Starting with Value Stream Mapping and Kanban

A practical workshop on Value Stream Mapping & WIP

Student Guide

|  |  |
| --- | --- |
| C:\Users\Evan\Dropbox\Speaking and Presentations\Training\cc-by-sa.png | Starting with Value Stream Mapping and Kanban by Evan Leybourn is licensed under a Creative Commons **Attribution-ShareAlike** 3.0 Australia License <<http://creativecommons.org/licenses/by-sa/3.0/au/>>  Evan Leybourn  [evan@theagiledirector.com](mailto:evan@theagiledirector.com)  Twitter: @eleybourn |

# C:\Users\Evan\Dropbox\Writing\directing the agile organisation\Promotion\Book Cover - Directing-the-Agile-Organisation-338x400.pngOther Works by Evan Leybourn

## Directing the Agile Organisation – by Evan Leybourn

[**http://theagiledirector.com/book**](http://theagiledirector.com/book)

* Embrace change and steal a march on your competitors
* Discover the exciting adaptive approach to management
* Become the Agile champion for your organisation

Business systems do not always end up the way that we first plan them. Requirements can change to accommodate a new strategy, a new target or a new competitor. In these circumstances, conventional business management methods often struggle and a different approach is required.

Agile business management is a series of concepts and processes for the day-to-day management of an organisation. As an Agile manager, you need to understand, embody and encourage these concepts. By embracing and shaping change within your organisation you can take advantage of new opportunities and outperform your competition.

Using a combination of first-hand research and in-depth case studies, Directing the Agile Organisation offers a fresh approach to business management, applying Agile processes pioneered In the IT and manufacturing industries.

# Table of Contents

[Other Works by Evan Leybourn 2](#_Toc421652153)

[Directing the Agile Organisation – by Evan Leybourn 2](#_Toc421652154)

[Table of Contents 3](#_Toc421652155)

[What Does Agile Mean? 4](#_Toc421652156)

[The Agile Manifesto 5](#_Toc421652157)

[Agile Methods 6](#_Toc421652158)

[Key Points 7](#_Toc421652159)

[The Origin of Lean 8](#_Toc421652160)

[Understanding Waste 9](#_Toc421652161)

[An Overview of Kanban as a Software Development Method 10](#_Toc421652162)

[Kanban 12](#_Toc421652163)

[Task Lifecycle 13](#_Toc421652164)

[Value Stream Mapping 15](#_Toc421652165)

[Kanban Boards 17](#_Toc421652166)

[Inspection 20](#_Toc421652167)

[Cumulative Flow Diagrams 21](#_Toc421652168)

[Cycle Time Run Charts 25](#_Toc421652169)

[Kaizen 28](#_Toc421652170)

[5S 31](#_Toc421652171)

# What Does Agile Mean?

‘On two occasions I have been asked, “Pray, Mr Babbage, if you put into the machine wrong figures, will the right answers come out?” [...] I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question.’

Charles Babbage, 1864

## The Agile Manifesto

The “Agile Software Development Manifesto” was developed in February 2001, by representatives from many of the fledgling “agile” processes such as Scrum, DSDM, and XP. The manifesto is a set of 4 values and 12 principles that describe “What is meant by Agile".

### The Agile Values

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

### The Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time-scale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity – the art of maximising the amount of work not done – is essential.
11. The best architectures, requirements, and designs emerge from self-organising teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

### The Lean Principles

In addition to the principles from the Agile manifesto, there are 7 principles defined for lean development.

1. Eliminate waste
2. Amplify learning
3. Decide as late as possible
4. Deliver as fast as possible
5. Empower the team
6. Build integrity in
7. See the whole

## Agile Methods

The term Agile actually refers to a concept, not a specific methodology. There are many, and sometimes conflicting, methods that can be used under the Agile umbrella. These include;

* Agile Unified Process,
* Behaviour Driven Development (BDD),
* Crystal Clear,
* Dynamic Systems Development Method (DSDM),
* Extreme Programming (XP)
* Feature Driven Development (FDD),
* **Kanban**
* **Lean Development,**
* Rapid Application Development (RAD),
* IBM - Rational Unified Process (RUP),
* Scrum,
* Test Driven Development (TDD),

## Key Points

All of the above methods have four key points in common.

1. Iterative design process
2. Continuous stakeholder engagement
3. Aims for quality and reliable systems
4. Short development cycles (up to a month) allows to regular delivery of improvements

This shows that an agile approach is appropriate in contexts where the outcomes are not known (or can’t be known) in advance and where the delivery of the outcomes cannot be fully controlled. This is especially relevant in business intelligence environments given the natural ambiguity around reporting requirements, data quality and information management.

The following figures[[1]](#footnote-1) are an excellent example of the differences between traditional (or phased) development vs. the agile approach of iterative development.

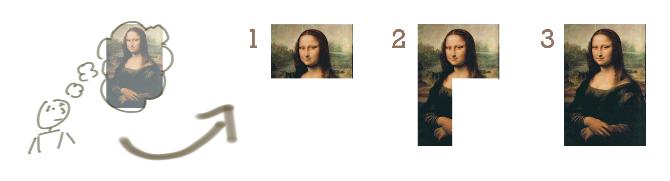


Figure : The Traditional Approach (Phased delivery of known outputs)



Figure : The Agile Approach (Iterative delivery to meet changing EXPECTATIONS)

## The Origin of Lean

Lean Manufacturing, often called lean production, or just ‘Lean’, is a streamlined manufacturing and production technique, as well as a philosophy that aims to reduce production costs, by eliminating all ‘wasteful’ processes. Put another way, Lean focuses on ‘getting the right things to the right place, at the right time, in the right quantity, to achieve perfect workflow’.

Lean Manufacturing provides a set of techniques to identify, and eliminate, waste which, in turn, improves quality, and reduces overall production time and cost. In addition, Lean Manufacturing also improves the ‘flow’ of production through the system. These techniques include:

* **Value stream mapping**: Analysing and planning the flow of materials and information that is required in the production process.
* **5S**: This is an approach to quality and continuous improvement. The five S’s are: Sort (to clean and organise the work area), Set in Order (arrange the work area to ensure easy identification and accessibility), Shine (mess prevention and regular maintenance of the work area), Standardise (create a consistent approach for carrying out production processes), Sustain (maintain the previous four S’s through discipline and commitment).
* **Kanban**: This will be covered later.
* **Fail-proofing**: Prevent human errors before they occur.
* **Production levelling**: Ensure that each step in the production process delivers at a constant rate, so that subsequent steps can also deliver at a constant rate. No step in the production process should produce goods at a faster rate than subsequent steps, or consume goods at a slower rate than preceding steps.

Finally, Lean Manufacturing emphasises Kaizen (改善) or Continuous Improvement; the ongoing, incremental and regular technique of improving all processes, products and functions relating to production, resources, organisational management, and the supply chain.

It should be noted at this point that many of the terms in Lean Manufacturing have been translated from the original Japanese. As such, they often lose the context, or secondary meanings, of the term. Where possible, this context is described throughout this course.

## Understanding Waste

The techniques and frameworks within Agile & Lean aim to increase development efficiency, by eliminating all ‘wasteful’ processes. Drawing on the successful concepts from the Lean manufacturing frameworks, we can define 3 major forms of waste.

* **Mura (Unevenness):** Mura exists where there is a variation in workflow, leading to unbalanced situations, most commonly where workflow steps are inconsistent, unbalanced, or without standard procedures.
* **Muri (Overburden):** Muri exists where management expects unreasonable effort from personnel, material or equipment, most commonly resulting from unrealistic expectations and poor planning.
* **Muda (Waste):** Muda is any step in the production workflow that does not add direct value to the customer. The original seven wastes, as defined by the Toyota Production System (TPS), were:
  1. Transport,
  2. Inventory,
  3. Motion (moving more than is required),
  4. Waiting,
  5. Overproduction,
  6. Over Processing (from poor design), and
  7. Defects (the effort involved in inspecting for, and fixing, defects).

Additional and new wastes are not meeting customer demand, and are a waste of unused human talent. There is further differentiation between Type 1 (necessary waste, e.g. government regulations) and Type 2 (unnecessary waste).

## An Overview of Kanban as a Software Development Method

The original concepts of Kanban (カンバン) were developed in the 1940s and 50s by Taiichi Ohno[[2]](#footnote-2) as part of the Toyota Production System, as a mechanism to control Just-In-Time (JIT) production and manufacturing processes. Kanban, which approximately translates as ‘signboard’, is described as a ‘visual process management system that tells what to produce, when to produce it, and how much to produce’. The modern Kanban method, as formulated by David J Anderson in 2007[[3]](#footnote-3), is an adaption of the original JIT approach, with an emphasis on staff welfare and continuous process improvement practices. This is ultimately a strategy that strives to improve a business return on investment by reducing waiting inventory and associated carrying costs.

At its simplest, each prioritised task (or card) on a Kanban Board passes through a visualisation of the team’s process, or workflow, as they happen. Each primary activity in the team’s workflow is visualised as columns on the Kanban Board, usually starting at task definition, and finishing with delivery to the customer. Of course, being Agile, these cards and activities are visible to all participants, including the customer.

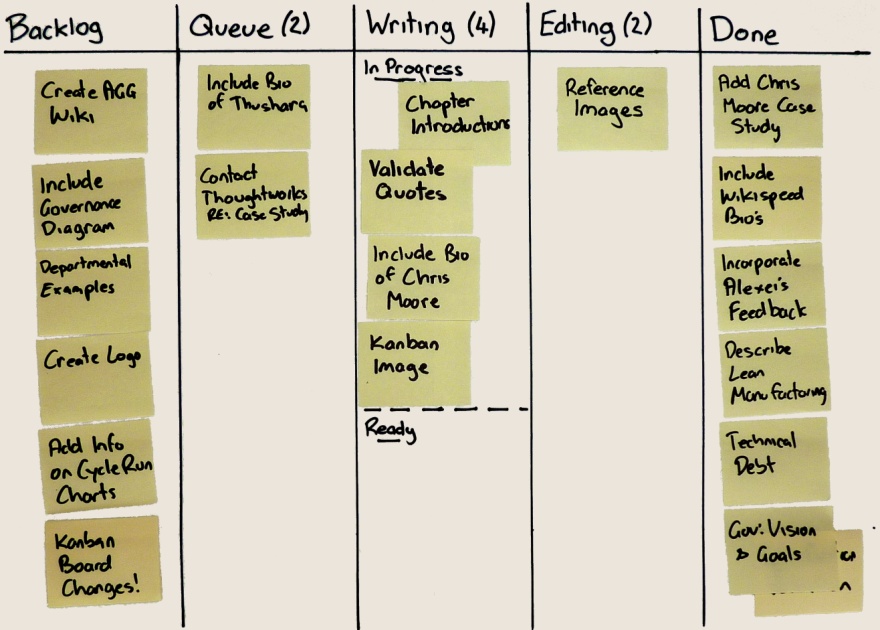


Figure : Example Kanban Board

While a Kanban workflow can become very complex, the simplest visualisation of workflow has only 4 different states which a task would progress through during its lifecycle. These states are;

1. Backlog – tasks that are waiting to be worked on
2. In Progress – currently being developed by a team member
3. Testing – undergoing integration, system or UAT testing
4. Done – complete and ready to be demonstrated and/or deployed

To identify, and control, bottlenecks and process limitations, each workflow state (or column) has a limit, called a WIP, or Work In Progress Limit, to the number of currently active tasks. This allows managers and team members to regularly monitor, and measure, the flow of work.

In total, there are 6 core elements to Kanban:

1. Visualise (Kanban / Card Wall)
2. Limit WIP
3. Manage Flow (and reduce bottlenecks)
4. Make Policies Explicit
5. Feedback Loops
6. Improve Collaboratively

# Kanban

‘Simplicity, carried to the extreme, becomes elegance.’

Jon Franklin, 1994

## Task Lifecycle

Agile processes are designed to promote a sustainable workload, where your teams, management, and customers, are able to maintain a constant pace indefinitely. Teams have the authority to design, plan, and estimate, each story, as well as the responsibility and accountability for delivery. This level of ownership for work, combined with integrated customer engagement, significantly improves workload management, which in turn reduces overtime and stress. However, for this to be efficient there needs to be a simple mechanism to manage, and level out, workflow within each team. This is where Kanban comes in.

Depending on your process workflow, a task will progress through a minimum of 4 different states during its lifecycle. Each task and state should be visible to the team, Customer Representative and customer; commonly this is done through a Kanban board (card wall) or integrated dashboard.

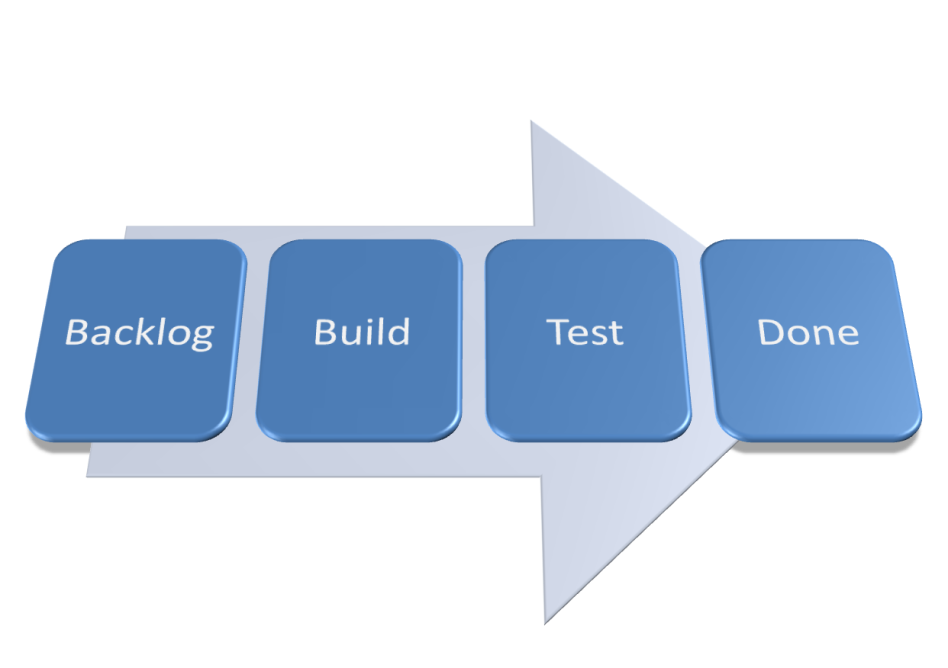


Figure : Basic Task Lifecycle

The Assignee for a task may change at any of these steps. It is important to understand that each state (and thus column on the Kanban board) represents a state within the development workflow, not a handoff between team members. Team members will proactively interact with their colleagues, and any internal parties, as required, to progress the assigned task to completion, including any quality control and review.

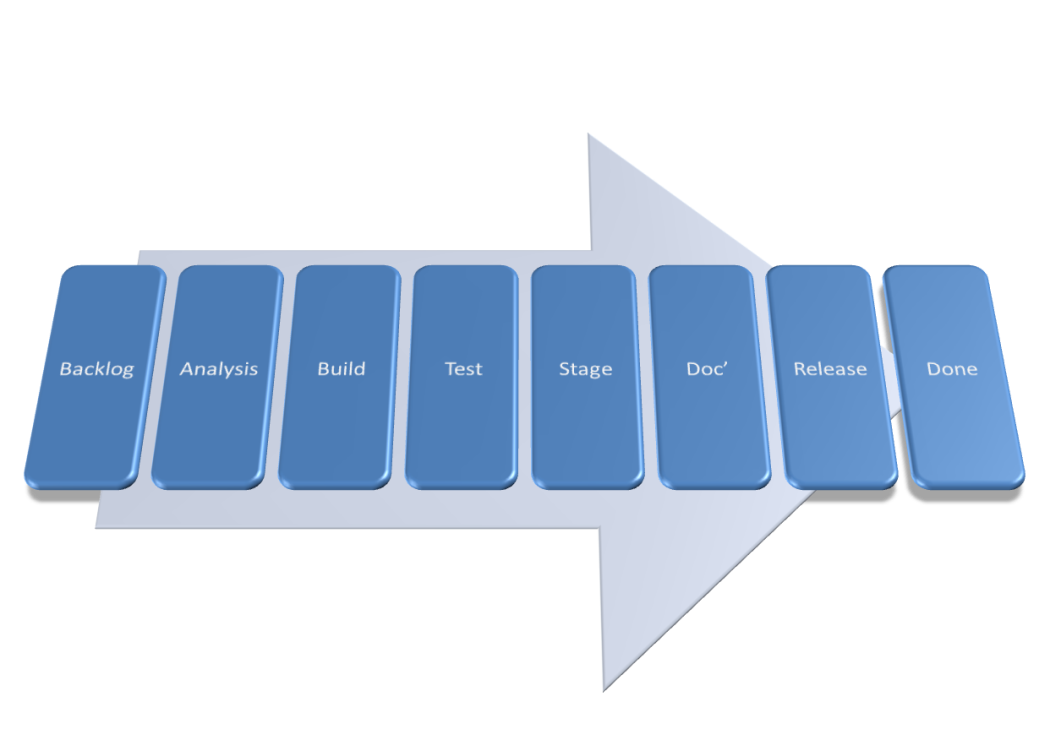


Figure : Complex Task Lifecycle

### Backlog

The Backlog is the list of user stories (or tasks) that the Customer Representative maintains. There may be a second backlog column “Ready” which contains those stories which have been technically designed and are ready to be developed. Based on the logical sequencing of tasks and agreed prioritisation, the project team members select the next task to work on and promote this to the “In Progress” state.

### In Progress

In Progress items are tasks that are actively being worked on. This includes both development and unit testing activities. Once the task has been completed it is promoted to the “Testing” state. In Progress tasks may include analysis, build, unit test and documentation.

### Done

Tasks are considered “Done” when, for example:

* Code has been produced, meets development standards, and has been checked in and run against current version in source control
* Unit and system tests written and passed
* Builds without errors for deployment
* Relevant documentation, including diagrams have been produced or updated

The decision over whether a user story is done is based on whether all the pre-requisite tasks associated with this story have been completed. The completed user stories are presented to the Customer Representative for acceptance and sign-off. This can either be done individually, or in a batch.

## Value Stream Mapping

Teams can create Value Stream Maps to defines the ‘As-Is’ steps & roles for each task lifecycle. That is, to create the workflow that will be used by the teams and visualised on the Kanban board.

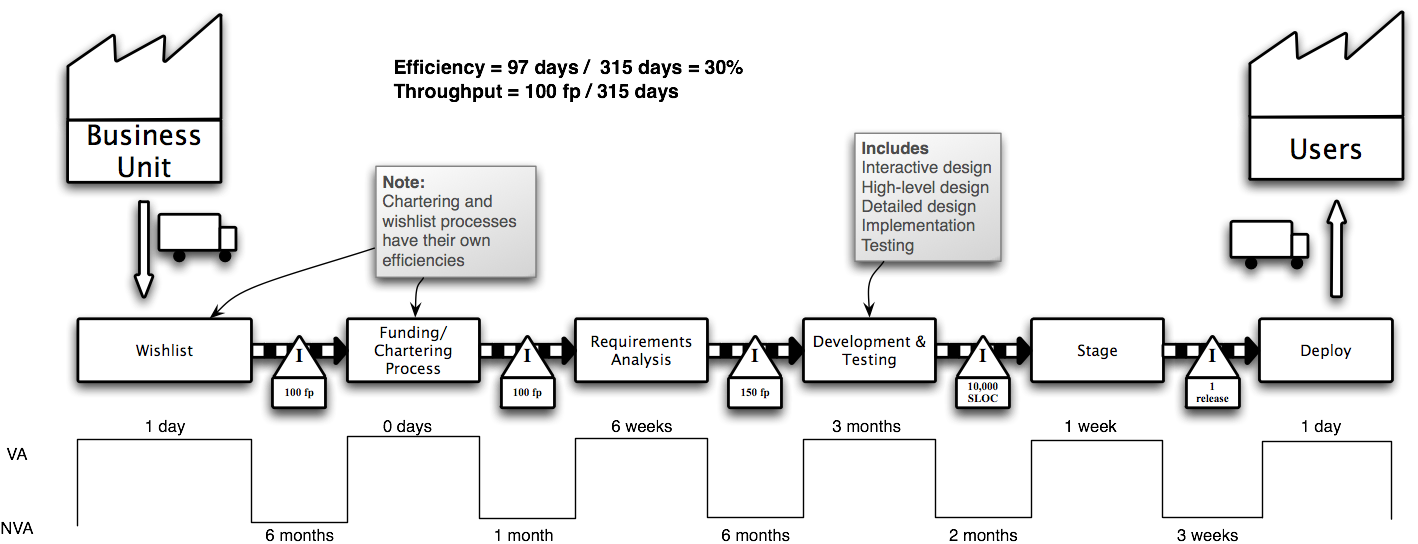


Figure : Example Project Value Stream Map[[4]](#footnote-4)

A value stream can be defined as all the steps – both value added and non-value added – required to take a product or service from its raw materials state into the waiting arms of a happy customer. Each step is defined, articulated and mapped on the value stream (above). The average time taken to complete each step (value add time) is measured, as well as the time taken to move between steps (non-value add time). In the example above, it takes 6 weeks to complete Requirements Analysis, and an additional 6 months from completing this to starting the Development & Testing.

Lean defines 10 steps needed to accurately define the value stream maps for each process within a product family group. If there is only one process to be modeling, you can start from step 5.

1. Gather Preliminary Information
2. Create a Product Quantity Routing Analysis
3. Group Customers and Sort Materials
4. Sort Product Families by Build Sequence
5. Choose One Value Stream to Begin With
6. Create an Operations Flow Chart
7. Walk the Shop Floor
8. Collect the Data
9. Construct the VSM
10. Summarize the Data and Get the Big Picture

Any process can be subject to a value stream map. Even eating cake;

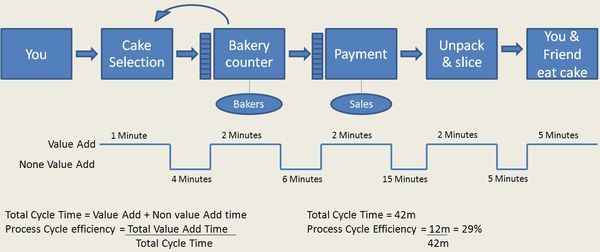


Figure : Value Stream Map for Buying and Eating Cake[[5]](#footnote-5)

By adding the total value add time to the total time (VA+NVA), it is possible to define the total process efficiency. In the example above, 29%.

## Kanban Boards

A Kanban board is a useful visualisation and control mechanism for both continuous product delivery. Starting at the Backlog, and finishing at done, each team will define the intervening states and workflows that make up the lifecycle (Value Stream Mapping) of their tasks (the Kanban). This can be as simple, or complex, as required. Teams working on several different types of tasks may have multiple Kanban Boards, visualising the different states and workflows for each type. When there is available capacity in a state, it will ‘pull’ any ‘ready’ tasks from the preceding state, thus moving tasks through the workflow.

The visualisation component, or cards, of a Kanban, helps identify the state of each task, when a task is ready, where there is spare capacity, and if there are any bottlenecks or impediments.



Figure : Example Kanban Board[[6]](#footnote-6)

Tasks or stories that have identified defects, need rework, or have upstream or downstream dependencies external to the team that are preventing progress, are marked as ‘blocked’, but do not change state. By leaving the task in the current state, the team can see where the blockage is, and identify where the process should resume once it is resolved. Similarly, by making all blocked tasks and stories visible on the Kanban board, customers and management are aware of the issues, and this simplifies any escalation and resolution.

Each task and state should be visible to your teams, customer, and customer representative; commonly achieved through a physical Kanban board. Each card describes a single task or story, as well as its estimate, and who is currently working on it. Keep cards simple, with additional information stored elsewhere. Divide the Kanban board into multiple, labelled columns, each representing a single state. Then further divide each column in half, the first half being ‘In Progress’ and the second half being ‘Ready’.

Some versions of Kanban also provide a single ‘Expedite’ track, at the top of the board, for urgent stories and tasks. There can only ever be one card at a time in this track, and it has the highest priority, above all other cards.

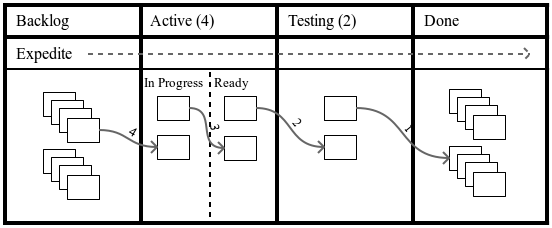


Figure : Kanban Board and Flow

Except for the Backlog and Done states, the number of cards allowed at any single time, in each state, is restricted. Called the Work In Progress (WIP) Limit, it includes both the ‘In Progress’ and ‘Ready’ Cards in any state, and matches the team’s work capacity. In general, smaller WIP Limits reduce lead times, help uncover problems in the workflow, and drive continuous improvement (Kaizen), whilst higher WIP Limits help to absorb variation and exploit opportunities. Teams using pair programming will have lower WIP Limits, as there is less simultaneous work in progress. By experimenting with various WIP Limits, you can identify, and resolve, process bottlenecks, adjust the impact of lead time, and create a predictable and efficient workflow. As a rule, your WIP Limit is too low if you hit bottlenecks every week, and too high if blockers don’t cause an issue.

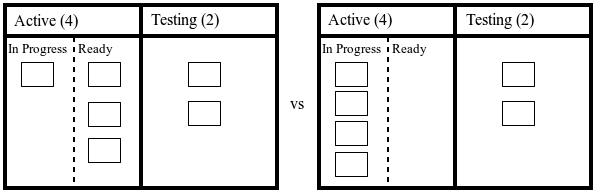


Figure : Bottlenecked vs. clear Kanban Boards

Kanban applies a form of Production Levelling to the process flow. This ensures that each step in the production process delivers at a constant rate and that no step in the production process produces work at a faster rate than subsequent steps. Additionally, Kanban uses the five focusing steps from the Theory of Constraints as the incremental process improvement model. These are:

1. Identify the system’s constraints: A bottleneck is an extreme example of a constraint. The slowest part of any process, no matter how smoothly it is working, will limit the throughput of the rest of the process.
2. Decide how to exploit the system’s constraints: Keep the constrained state focused and busy at all times, by focusing on value adding work, removing impediments, and providing high-quality tools and materials.
3. Subordinate everything else to the above decision: All other states should not produce more than the constraint can process. This means that they have available capacity that can support the constrained state to focus on its core responsibilities.
4. Elevate the system’s constraints: Once the constrained state has been fully exploited, the team, or organisation, needs to invest in additional capability, in order to increase its overall capacity.
5. If, in a previous step, the constraint was broken (e.g. it is no longer a constraint), go back to step one: At this point, the overall system throughput will have increased, and the system constraint will move to a new point. This encourages continuous improvement (Kaizen) within each team’s processes.

Note: Each team will go through different iterations of their Kanban board, as their workflow and work processes improve and evolve. It is not unusual for a productive team, one that has embraced continuous process improvement, to go through several Kanban board designs a year. A team that does not change their Kanban board is probably only using it to track work, not as a method for improvement.

## Inspection

As well as visualising progress, it is important to measure the time a task or story sits in the Backlog before being actioned, and the time task or story sits in each state (e.g. the time taken to move from Active to Testing, and from Testing to Done). The primary measures for this are Lead Time and Cycle Time. Lead Time and Cycle Time do not measure effort, but the elapsed time (or duration).

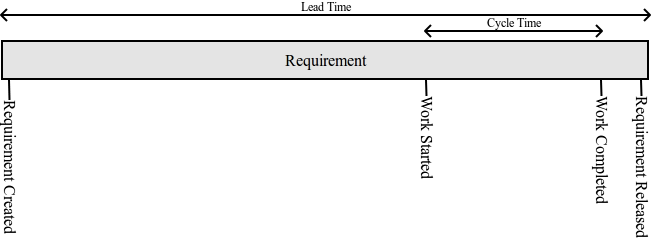


Figure : Lead Time vs Cycle Time

Lead Time is defined as the time taken between adding a story to the Backlog, and releasing it to the customer. Whereas, Cycle Time is defined as the time taken between starting, and completing, work on a story.

But, numbers alone don’t provide enough useful information to manage your teams. Using Cumulative Flow Diagrams, and Cycle Time Run Charts, you can represent, and visualise, the scope of work, planned delivery, and actual delivery of tasks and stories. To ensure full transparency between your teams and customers, these charts should be available to everyone even remotely involved with the team.

## Cumulative Flow Diagrams

A Cumulative Flow Diagram (CFD) visualises the flow of work over elapsed time and the number of tasks or stories remaining in each state. This is important in verifying the efficient delivery of work.

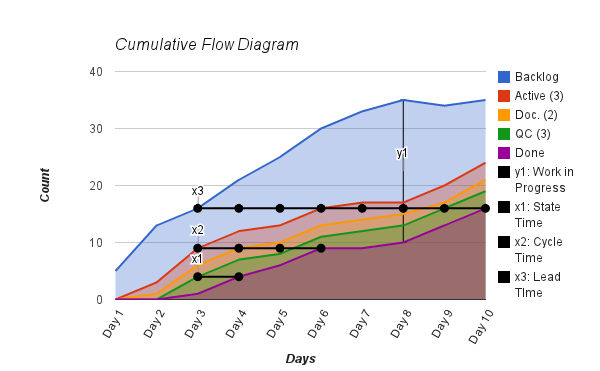


Figure : Example Cumulative Flow Diagram

The CFD tracks each state in your workflow separately, from when a card (or task) enters that state, to the time the card enters the subsequent state. The vertical distance (y1) in each charted line shows the number of tasks currently in progress. This distance should never be greater than the WIP limit for the state. The horizontal distance (x1) shows the time taken for a task to progress to the next state. The horizontal distance (x2) shows the Average Cycle Time, the time taken from when a card leaves the Backlog state, until is it done. The final horizontal distance (x3) shows the Average Lead Time, the time taken from when the card enters the Backlog, until it is done.

Each line on the Cumulative Flow Diagram should appear smooth; any flat vertical or horizontal generally indicates impediments, or an uneven flow of work. You can quickly, and easily, identify bottlenecks, when the area between two bands narrows or, in the worst case, reduces to zero. Keeping low WIP limits simplifies the identification of bottlenecks, when analysing Cumulative Flow Diagrams.

|  |  |
| --- | --- |
| Bottleneck **Identified by:** A band reducing to zero.  **Identified issue:** The Documentation state is a bottleneck in the process, and has starved the Quality Control state of any work.  **Resolution:** Improve the delivery through the bottlenecked state by exploiting, subordinating, and elevating the constraint. | J:\Downloads\cfd_bottleneck.png  Figure : Problem CFD (Starved state) |
| Poor flow **Identified by:** Jagged, widening, and narrowing bands, between two or more states.  **Identified issue:** Caused when there is not a smooth flow of work through the system. States that jump to the maximum WIP, and back down again, can also be indicative of bottlenecks, or other impediments, throughout the work processes.  **Resolution:** Identify the cause of the impediments of bottlenecks, and remove them, to improve the flow of work. | J:\Downloads\cfd_poor_flow.png  Figure : Problem CFD (Poor flow) |
| No, or large, wip limit **Identified by:** A large distance between each band, causing a false sense of smoothness.  **Identified issue:** This example actually shows the same variation as the ‘Bottleneck’ example above. However, as the WIP Limit is very large, it is difficult to identify any fluctuations in the chart.  **Resolution:** Reduce the WIP Limit to an appropriate number. | J:\Downloads\cfp_no_wip.png  Figure : Problem CFD (Large WIP) |
| Long lead time **Identified by:** A very slow, and shallow, rise in all states.  **Identified issue:** There is a long lead time (and cycle time) between raising a story, and it being delivered to the customer.  **Resolution:** Reduce the WIP Limit, or reduce the size of the stories, to improve the speed of the workflow. | J:\Downloads\cfp_long_lead.png  Figure : Problem CFD (Long lead time) |
| Plateau **Identified by:** Tracking well, followed by no visible progress for several days.  **Identified issue:** The flow of work has stopped (or dramatically slowed), usually caused by critical production issues, large-scale staff absences (e.g. Christmas holidays), or waiting for customer sign off.  **Resolution:** Identify what is causing the halt of workflow, and (if appropriate) resolve the underlying issue. | J:\Downloads\cfp_plateau.png  Figure : Problem CFD (Plateau) |
| Nothing wrong Tracking well in terms of consistent rise, and no major widening or narrowing of bands. | J:\Downloads\cfd_nothing_wrong.png  Figure : Normal CFD |

## Cycle Time Run Charts

Teams that measure Cycle Time and Lead Time can visualise these metrics using Cycle Time Run Charts (sometimes known as statistical process control charts). By looking for trends, cycles and outliers above expected tolerances, Cycle Time Run Charts will help you to identify both normal, and uncontrolled, variations in process flow.

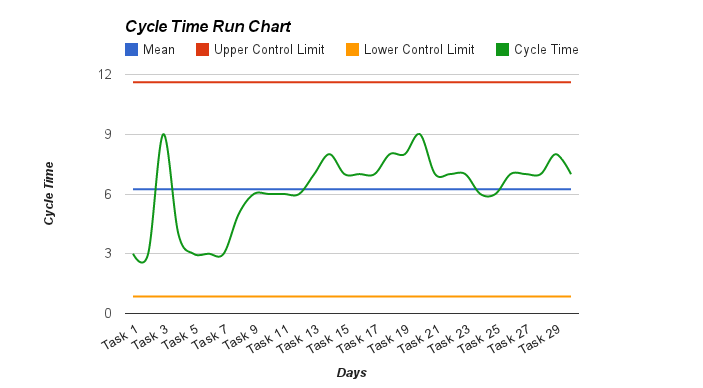


Figure : Example Run Chart

Run Charts plot the Cycle (or Lead) Time of each story, against the long-term average, known as the centre line. From a continuous improvement perspective, you should aim to improve your work processes, so the centre line (and thus your average Cycle Time) meets your customer’s needs.

If you know the expected variance within your process (usually ± three standard deviations), you can plot the Upper Control Limit (UCL) and Lower Control Limit (LCL). These limits are the primary mechanism to identify special cause events. This means you should investigate anything beyond the limits, as they can indicate a process out of control.

It is simple to calculate Cycle Time Run Charts when all stories are of approximately equal size and effort. However, they can still be effective for stories of varying sizes, but will have higher Control Limits, and need a larger dataset to calculate a meaningful average.

|  |  |
| --- | --- |
| Process trend **Identified by:** A run of points that are continuously increasing or decreasing.  **Identified issue:** There is a progressive trend in the Cycle Time, implying that something is gradually shifting over time.  **Resolution:** If the trend is in the right direction (usually down), it may be part of ongoing process improvement, and your team should sustain the change. Otherwise, your team needs to determine the cause of the variation, and resolve it. | Figure : Problem Run Chart (Process trend) |
| Process shift **Identified by:** A run of points on a single side of the centre line.  **Identified issue:** There is a sustained shift in Cycle Time, and may have reached a new equilibrium.  **Resolution:** If the shift is in the right direction (usually down), it may be part of ongoing process improvement, and your team should sustain the change. Otherwise, your team needs to determine the cause of the variation, and resolve it. | Figure : Problem Run Chart (Process shift) |
| Extreme process variation **Identified by:** Points above the UCL, or below the LCL.  **Identified issue:** Extreme variation (special causes) in the team’s process that may indicate a process out of control.  **Resolution:** Identify the cause of the outliers, and if systemic, resolve the underlying issues. | Figure : Problem Run Chart (Extreme process variation) |
| Nothing wrong No major variations in process flow. | Figure : Normal Run Chart |

# Kaizen

‘Fall seven times. Stand up eight.’

Old Japanese Proverb

All Agile processes are, by their very nature, cyclical, based on the inspect and adapt cycle. With very little additional effort, this continuous feedback can become ongoing improvement, and provide the mechanism for organisations to adapt to changing circumstances. This process of continuous improvement, or Kaizen, leads to a culture of continuous improvement.

There are five main elements to Kaizen:

1. **Teamwork:** All staff, regardless of rank or status, work together towards the common goals of your organisation. By extension, all staff, across every team, need to understand the implications of their work for the rest of the organisation, and share the responsibility for your organisation’s success.
2. **Personal discipline:** Staff should be accountable for their actions. Not only for their core responsibilities, but for all aspects of their work, including quality control, time management, and their professional relationships with colleagues and customers. There is a corresponding requirement for organisations to set reasonable standards, and challenge staff to meet them.
3. **Improved morale:** Teams share responsibility to build an environment where they feel empowered, secure, and have a sense of ownership. An organisation with low morale, or conflict between managers and staff, will suffer from high absenteeism, poor engagement, and reduced productivity.
4. **Quality circles/Retrospective workshop:** The retrospective workshops are the primary forum for teams to suggest improvements to your corporate culture, delivery processes and management arrangements. Teams should be encouraged to hold cross-team retrospectives, as a means to share ideas, skills and technology improvements. Teams also need the authority to experiment with, and implement, local changes, and the organisation should be quick to respond to any large-scale suggestions that have implications beyond the team.
5. **Suggestions for improvement:** All business functions are candidates for Kaizen, and, as such, each team member has an obligation to participate in the continuous improvement process. Learning, observing, and putting forward new ideas, especially in relation to their core responsibilities, will help remove any impediments, and increase work efficiency.

Kaizen is a truly continuous process, where teams should be seeking new ways of improving the business every day. Staff should be encouraged to experiment with different process changes, to drive continuous improvement.

The regular (weekly or biweekly), retrospective workshop, provides a formal forum for each of your teams for introspection, and reflection on the processes that support their development. The goal of this workshop is to suggest improvements, and the focus should be, in order of importance, people, relationships, process, and tools. The full team should be present for each retrospective, as they are ultimately responsible, and accountable, for driving process improvements in their team.

Teams should be free to discuss any relevant topic. During this short meeting, between an hour to half a day, the team should reflect on the processes since the last Retrospective. This may include:

* Discuss the processes that worked well, and were effective, or improved, since the last retrospective. By reflecting on the positive outcomes, the team can identify their strengths, and use those to overcome specific weaknesses. It is also important, as a mechanism, to provide positive feedback to the team.
* Discuss the processes that did not work as well as expected, and need improvement. By reflecting on the negative outcomes, the team can focus their effort on improving in that area, or make modifications to the process to better play to their strengths.
* Suggest any specific improvements to the processes used within the team. As mentioned, continuous improvement (Kaizen) is a core concept to Agile, and the team is responsible for driving most of this change. It is important to ensure that each improvement is actionable, and assigned an owner.

At the end of the Retrospective, the team should have a list of ‘assigned’ and ‘actionable’ improvements to the management processes.

Each retrospective provides the team with the opportunity to reflect on the time since the last retrospective, and drive continuous process improvement out of any learning’s since then. Through this process of Kaizen, the delivery of each story should be more effective, and enjoyable, than the last.

## 5S

5S is an approach to waste reduction, quality & continuous improvement (housekeeping) defined by the lean approach. The 5 S’s are:

* **Seiri (Sort):** to clean and organise the work area
* **Seiton (Set in Order):** arrange the work area to ensure easy identification and accessibility
* **Seiso (Shine):** mess prevention and regular maintenance of the work area
* **Seiketsu (Standardize):** create a consistent approach for carrying out production processes
* **Shitsuke (Sustain):** maintain the previous 4S's through discipline and commitment

1. Images with thanks from Jeff Patton: <http://www.agileproductdesign.com/> [↑](#footnote-ref-1)
2. Toyota Production System: Beyond Large-Scale Production, Ohno (1988). [↑](#footnote-ref-2)
3. Kanban: Successful Evolutionary Change for Your Technology Business, Anderson (2010). [↑](#footnote-ref-3)
4. Image thanks to: http://www.nexusis.com/solutions/cloud/cloud-consulting/ [↑](#footnote-ref-4)
5. Images thanks to: http://leadinganswers.typepad.com/leading\_answers/2011/09/pmi-acp-value-stream-mapping.html [↑](#footnote-ref-5)
6. Image thanks to: New Zealand Postal Group via Directing the Agile Organisation [↑](#footnote-ref-6)