The following techniques help to enable flow by creating prioritization process policies that the management and the team can agree on and follow.

Input Queue Replenishment Meetings

Kanban decouples input queue replenishment cadence from development lead time and delivery. This is unlike Agile methods that couple prioritization activity to a time-boxed period of development activity such as 2 weeks.

Frequent queue replenishment meetings provide visibility for upstream stakeholders and help to build trust.

Ad-hoc or on-demand input queue replenishment meetings can make sense for high-maturity organizations with strong capability to coordinate such meetings at short notice. Such teams can call such a meeting whenever they pull a new feature to develop. Other organizations should schedule input queue replenishment with a regular cadence such as weekly.

Capacity Allocation by Work Item Type

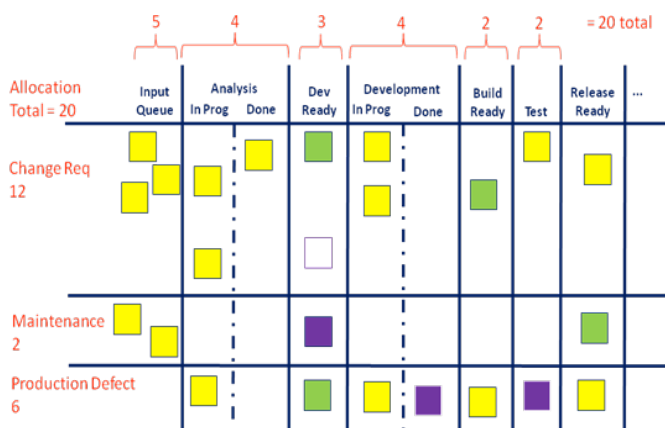
Different work items can have different workflows. For example, a defect may enter the development process at a different point and may be processed in a different way than a new feature. A "defect" and a "new feature" are 2 examples of work item types.

Relative priorities can be set for each work item type. Establishing a policy around how much of the development team's availability will be used for each work item type at any time is called capacity allocation by work item type.

It is useful to set an allocation by work item type to provide clarity about which kinds of work the team is spending time developing. It sets an agreement with the development team and management about how their resources are best spent.

Displaying on the card wall - Horizontal swimlanes are often used for each work item type and a WIP limit set for each swimlane.

Setting policies - Capacity allocation requires comparative demand analysis across the different types of work received by the kanban system. Some work item types will enter the system more frequently or they may be considered more urgent. These factors should be considered when creating a policy about the quantity of resources to allocate to each work item type.



A card wall showing allocation by work item type & limits

Prioritization by Class of Service

Class of service allows the development team and the management to assign handling priority to work according to its urgency.

If classes of service are used properly and combined with a regular delivery cadence, very few complaints are likely to be received, even if a significant portion of items miss their target

lead time. The most time critical work items will be prioritized ahead and should be delivered on time.

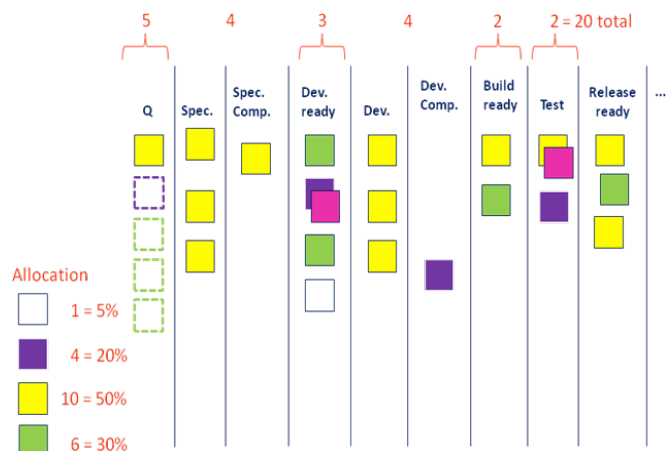
Classes of service enable a team to self-organize around the flow of work and free up management time to focus on the process and its performance (or capability).

Establishing Class of Service - Work items can be assigned to a class of service according to their business impact. Examples of class of service would be "fixed date", "expedite", or "standard". It is typical to assign class of service based on impact and time-criticality. Items that incur high costs or lose value quickly will receive faster treatment via a higher class of service.

Setting policies - A set of management policies should be defined for each class of service. The policies should show which class of service is highest. For example, an "Expedite" class might always be prioritized ahead of a "Standard" class item. These policies will determine which work item to pull on a daily basis. Well defined and understood classes of service enable managers to trust that team members make appropriate risk decisions when selecting work.

Displaying on the card wall - Classes of service should be visually displayed by using, for example, different colored cards to represent the class of service or different horizontal swim lanes on the card wall.

Capacity Allocation - It is also possible and useful to allocate capacity within the kanban system to classes of service. It is helpful to set an allocation to make to ensure that items in the lower priority classes are completed. It is also generally helpful for the team and the management to specify allocation as an agreement of how their resources should be spent.



A card wall showing allocation by Class of Service

Predictability via Service Level Agreements (SLA)

A Service Level Agreement (SLA) should include the target lead time from when a work item enters the input queue to when it will be delivered. The SLA represents the promise to the customer for a work item of a given type and class of service

Work items should be tracked and measured as they move from step to step through the system. Performance against the

SLAs should be monitored and reported at operations review meetings. Process improvements should be made to improve predictability against the customer promise defined in the SLAs. SLAs should be matched to system capability, and the system improved to match with customer expectations and market demands.

WHAT NEXT?

The best approach to working with the Kanban Method is to get started. It is not necessary to implement all 5 core principles initially. They can be introduced gradually and adjusted to match the needs of the organization and its culture.

There are books, blogs, mailing lists and conferences dedicated to kanban and lean. Some of the best resources are located at kanban101.com, limitedwipsociety.org & leanssc.org.

Kanban Book and Website

"Kanban" by David J. Anderson is the 250-page definitive guide to Kanban as a change management system for software development. It explains what Kanban is, why it is used, and how to implement it.

"Kanban" is available at major online book retailers and also at channelkanban.com.

ABOUT THE AUTHORS



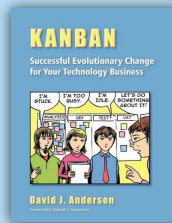
David J. Anderson leads a management consulting firm focused on improving performance of technology companies. He has been in software development for 26 years and has managed teams on agile software development projects at Sprint, Motorola, Microsoft and Corbis. David is credited with the first implementation of a kanban process for software development in 2005. David was a founder of the agile movement through his involvement in the creation of Feature Driven Development. He was also a founder of the APLN, a founding signatory of the Declaration of Interdependence, and a founding member of the Lean Software and Systems Consortium. He moderates several online communities for lean/agile development. He is the author of the books "Kanban - Successful Evolutionary Change for Your Technology Business" and "Agile Management for Software Engineering - Applying the Theory of Constraints for Business Results". David has worked to create a synergy of the CMMI model for organizational maturity with Agile and Lean methods through projects with Microsoft and the SEI. He is based in Seattle, Washington, USA.



Janice Linden-Reed is a Lean-Agile Certified Scrum Master. Janice has 20 years of experience as a project manager, including over 4 years of agile, lean and kanban management work. She has worked with all sizes of software development organizations including large international teams. Janice has a particular interest in lean startups and teams making the transition to agile or lean. To help novice users, Janice created the Kanban101.com website. She has presented to agile groups on kanban fundamentals and on lean-agile transition topics.

Janice is an Associate with David J. Anderson and Associates, Inc. She was an organizer for the Lean and Kanban conference as well as the Lean Software and Systems Conference. She actively manages multiple distributed teams as a CSM.

RECOMMENDED BOOK

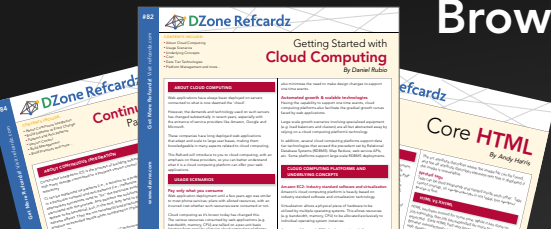


Order the definitive guide to Kanban. The Kanban Method will improve your existing development process. This book explains why and shows you how to get started using it right now. Case studies and illustrations make it easy to adopt the improvements you need for your unique situation.

BUY NOW

books.dzone.com/books/kanban

Browse our collection of 100 Free Cheat Sheets



Free PDF

Upcoming Refcardz

- Apache Ant
- Hadoop
- Spring Security
- Subversion



DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheatsheets, blogs, feature articles, source code and more. "DZone is a developer's dream," says PC Magazine.

DZone, Inc.
140 Preston Executive Dr.
Suite 100
Cary, NC 27513
888.678.0399
919.678.0300

Refcardz Feedback Welcome
refcardz@dzone.com
Sponsorship Opportunities
sales@dzone.com

