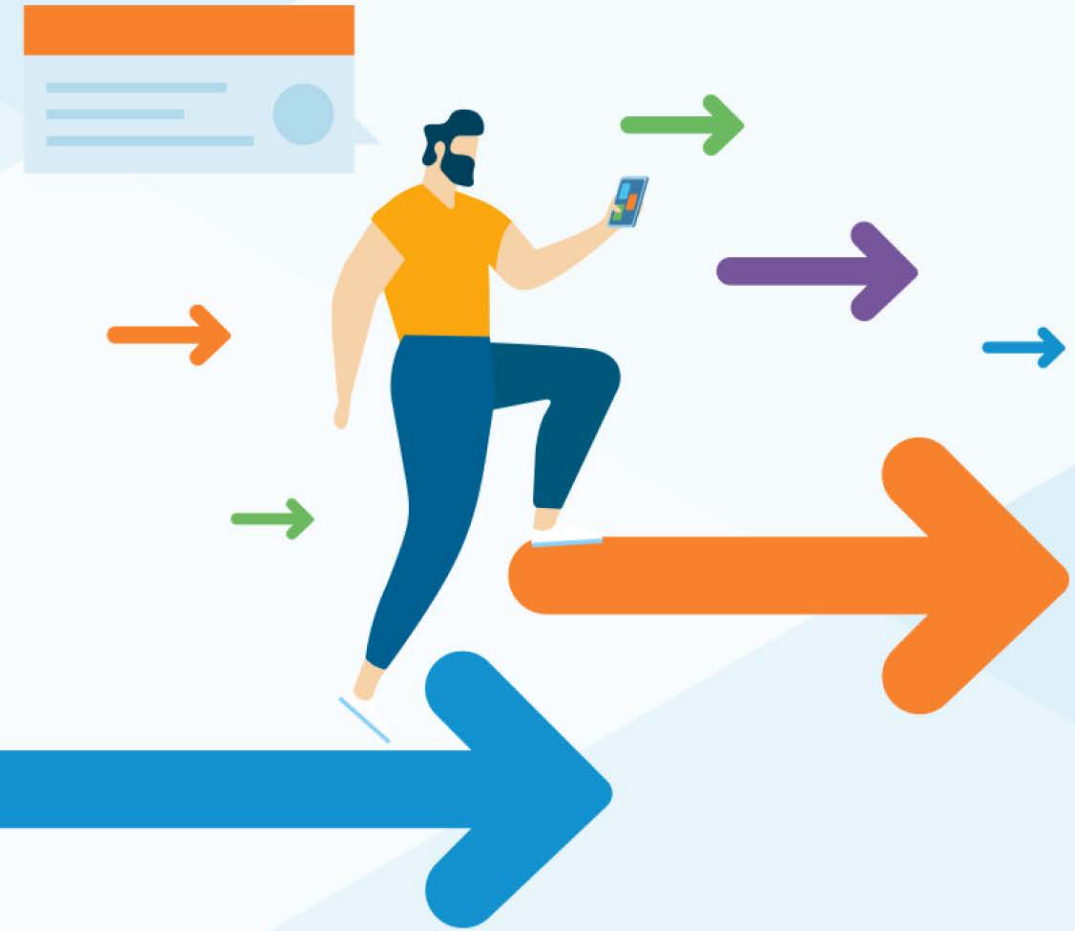
Project Manager's Guide to **Kanban**



Dimitar Karaivanov
Kanbanize CEO and Co-founder

kanbanize

Project Manager's Guide to **Kanban**

What's in it for you?

Project Manager's Guide to Kanban provides valuable insights on Kanban Project Management and will foster your theoretical learnings with practical examples. It is structured in six chapters, starting with outlining the basics of Kanban and the benefits of scaling it from team to management level. Then, it will explore topics such as Planning, Execution, Tracking and Forecasting, Risk Management. In conclusion we will focus on the overall value and the outcomes that you could expect.

As modern Project Management has realised the necessity for more lightweight and efficient techniques, we focus on the benefits of choosing Kanban. Join the next generation Project Managers and help your team minimize waste and increase efficiency through continuous improvement.

Introduction to Kanban for Project Managers with practical implementation examples.

Preface

In the modern world of globalization, disruptive technologies, and automation, we need to adjust our behavior to handle all this innovation and knowledge efficiently. The new wave of Project Management is not something that we should expect, rather, it is already here. In the quest of processes optimization, there is a hunger for modern lightweight approaches that would help Project Managers acquire almost superhero characteristics.

Well, there is no magic framework that would turn Project Management into a safe playground but we have the remedy for those who are tired of supporting the “house of cards”. Riding the wave of Kanban Project Management would transform the way this industry looked like a few years ago. And if you think this is a bold statement, let’s go through the following chapters and follow the practical implementation examples which will shed light on the truly revolutionary approach.

We have outlined the main principles and practices of Kanban, the benefits it will bring and the difficulties that may arise during the implementation. One of the most important aspects is how to scale a successful practice from a team level to the the management level and we will explore it in detail. We will discuss what needs to change in the Planning, Execution, Tracking and Forecasting, and Managing Risk.

There is a better way to manage projects, the Kanban way!

If you are ready to apply kanban principles to your work, I recommend you try Kanbanize for free.

Try Kanbanize for free

CHAPTER ONE

Why Bother with Kanban?

Project management has been around for decades, if not centuries. It flourished throughout the industrial revolution and became an instrument for the astonishing progress humanity has achieved.

Most of the world around us is the result of many smart people managing projects and programs. We see it in every building, vehicle, desk, chair, window… they were all the result of projects well managed (or not that much).

However, a lot has changed since the industrial revolution. The factories that used to harness human labor are now powered by robots, infinitely more productive than the human hand.

Computers became mainstream, we invented the mobile phones, the internet, AI, machine learning, blockchain and what not. If that wasn't enough, globalization took over and that completely disrupted the way companies do business.

Compared to the 50s, the world has become a much more complex place…

In this new paradigm, the effectiveness of our management processes has continuously degraded. This dissonance created a vacuum that sucked up billions of dollars in failed projects or missed opportunities. The generated pain and disappointment ultimately led to the creation of new, more effective methods that were better suited to operate in an environment of uncertainty and complexity.

In just 20 years, the so-called Agile methods managed to dominate the IT and Software Development spaces. We saw the creation of frameworks such as XP and Scrum that are widely adopted and are quite the norm for tech teams in most of the companies today.

Some ten years back, a very important event took place - David J. Anderson published the first book about Kanban (also known as the blue book). This marked the birth of a strong movement that led to the existence of the "Kanban method" the way we know it today.

The Kanban method is truly remarkable. In less than a decade, it was adopted by more than half of the IT / Software industry. A recent survey ("State of Scrum") shows that 60% of the Scrum teams use Kanban. This data is stunning because only several years ago, Scrum in its pure form had virtually no alternative.

The reason why so many teams and companies are turning to Kanban lies in the core principles and practices of the method.

The Kanban Principles

There are three guiding principles that represent the Kanban way of thinking:

1. Start With What You Do Now
   - » Understanding current processes, as actually practiced
   - » Respecting existing roles, responsibilities & job titles
2. Gain agreement to pursue improvement through evolutionary change
3. Encourage Acts of Leadership at All Levels

If you compare these principles to any other popular approach, you will quickly realize that **Kanban is very humane**. It doesn't tell you that there's a right way of doing things and that you are doing it wrong. On the contrary, it respects the current situation and helps you improve from where you are. There is

no judgment in Kanban!

In that regard, Kanban is a truly Lean method, following the principles of continuous improvement and respect for people. The principles we know so well from Toyota.

The Kanban Practices

Going from the philosophy of things (principles) to what we do on a daily basis, we identify six core practices in Kanban.

- Visualize work
- Limit Work in Progress
- Manage Flow
- Make Policies Explicit
- Implement Feedback Loops
- Improve Collaboratively, Evolve Experimentally

The goal of this book is not to explain what Kanban is so we won't discuss all the practices individually. However, we will be providing concrete examples further in the book. One thing is certain – if you use these practices you will be much more productive in your daily work, be it management, coding, design, etc.

If you're still wondering if Kanban is for you, have a look at the benefits of using the method:

- Dramatic improvements of your cycle times realized soon after starting out. The improvements usually go way beyond 100%.
- Relief from overburdening and happier workforce.
- Increased customer satisfaction due to shorter delivery times and increased predictability.
- Compatibility with any process. Kanban improves the flow of value without forcing the organization out of its ways.
- Universally applicable – it is not bound for use by the IT or software development teams only.
- Survivability and sustainability of the business, if applied strategically.
- Understanding of the work state and project status.
- Qualitative and quantitative understanding of the workflow.
- Increased predictability and an ability to respond with confidence to tough questions like "When will you deliver?", "When should we start a job to deliver by a certain date?".

Do you have issues with one or more of the items above? If yes, this book is for you. We aim to provide practical guidance for Project Managers that are seeking ways to improve their workflows without necessarily adding more people or resources.

We will show you how to plan your project in an alternative way and how to connect the planning to the execution. We will also suggest a scalable way to track one or multiple projects

using hierarchical Kanban boards. On top of that, we will discuss forecasting that's based on the actual data that you generate as the project goes. If you're ready to get started with a kanban tool, you can try Kanbanize for free.

It will be quite a ride, we hope you'll enjoy it!

Try Kanbanize for free

CHAPTER TWO

Why Kanban Project Management with Kanbanize?

This book talks about the Kanban method but it does so through the prism of Kanbanize being a supporting tool. We believe that it is not possible to effectively manage even a single project without the proper software solution and we won't separate the theory from the practice. We must accept the reality: work is getting more and more complex, and that requires more sophisticated and capable tools to manage our projects.

You may think that all we want to do is sell you our software but we're truly driven by the desire to improve how projects

are managed worldwide. We want to give companies the tools to innovate faster and by that change the world. All that comes from the desire to make our civilization sustainable, something that we struggle with badly. We see Kanban and Kanbanize as our small contribution to a cleaner planet where global warming, hunger, and war are words left in the past.

We know this is possible because Kanbanize was born out of necessity. We saw with our eyes how applying Kanban across the whole organization can turn a tiny startup into a leading software vendor. That's why we decided to create a solution that helps managers replicate that success, irrespective of the industry or the company size.

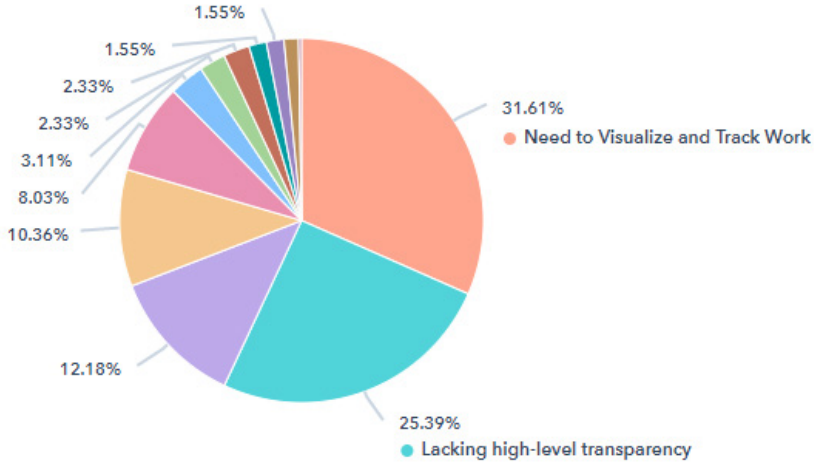
Kanbanize is the next generation project management software because it solves some of the fundamental problems of today's management and adds the Lean/Kanban concepts on top of that. This is a unique combination with no real alternative in the world as of now. If you are not convinced yet, you will be by the end of this book.

To support these claims with something real, let's discuss the challenges project managers and their teams face every day.

Lack of transparency and visibility into who is doing what

At Kanbanize we talk to thousands of customers every year. You will be amazed by the number of people who come to us

just because they want to know what is going on. We even have the data to support this claim:



Almost one-third of all the people we've talked to turned to Kanbanize because they needed better ways to visualize and track the work.

One-quarter of all said they were lacking higher level transparency.

The widely accepted management ways have brought us to the state where more than half of the companies simply don't know the real status of their work. Isn't this alarming and even disturbing?

And, please, don't say that these are bad companies skewing the data. These are all sorts of companies, including many fortune 500 businesses. It's a big data sample too (more than

20,000) so this issue is very, very real.

Disconnect between planning and execution

This is an issue that we see far too often. Simply put, it is caused by the fact that the project plan and the actual work live in two separate systems. These are typically MS Project holding a nice-looking Gantt chart and a work management tool of choice for the day-to-day work items.

This is a huge problem as it requires never-ending updates of the two separate systems, so that they are in sync. We will not even go into the problem that the plan is usually rendered invalid on the very next day, when something unexpected happens.

The bigger issue here, though, is the lack of feedback loops between the teams who are actually doing the work and the management chain. Unexpected situations happen all the time when you're dealing with knowledge work. It just happens that a piece of work is more complex than expected, or the integration of two systems does not go well. As we call it, life happens.

When life happens, management usually learns from a report or a status meeting. But what if life happened on Monday and the status meeting is on Friday? Should we wait a whole week to reflect the change in our plan? What if it's once per month

or once per quarter?

And what if a problem in one of the teams is hindering the progress of a strategic initiative three levels up in the hierarchy? How is this VP or C-level person supposed to know? How much time will pass before they realize that there is a problem? What if they could solve it instantly by allocating more budget or re-prioritizing some of the key initiatives?

Feedback loops and instant information availability are key to the successful management of an Agile organization. When things can change rapidly you need to make sure you have the means necessary to provide a swift response to the situation.

Unrealistic capacity expectations

You might not believe that, but project managers rarely know the real capacity of the teams they work with. If this wasn't so, all projects would have been on time and within budget, something that hardly ever happens.

There are several reasons why projects can't fulfill the original plan, even though all projects have a buffer (sometimes as high as 50%). One of the biggest issues here is estimation and the false sense of certainty. Somehow, we believe that when we've estimated every single task in the project, we know what's going to happen but nothing could be further from the truth. Estimations are subjective and no matter how much

experience you have, if work is not repetitive, you are bound to be off. How much off you are depends on personal qualities and luck.

As much as we want to be able to predict the future, we can't. As much as we want our plan to succeed, it's inaccurate by definition. The only thing we can do about it is to base our forecasts on actual historical data, which reveals our true capacity, and then be swift to respond to the slightest deviation. To do so, we need the proper tooling that provides the historical data and that is capable of detecting these deviations proactively.

Push is the predominant approach

Many organizations manage their work on the basis of Gantt charts and strict schedules. This would have been fine if we knew the true capacity of our teams at any given moment in time. However, this is not the case and we already discussed it in the previous point.

When you manage work based on a schedule, you build a parallel reality, which is not the one you're currently having. If the plan says that you should start working on B but the team is still starting A, what should you do?

If plans are followed blindly, the manager will force the teams to start working on B, even though they have not yet finished

A. This is known as a PUSH system.

If the manager is experienced in the realm of Lean and Kanban, they will ask the team to first finish A and then go to B. Respecting the current capacity and scheduling work accordingly is known as a PULL system. Something that we know so well from TOYOTA.

Unfortunately, we still see that PUSH is the dominant approach. This is a trend that we hope to see reversed in the upcoming years.

All of the challenges listed above create overhead for everyone involved in the project and generate monstrous amounts of waste. Waste in your processes makes you less competitive, so it's something to be avoided at almost any cost.

As you've probably already guessed, Kanban by itself does not offer a solution to these problems. As a matter of fact, no Agile method or framework has ever provided guidance on how to solve these problems.

To be fair, we must acknowledge that there are frameworks, such as SAFe, that claim to be solving these challenges, but they require an enormous investment and generate even more waste by changing your entire organization at once.

This creates significant stress for your teams and is not a recommended approach to start with. You may end up doing everything that SAFe suggests but you should get there step by step, following the evolutionary change practices of Kanban.

Struggling with these challenges for years, we decided to end the suffering once and for all. That's how we created Kanbanize - we wanted to simplify the life of both managers and teams by providing a solid Kanban platform for project management that is capable of:

- Visualizing work across one, two or more hierarchical levels and ensuring that the necessary dependencies are visualized.
- Connecting planning with execution by providing automated data-driven feedback about the feasibility of the plan.
- Making capacity allocation explicit and well-understood by all stakeholders.

Throughout the rest of this book, we will discuss the concrete steps that you can take to solve these problems for good. Some of the ideas might sound too radical, but they work. Just make sure you bring your open mind with you!

CHAPTER THREE

Project Planning

In some companies, planning means creating a definitive work breakdown structure and assigning start/end dates to the individual activities. This kind of planning is suitable for contexts with high certainty, e.g. manufacturing, construction, etc. However, this approach fails big time in knowledge work.

Deterministic vs. Probabilistic Thinking

Let's have a quick detour and talk about probability. Probability is a strange thing because it's natural for some people and very distant from others. If you ask a hundred people how many strawberries there are in a kilogram, some will come up with a concrete number and others will offer a

range.

The same will happen if you try to guess how much time a task will take. Some will say 6 hours, others will say 4-8 hours. The ones that specify the number think deterministically while the people providing ranges are closer to what we call probabilistic thinking.

Deterministic thinking in knowledge work must be avoided. If you are a developer, a marketer, an IT engineer, a designer or anything like that, you don't know how much time exactly you need to complete a given job. If asked for an estimation, you should always provide ranges instead of precise numbers. This is probabilistic thinking in its pure form.

Bear in mind that this book won't ever consider deterministic thinking. Wherever you see the words "plan", "estimation", "forecast", etc. assume there's a probability attached to them, even if it's not explicitly mentioned.

Planning with Kanban

Transitioning from traditional Project Management to Kanban Project Management might be challenging. We often hear that managers refuse to go the Kanban way because "they cannot plan properly".

This statement is plain wrong because you can most definitely plan with Kanban. It is actually advisable that you plan whatever you're doing. As Dwight D. Eisenhower says *"In preparing for battle I have always found that plans are useless, but planning is indispensable"*. However, Kanban and the good old Gantt chart don't mix well together. Here's why.

In its core, Kanban is a PULL system. A PULL system allows new task/project to be started only when a task/project has been finished. In other words - when there is free capacity. Imagine a rope in a long tube. To run the rope through the tube it's easier when you're pulling it and not pushing it. When you push, it gets tangled inside the tube and eventually gets stuck. That's what happens in a PUSH system - work gets stuck and takes forever to finish.

The Gantt chart is an amazing visualization tool but it's just not fit for planning purposes in knowledge work contexts. To build such a chart, you need a start and an end date for all the activities involved in the plan. However, assigning start and end dates to tasks represents a PUSH system because it ignores the availability of capacity - it just assumes it will be there. This approach fundamentally contradicts to the PULL lean principle and should not be used for planning.

But what if we told you that you can have a Kanban gantt chart? What if we could have a plan that looks like a traditional plan but doesn't force us into a PUSH system? Yes, that's

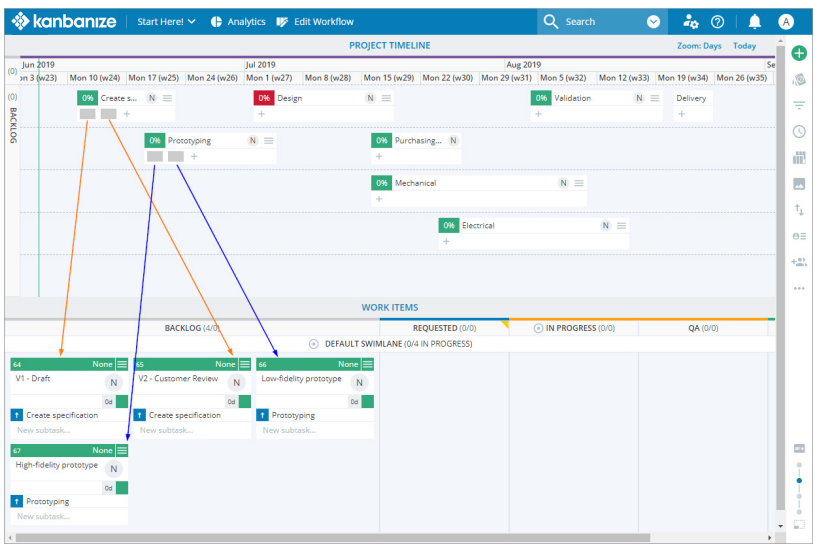
possible and Kanbanize is the only software in the world to provide such a solution.

The Kanban Board as a Planning Tool

Acknowledging that plans are an essential part of every management method, and even life in general, Kanbanize offers a lightweight approach to planning.

To realize a Kanban project plan, all you need to do is add a "Timeline" workflow to your board. Typically, you put the Timeline at the top of the board.

Take a look at this sample image:



As shown above, the board is split in two areas horizontally. The top area represents a special type of workflow that visualizes the project deliverables on a timeline. We call these deliverables “Initiatives” as they represent bigger work items and not day-to-day tasks.

In the example above, the Initiatives on the timeline represent the key deliverables of a project to build custom machine (Create specification, Prototyping, Design, etc.).

Once the project plan is in place, the Initiatives are broken down into tasks that are placed in the bottom part of the board where we have another type of workflow “Cards Workflow”. The card’s workflow is meant to be used for tasks that the team works on every day. The tasks in the “Cards Workflow” are children of the Initiatives in the Timeline.

Before you think this just a timeline component like any other you’ve seen, let’s quickly clarify some points:

- The only way you can start an Initiative is by starting its first child card.
- The only way to complete an Initiative is by finishing all children cards.
- The timeline workflow in Kanbanize can have a WIP limit. If you want to limit the number of parallel Initiatives you work on, something that we highly recommend, you will be focusing your team’s’ energy onto a small number of items and thus deliver them faster. This is a key element in Kanban and a unique feature of our platform.

- The Initiatives that are put one after the other on the same track in the timeline are automatically liked as predecessor / successor. This means that you won't be able to start working on any of the children of the successor initiative unless the predecessor's children have all been completed first. With that we ensure the proper sequencing of work on the team level.

Something that is currently work in progress (as of June 2019) but we expect to have by the end of the year is a revolutionary continuous forecasting module that will validate your plan in real time, based on the status of all children tasks. The continuous forecasting module will be able to:

- Show you the probability of your team matching the planned start date of the initiative. In other words, the software will tell you if this initiative is really going to start when you think it will start or not.

- Show you the probability of the initiative meeting its planned end date. You may think that the work will finish on time but the software will be able to give you precise information about that and it's going to be based on your actual progress, not estimation. As a matter of fact, no estimation at all will be required, as the software knows how much time it typically takes your teams to deliver similar tasks.

With this plan, which is connected to the actual execution you will be able to see in real time how everyday decisions affect your plan. Your teams will be informed about the order of work and will also be able to make better decisions regarding which work item to focus on.

This setup is meant to be a "living organism" that is automatically updated based on your actual progress. So, don't expect to always see the same forecast, as it's unlikely.

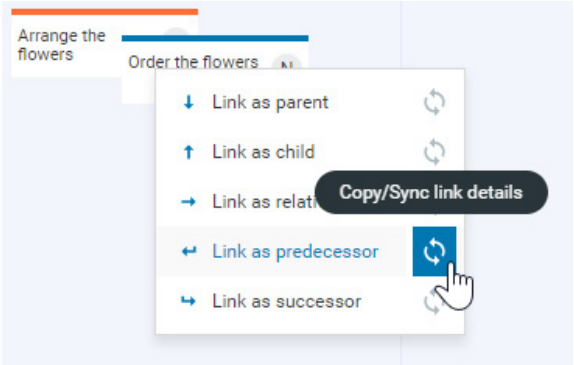
The forecast will change with every card moved from column to column and that's why we call this module "Continuous Forecasting".

The beauty of this approach is that it shows you what is real TODAY. It doesn't show you what you planned two months ago, it cannot see the future and guarantee that things will turn out to be like that but it will show you what is really going on. This is the best kind of help you could ever get from a software tool.

Dependencies

Once all the work items are available, the teams should be ready to start the execution. However, sometimes certain tasks should be completed before others. In our wedding example, the flowers can be arranged only after the flowers have been ordered, so the task to order them should be completed first.

Kanban, and Kanbanize, in particular, borrows the concept of predecessors/successors from the traditional project management tools. This means that one card can be linked as a predecessor of another. To do so in Kanbanize, just drag one of the cards over the other with Ctrl key pressed:



When you do that, the successor card will not be “pullable” unless the predecessor has been completed. This means that if anyone tries to move the successor card to In Progress and start working on it, the system will show a warning and will automatically return the card to its position.

Equipped with the project scope, the overall plan, and the breakdown structure, you’re ready to start the real execution. Making the project happen using Kanban is the most exciting part of this book, so prepare yourself to have some fun.

CHAPTER FOUR

Project Execution

Let's start this chapter with a quick overview discussing what roles exist in Kanban and whether you should have them or not.

Roles

This is a less-known fact, but actually, there are two formal roles that can be implemented by teams practicing Kanban:

- Service Delivery Manager (SDM)
- Service Request Manager (SRM)

The Service Delivery Manager is a role dedicated to improving the efficiency of your workflow and is also known as "Flow manager".

The SDM's functions are related mostly to the 7 Kanban Cadences, which briefly explained are different types of flow management meetings aiming to drive evolutionary change and "fit for purpose" service delivery. In other words, the Service Delivery Manager makes sure that work items flow across the Kanban board and facilitates continuous improvement activities.

The Service Request Manager is another under-the-radar Kanban role. It is a role suitable for "middlemen" who have deep knowledge of the team or company's value stream but don't necessarily create direct value to the customer.

The primary goal of an SRM should be to serve as a risk manager and facilitator. To describe it with actual responsibilities, it is associated with:

- Ordering work items from the backlog and facilitating prioritization of what comes next.
- Owning the policies for the system which frame decisions together
- Improving corporate governance, consistency of process, and reducing personnel risk associated with a single individuall

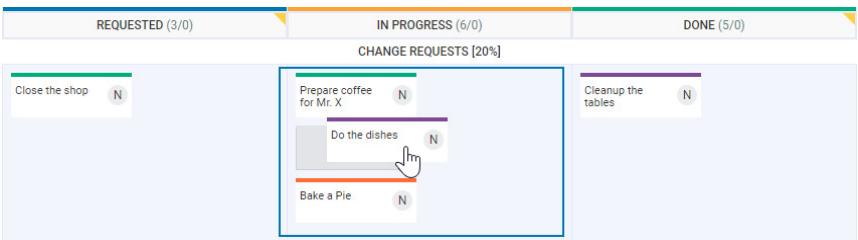
In the context of Project Management, where the assumption is that a Project Manager exists, this role is likely to be taken by that same person. However, the implementation of the Kanban roles is not mandatory and should only be considered in case it adds value in the specific context. Do not implement the roles just because some smart people told you, this is likely

going to be a mistake.

Prioritization

This is one of the areas where Kanban really shines compared to other management methods because prioritization is really, really simple and explicit.

The cards that are at the top of the column are the most important ones. The cards at the bottom are the least important. In case you want to change the priority of a card, just put it in the right spot. That's it.



If you thought that was too easy and it wouldn't work for your complex scenario, there's more. But let's first introduce two other concepts: Classes of Service and Capacity Allocation.

Classes of Service

Classes of service are sets of rules and policies that describe how work should be treated. Each class of service is agreed upon by the teams and the stakeholders and is a formal way to

prioritize and sequence work.

It's important to note that you can have a given work type that has different classes of service. For example, a meeting with the customer has a higher priority than a meeting with the HR manager, requires formal dress code and must be attended by at least three employees.

These are the four main classes of service you can start with:

Expedite – expedite items require immediate attention. They can violate the WIP limits and by definition should be the most important work for everybody. Be careful, though, if you are constantly expediting work, this is a signal that something is wrong. Expedite work should not be the norm.

E.g.: an emergency landing of a plane that's run out of fuel.

Fixed date – Work that must be delivered by a specific date is assigned the fixed date service class. These are tasks or deliverables with a strict deadline.

E.g.: all passengers should board the plane by 5:45pm.

Priority – Sometimes certain work items are just more important than others. If that's the case, you can use the "Priority" service class.

E.g.: priority boarding for the business-class passengers.

Standard – The majority of the work should be assigned the "Standard" service class. It represents the typical scenario where we can't realize any economic value before we deliver the work item, but there is no concrete deadline to manage.

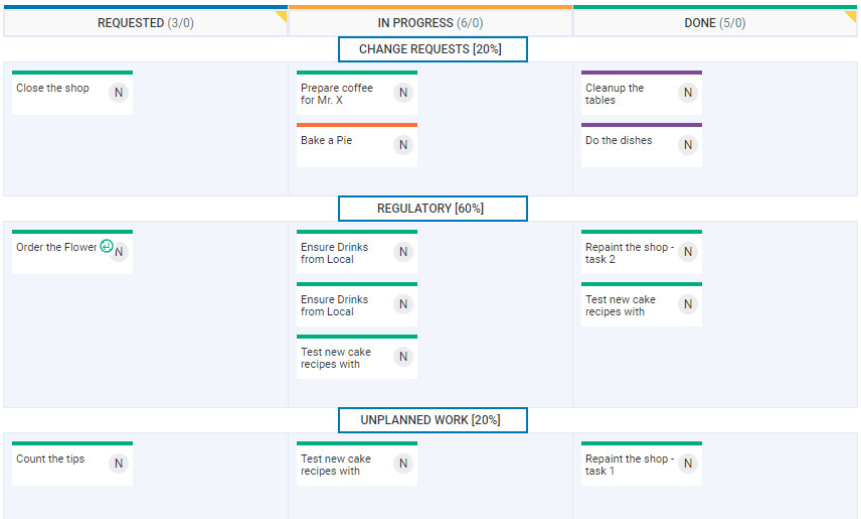
E.g.: luggage should be collected by the passengers upon arrival.

Capacity Allocation

Having several work types and classes of service competing for people's time and resources is the norm for most projects. When that's the case, how do you decide what to work on? Should it be this change requests, the regulatory requirement or that complex thing that requires a specialist from another team?

The distribution of your capacity across the different work types and/or classes of service is called capacity allocation. For example, you can allocate 20% of the available capacity to change requests, 60% to regulatory requirements and 20% to unplanned, expedite work.

Kanban has a very neat way of dealing with Capacity Allocation. It's a very simple one - splitting the board horizontally into what we call "swimlanes".



So, how do the swimlanes help with prioritization? Again, it's simple – the work items in the top lane are the most important ones. Then you go to the second, then to the third and so on. Of course, this goes hand in hand with the capacity allocation, because in case the top lane contains many work items, you may never get to the second.

Sizing and Estimation

There are many Kanban myths and one of them is regarding estimation. They say that there's no estimation in Kanban. While this might be true and even the desired way to manage work with Kanban, there's no rule that says you shouldn't estimate.

Estimations are very close to the human mind and even if we wanted to, we can't ignore them. What is the first thing you want to know when you get your car to the service? When is it going to get done? Right?

Just ignoring estimations is a bad idea. On the contrary – we should pay a lot of attention. However, the way people usually do estimation is inefficient and sometimes even flawed.

Instead of trying to guess how many hours or days exactly a given task will take, take an alternative approach. Identify your service level agreement (SLA) for each of the classes of service and work backward.

Let's say you have a project that has to be estimated. Ideally, you wouldn't have to do it, but we live in a world where customers expect fixed cost/deadline. If that's your reality and you don't have historical data to base your estimation on, you need to have a way to do it.

You know from experience that delivering work frequently is a good practice. That's why you decide to deliver something every three days, which effectively means that your SLA can be 3 days.

To estimate the project, start slicing it into tasks that will take three days or less with some probability (say, 85%). Don't get hung on the precision here. Just make sure that the task is

something like three days of work. Sometimes it will take 3 minutes, sometimes 3 weeks, that's expected. However, if you are calibrated, the variability will likely balance itself, making your estimation somewhat correct.

In the end, you will have a number of tasks each totaling three days of work or less. Multiply the number of cards by three – this will give you the approximate duration of the project.

It's important to note that your estimation has some error by definition. The bigger the project the greater the error. That's why it's crucial that you use forecasting methods to make sure you're on track throughout the project. Luckily for you, Kanban has a very powerful offering to help you out. We will talk more about this in the next chapter.

Commitment

The first thing that comes to mind when discussing commitment is that you must defer it to the last responsible moment. There's no point in committing to a huge scope upfront. Something might change during the course of the project and you should have the option to change your mind.

Take a look at the Kanban board below. To demonstrate how you should think about each of the sections of the board, we've

replaced the standard Backlog / To Do / Doing / Done with the stages of a relationship - Dating / Engaged / Married / Children.

Commitment in Kanban is like progressing through the stages of a relationship. When a card is in the Backlog, you don't need to do anything special - you're just dating. Once you take the card out of the Backlog, you commit to working on it in the future - you're getting engaged. Once you start working on something - you're fully committed to it and there's no turning back - you're married. Finally, when you have children, there's no undo. This is how you should think about getting things Done.

The colored arrows represent the possible transitions on the Kanban board. These are the so-called commitment points. That's so because you are supposed to cross them only from left to right (the green arrows) and never backward.

The orange arrow between Backlog and Requested is there to say "Think before you pull something out of the backlog". Nothing more than that. In many situations, it's okay to put something back in the Backlog, even if you've been "engaged" to it, but if you could avoid it, it'd be better.

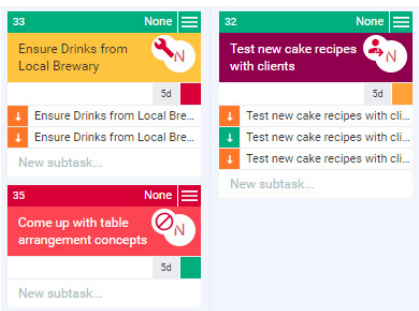
There's no arrow pointing from Married to Engaged or from Children to Married because you're not supposed to go back. Even if you could technically do it, do your best not to. It will be way worth the effort, but we will talk more about this in the last chapter of the book when the topic is "process stability".

Blockers & Escalations

As much as we dislike it, issues are part of our daily work. Sometimes we'll have a serious problem that hinders the progress – that's what a blocker is in Kanban. If a card cannot proceed to the right, it's blocked.

Visualizing blockers on your boards is crucial, as that's how you notify everyone about the issue. Besides, that's how you ask for help. If you have a healthy environment that promotes collaboration, blocking a card should lead to people swarming on the blocker, trying to resolve it as quickly as possible.

This is how blocked cards look in Kanbanize:

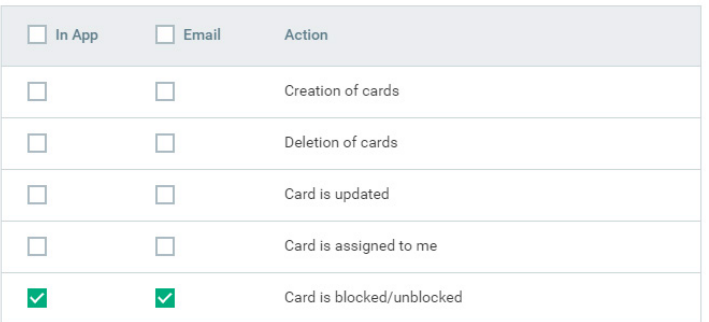


Note that each blocked card has a different color and icon. That's how you visually communicate the reason why the card is blocked, as sometimes you might have more than one blocker reasons.

As a project manager, you might want to know when blockers occur. The traditional way to do it would be via email or via a meeting. If people are expected to notify you about blockers, they might sometimes forget. If you need to wait for the meeting, even if it's a daily one, then communication is hindered.

If you use Kanbanize, there are several ways to inform yourself about blockers and manage escalations accordingly.

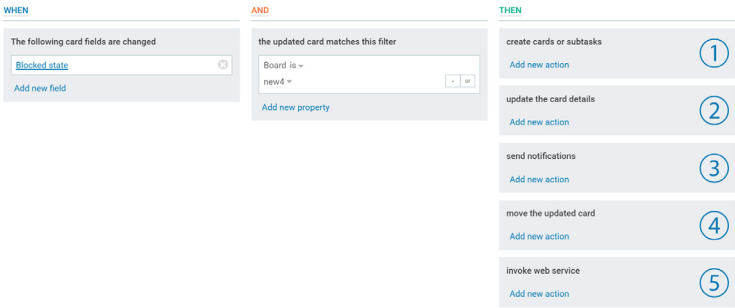
Standard notifications:



In App	Email	Action
☐	☐	Creation of cards
☐	☐	Deletion of cards
☐	☐	Card is updated
☐	☐	Card is assigned to me
☑	☑	Card is blocked/unblocked

You can subscribe yourself for notifications about blocked/unblocked cards for each separate board in the account. This means you can receive notifications for blockers that occur in one board but ignore the ones that happen in another.

Custom action via a business rule:



Using a business rule in the system, you can do any of the following in case a card gets blocked/unblocked

1. Create another card in the system or a subtask for the blocked card. This can help you standardize the blocker handling process.

2. You can change specific details of the blocked card. E.g. you can add a tag “blocked” that will, later on, help you analyze how many of the cards have been blocked historically.

3. You can send a custom notification to users in the system or external email addresses. The text of the notification can be customized using fields from the card itself.

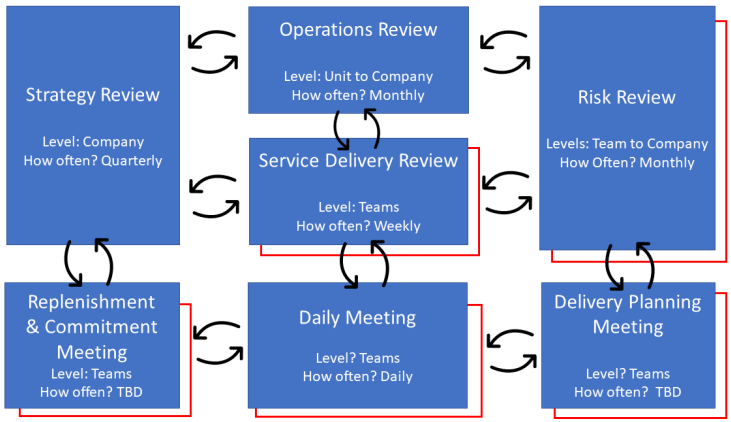
4. In case the card has been unblocked, you can move it to a specified position on the board.

5. You can invoke an external web service and notify it for the change. This is useful for integration purposes.

Meetings & Feedback Loops

One of the Kanban practices states that you should implement feedback loops. The feedback loops are there to ensure smooth information flow and set the base for continuous improvement.

There are seven meetings in Kanban and ten feedback loops (don't get scared):



This absolutely doesn't mean you should have seven new meetings. You should not have those meetings just because Kanban says so! The idea here is to check where feedback loops are necessary and make sure they exist.

Most probably, you already have these meetings in place in one form or another. If you do, use them instead of creating new ones. If you don't, ask yourself the question whether you

should have them and why. If no good reason comes up, keep doing what you're doing.

One meeting that we dare to classify as mandatory is the Daily Meeting. It's for the teams and it's there to ensure flow on the team boards. This meeting is best held in front of the Kanban board in less than 15 minutes. The team members go through all the cards that are in progress and discuss if anything is hindering flow. If you don't have this meeting, make sure you implement it.

Regarding all other meetings, feel free to experiment, using the image above as a reference and not a recipe.

CHAPTER FIVE

Project Tracking and Forecasting

We're entering the last part of the book. With the knowledge you've accumulated so far you're ready to go and start your first Kanban project. However, your approach wouldn't be complete without a way to measure progress and an answer to one of the most important questions – "Is my project on time?".

This chapter is entirely focused on metrics and the core flow analytics you'll need to make sure everything is on track. However, we will only scratch the surface of the topic. Taming flow through analytics is an advanced topic and this book can only show you what's possible. If you really want to dig deep, you should sign up for a trial with Kanbanize and see how

Flow Analytics work in practice.

The First Principle of Lean - Value

Yeah, we said we'll talk about metrics, but one thing precedes all charts, data, graphs, statistics, equations and all that – value. The principle of value sits at the core of the Lean philosophy and we, as managers, need to constantly remind ourselves of what Peter Drucker said:

"Nothing is less productive than to make more efficient what should not be done at all."

Before you gather all those metrics and show how productive your teams are, ask yourself what value you're producing. For whom? How do you measure if value is really being generated? How do you know your happy customers are happy?

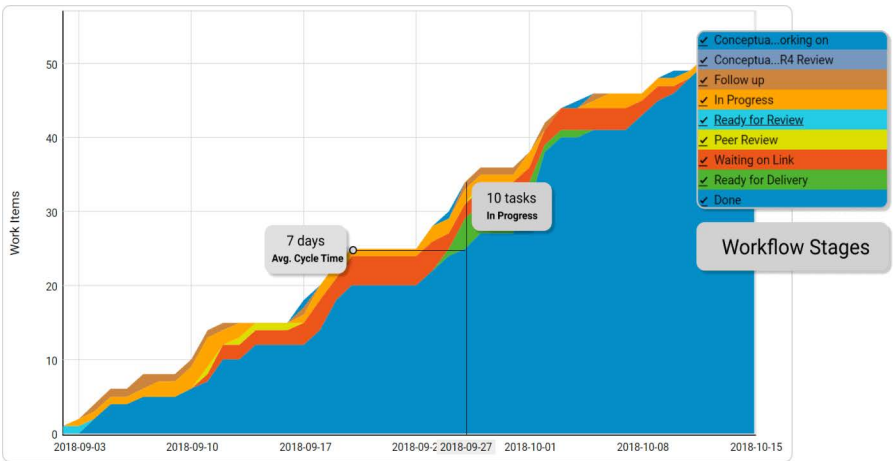
Actually, happy is not enough. Your customers have to be ecstatic about your solution. If they aren't, metrics are irrelevant. Just bear that in mind throughout the rest of this chapter and make sure you use data to get your customers to crave what you do.

Alright, let's dig into the charts...

Cumulative Flow Diagram

The cumulative flow diagram (also known as CFD) is one of the most common charts in Kanban. It provides a concise visualization of the three most important metrics of your flow:

- Cycle time (the horizontal distance between the lanes)
- Work in progress (the vertical space between the lanes)
- Throughput (the slope of the chart)



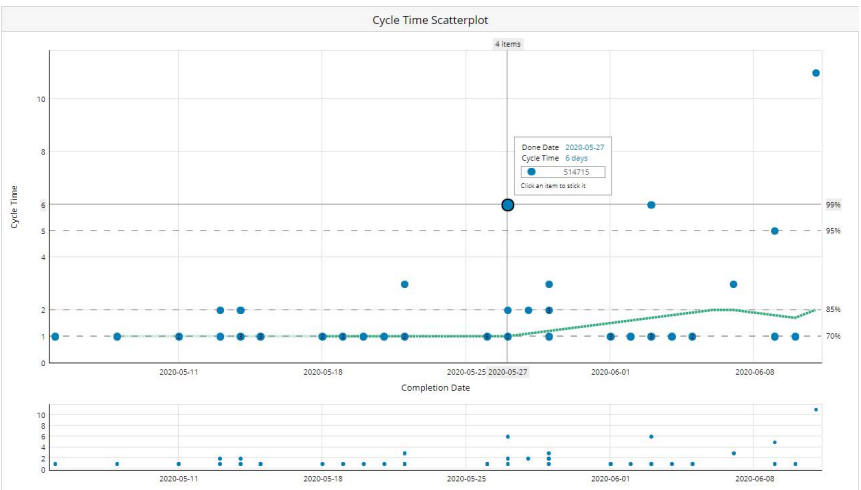
Its main purpose is to show you how many cards have been completed (y-axis) over time (x-axis). The top-most line of the chart visualizes the arrivals (new cards entering your workflow) and the bottom-most line visualizes the departures (done cards). Because it is a cumulative chart, it always goes up to the right.

You can use the CDF in your project to make sure that:

- WIP doesn't grow over time, as that would make you slower. More to that in the last chapter.
- Your throughput is stable and does not decline. If the slope of the chart is stable or becoming steeper, things should be fine.
- Your cycle time is stable (this will be the case if your WIP is not growing).

Cycle Time Scatterplot + Cycle Time Trend

The cycle time scatter plot is an adapted version of the well-known chart. It is revered in the Lean management world because it provides a very detailed picture of one of the key metrics in Lean – cycle time.



The goal here is to visualize the cycle time of all cards within a given time frame. Each dot on the chart represents a card on your Kanban board. The positions of the dots are determined by the completion date (x-axis) and the cycle time (y-axis) of the corresponding card.

You can use the scatter plot in your project to:

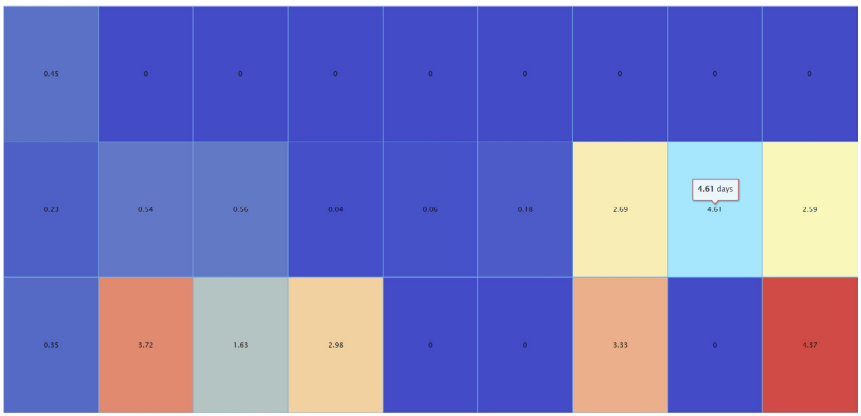
- See how the shape of the cycle time is changing. If the dots on the chart form a triangle (as shown in the picture above) this means your cycle time is growing.
- Investigate individual cards that took too long to finish. It can be a source of information for continuous improvement meetings.
- You can forecast the cycle time of individual work items. Looking at the image above we can tell that 85% of the cards took 11 days or less (see the dotted lines).

On the Cycle time scatterplot you can also see the cycle time trend line (the green line on the diagram). As the name suggests, the cycle time trend shows how your cycle time is changing with time. It is one of the most useful metrics to help you perform historical analysis of your data.

If the green line increase to the right, it means you're becoming slower with time. This is a negative trend that has to be paid attention to and reversed, if possible. On the contrary, if the line goes down, then you're improving.

Cycle Time Heat Map

The cycle time heat map visualizes the bottleneck in your process. The red areas on the map show where tasks spend the most time on average or cumulatively. Therefore, the red areas are the most logical places for you to improve.

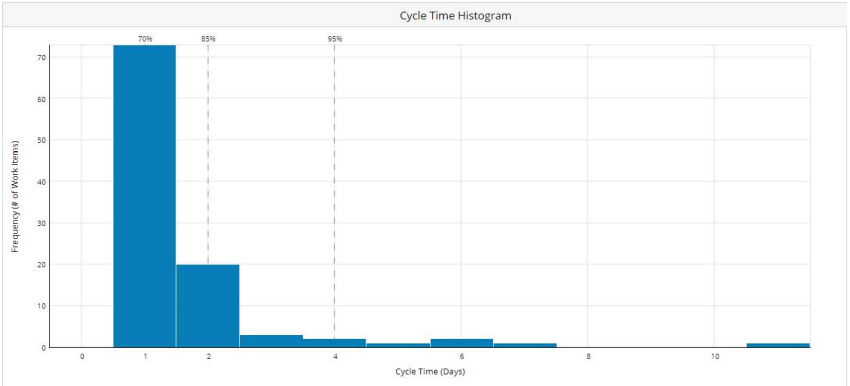


Cycle Time Histogram

The cycle time histogram remains among the most widely-adopted charts in Lean management.

If you have been practicing project management for a while, you probably know that metrics vary and there are plenty of ways and tools to measure them.

The cycle time histogram shows the distribution of cards based on their cycle time.



The horizontal axis of a typical cycle time histogram visualizes the different cycle time values of all the tasks that were on your Kanban board during a predefined time frame.

The vertical axis shows the frequency of the cycle time that work items on your board share. Each column on the chart serves as a bin containing the number of tasks that had the same cycle time. Their height depends on the number of tasks that fall into it.

Flow Efficiency

If you are new to flow metrics, measuring flow efficiency might seem like a difficult task. In reality, it is done with the help of this simple formula:

Flow E iciency[%]=Value-added Time/Lead Time*100

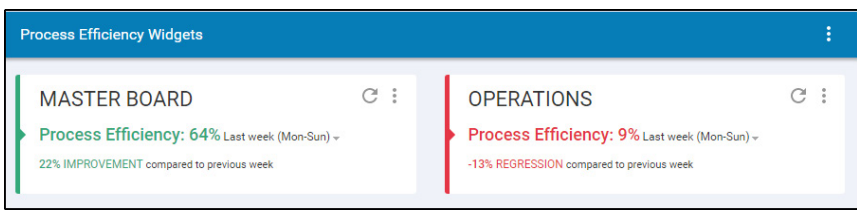
Explaining it in a practical manner, if a task needs 1 day to be completed, but it's been actively worked on only 2 hours per day, it will be finished in 5 days (assuming a working day of 8 hours).

Then, the formula would be:

Flow Efficiency[%] = 1d/5d*100 = 20%

You may wonder what value-added time is. It's the time we actively spend working on something. You get it when you subtract the wait time and the block time from the total time.

Now you may realize why we have a special marker for the queues columns in Kanbanize. Right, because we track these times and then help you calculate the flow efficiency automatically. We achieve this using a special type of Widget that you can pin to your dashboard:



The widget will compare your current process efficiency with a previous period and will color-code the outcome. If you've been improving, the widget will be green. If not, the color will be red.

It's a lot of information, but the thing you should pay attention to is the big number at the top. If it's greater than 50% you're doing exceptionally well. If it's greater than 20% you are okay. Below 20% is not something to be proud of but it's not terrible. If you're getting 10% or less, don't panic, you're soon to be a hero in your organization!

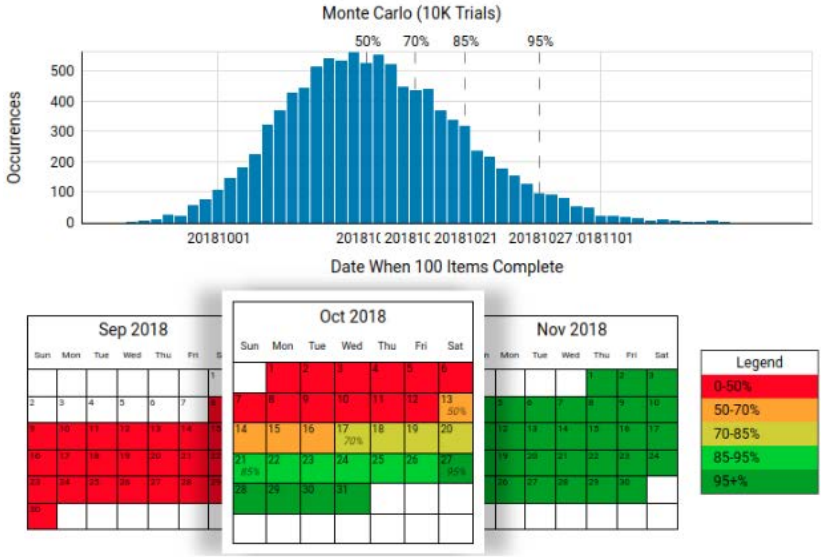
Monte Carlo Forecasting

"When is it going to be done?" – this is one of the most important questions that project managers hear all the time. As important as it may be, we rarely have the answer.

The good news is that Kanban solves this problem for you. If you keep several simple rules you will be able to forecast with big confidence using what we call Monte Carlo simulations.

These simulations are based on your historical data (historical throughput) and surprisingly enough, you don't need loads of it to get started.

Of course, the more data you have the more accurate the forecasting will be, but you can start once you have 20-30 completed tasks.



If you recall the beginning of this book, we promised that we are not going to promote deterministic ideas. This holds true for the Monte Carlo simulations too, which means that whenever we pick a date, we should attach a probability to it.

That's why the dates in the calendar above are color-coded. The red dates show a very low probability of success (0-50%). The green dates show a probability of 85% and above.

For example, we shouldn't claim that our project will be finished by the 26th of October. We should say instead: "There is an 85% probability that we complete this project by the 26th of October". It's a different statement.

Based on the input parameters the simulation can answer two different questions:

1. How many tasks can be finished by the deadline X?
2. When are the next X tasks going to be finished?

The mechanics of the forecasting process are embarrassingly easy.

1. You open your Kanban board and you check how many work items you have in the backlog.
2. You go to your analytics engine and enter this number in the Monte Carlo simulation.
3. Two seconds later, the simulation reveals the projected date with a given probability (85% or 95% is preferred).

It's hard to believe that this can work for real, but it does. As a matter of fact, Monte Carlo is the only popular scientific way to approach estimating a project and forecasting when it is going to get done. It works much better than any human judgment because it's not biased, relies on actual data and it's so fast, that you can do it every day.

So, even if the model is a bit off in the beginning, with more accumulated data, it will become more and more accurate every day. This is not something that holds true for human judgment, especially if we move from one person to another.

It wasn't that hard, was it? As we promised, analytics in Kanban are very powerful, but not too complex to start with. One thing is certain - if you've gotten that far, you are really close to making your next project a wild success. Only one thing is missing - how do we manage risks?

CHAPTER SIX

Managing Risk

If you have tried out the concepts outlined in this book in practice, you've come a long way. You've created a visual management system based on Lean and Kanban principles such as visualization, limiting work in progress and managing flow.

You've created a Kanban project board and connected it to one or more team Kanban boards. You've learned about the major charts in Kanban and you're ready to analyze the health of your flow. Believe it or not, this is a pretty unique accomplishment, as few people know how to do these things.

There is just one question that hasn't been answered yet.

“What if things go wrong?”

Preventing things from going wrong is the subject of risk management, which is an important project management discipline.

One of my recent tweets perfectly fits this chapter of the book, so here’s a shameless plug:

The first part of the tweet is “The best risk management strategy that I’m aware of is optimizing flow and making your teams/services predictable”. This goes in a very different direction than the traditional risk management where you fill in a spreadsheet, list the risks, put a probability, add some weights and never look at it again.

Even if you inspect the spreadsheet regularly, besides thinking about the potential risks (which is a good thing), nothing really changes. From my experience, risk management is an artificial procedure that ticks a checkbox. If that’s not the case with you, feel free to ignore my ranting.

You must have heard about Amazon’s cloud (EC2). You must have heard about Netflix as well. Netflix runs on Amazon’s

EC2. Judging by the annual revenue of Netflix (11.6B), having the system offline or partially offline could lead to insane losses. A simple calculation shows that Netflix makes more than a million per hour. If you're making that much in sixty minutes, the last thing you want is your servers to crash.

Yet, Netflix came up with this: Chaos Monkey. Their own description of this program is "a resiliency tool that helps applications tolerate random instance failures."

In reality, this tool randomly kills some servers in your production environment. I'll repeat: **this tool randomly kills production servers.**

Please, pause for a second and ponder over this. The company that makes more than a million per hour invested time and money to create a tool that randomly kills production servers. This must be the definition of stupid but it's in fact genius.

Acknowledging the fact that failures will occur (EC2 is not perfect), the engineers at Netflix took an alternative approach. Their thinking was that when failures occurred, they had to be able to handle them gracefully and then recover without tragic consequences.

Knowing that at any moment in time a server can simply disappear forces the developers to design systems that are fault-tolerant and resilient. When your base point is that

failures will happen, you think about the world differently.

This strategy might sound crazy, but it's everywhere around us. Nobody likes crashing brand new vehicles into a wall but the automobile engineers know that crashes happen every day. The construction engineers in Japan know that earthquakes will come, that's why they use giant pendulums inside the buildings to counterbalance the force. As we said, when we start from the point that things will go wrong, we act differently.

Translating this to the language of project and risk management, we need to assume that our project will get late, that our people will get ill or quit, that our estimates are wrong, that cost forecasts are inaccurate, that servers will crash, that the customer will change their mind, that the competition will be stronger, that the market won't be ready for our product and many, many more. If we go back to the Netflix example, we must operate as if Chaos Monkey is always on.

The only reasonable way to face all these challenges comes down to mastering flow. We would go even further and talk about getting Lean in general, but it would take a whole new book. That's why we would like to focus on three things only:

- Relentless focus on value
- Pursuing process stability and predictability
- Continuous improvement

Eliminate Waste

It's a common misconception that the main goal of Lean is to eliminate waste. This is actually not true. Lean is all about delivering as much value as possible with the least amount of investment possible. Eliminating waste is a consequence.

However, some forms of waste are easy to spot and get rid of. That's why we will talk about waste in a bit more detail. Note that we won't be covering the 7 wastes of Lean but something else that's more applicable in the knowledge work context that we are in.

Allen Ward and Durward Sobek talk about the wastes in product development, which is closer to knowledge work than manufacturing. In their book "Lean Product and Process Development", they explain the three wastes in product development - scatter, handoffs and wishful thinking.

Waste #1: Scatter

"Scatter can be defined as the actions or inactions that make knowledge and information ineffective by disrupting its flow."

With no intention of perpetuating the cliche that it's always the management's fault, it has to be noted that most of the Scatter is caused by bad management decisions or activities.

Below are some of the common causes for scatter.

Reorganization

When things aren't working well, a typical response by the senior management is to reorganize the department or even the entire company. Teams will be broken and new groups will be formed. This creates a lot of information blockages, which in turn affect the entire business in a very negative way. Reorganizations within the lifetime of a project might be devastating to the end result, so only do this after carefully examining the consequences.

Expediting Work

Another typical example is expediting work. When you are forced to do something urgently, or when you force others to do it, scatter is inevitably created. That happens because the normal flow of work is disrupted and people have to switch contexts involuntarily. Switching context to address an expedite request, which by definition increases the work in progress, is inevitable at times. However, if it happens all the time, there's something wrong with the environment, that has to be addressed.

Adding More People to a Project

You've all heard the joke that 9 mothers can't give birth to

a baby in one month. It's certainly not a joke in knowledge work, where adding more people to a late project actually makes it later. Simply put, this is a very common source of scatter.

Additional Checks & Procedures

Imposing additional checks and procedures to address quality issues, for example, might be a really bad idea. If the root cause of the issue is somewhere else, adding a checklist will likely improve the quality but will continue to waste time for the organization.

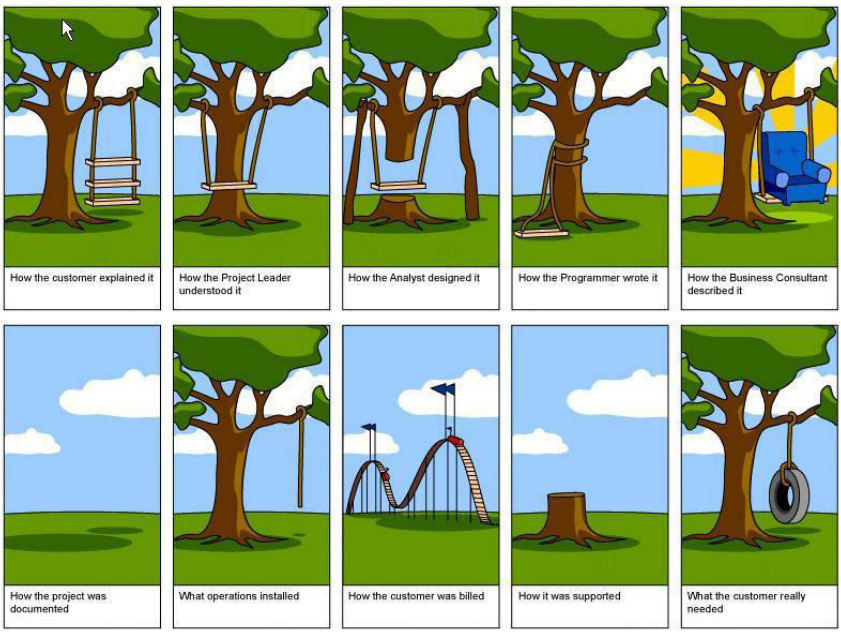
Bad Tools

Many companies attempt to improve their delivery by adopting lean and agile methods. However, they rarely invest in the right tooling, which is a necessary step for each reasonable agile adoption. When the teams are expected to deliver more in a much more dynamic environment, tooling turns into a problem and not addressing it, creates scatter waste.

Waste #2: Hand-offs

A hand-off is any transfer of knowledge, responsibility, action or feedback. A typical example of a hand-off is a designer passing the design to the UI developers. Marketing handing off a piece of content for a web page to be coded.

You've seen this image which is supposed to ridicule the waterfall approach to knowledge work. However, what this image demonstrates is what hand-offs do to knowledge work - results get worse with each of them.



Hand-offs are a terrible form of waste, as they cause information and context decay. The customer handed off their requirements to a project leader. Then the project leader handed off this information to the analyst. The analyst handed off the information to the programmer and so on...

To tackle the waste caused by hand-off you should try to minimize the number of different parties involved in a project.

If you've heard about Jeff Bezos's "two-pizza team", then

you'd know how to form teams that won't suffer from hand-offs badly. Having a team owning the requirements, deployment and, if possible, financials is the best possible environment for a successful project.

Waste #3: Wishful Thinking

This is the worst form of waste because it's usually invisible. Wishful thinking means making decisions based on your gut feelings or opinions, without considering data. This happens very often in organizations that develop products or services based on a predefined master plan, that is being followed blindly.

The damage caused by wishful thinking is usually huge because it accumulates over time. If a project was not to be started at all, but we still decided to invest in it, it will take months or even years to realize we shouldn't have done it. By that time, we would have sunk significant cost, which wouldn't let us kill the project. This makes things even worse because we keep investing hoping that the project will be a success, which rarely turns out to be the case.

To tackle wishful thinking, always use data. If no data is available, experiment, learn and then use what you've learned to make decisions. Use experimentation extensively throughout the lifecycle of the project in order to reduce uncertainty, when it exists. This is the only way to protect yourself from this

terrible waste.

Stability & Predictability

To avoid risks turning into reality, you need to be informed what's going to happen in the future. In other words, your process needs to be predictable. However, predictability is a difficult game to play.

The good news is that stability and predictability go hand-in-hand with something that you already know - metrics.

In order to have your process stable and predictable, you need to follow these rules:

- What goes into the system should go out (work mustn't disappear).
- Your WIP age should be neither increasing nor decreasing

Work mustn't disappear

This rule simply says that if you start working on something, you must finish it. That's why deleting cards from your Kanban board isn't a good idea. If it happens often, you'll be skewing your metrics and you won't have reliable data to base your forecasting on.

Do you recall the Kanban board describing the relationship

statuses (engaged, married, etc.? It is the same concept. Just flow your cards to the right until completion and you'll be fine with this part of the game.

WIP aging should be constant

The WIP aging chart, that we talked about earlier, shows whether some work items are taking too much time compared to your historical data. If that's the case, then your data won't be stable and your system won't be in predictable.

This doesn't mean you shouldn't improve your cycle times! If you are optimizing your cycle time, the WIP age will also go down. This is good! However, if that's the case, make sure you're not using this data for forecasting. You better wait for the cycle time to stabilize or just be conservative and assume a bigger error margin.

Continuous Improvement (Kaizen)

It occurred to me rather late that continuous improvement is a risk management strategy. I used to look at it as something you do to be better and to achieve more but the more I learned about Kanban and Lean, the more I was becoming aware of its risk management potential.

If you are improving your system, you are by definition eliminating potential sources of failure. In doing so, you proactively reduce the probability of a risk materializing.

This approach significantly differs from the reactive way of awaiting the risk to become something real, after which we follow the procedures defined in the risk management plan.

A very good example of this it the air transportation. When flying became affordable, many people preferred it to driving or sailing for days. Naturally, this led to the explosive growth of commercial flights.

Unfortunately, the technology was still imperfect and there were a lot of fatalities, especially in the 70s and 80s. This led to serious investigations and analysis of the root causes of the crashes. As a result, many improvements were made and the air travel is now considered one of the safest ways to move from point A to point B.

This is how we should be thinking about our projects. By investigating all the root-causes and doing the necessary not to repeat the same mistakes again, we are giving ourselves a chance to avoid many of the issues we would have had to face otherwise.

Conclusion

Now that you've read about Kanban Project Management it's time for you to actually start doing it. Only then will you discover the true benefits of visualizing work, limiting work in progress and managing flow on both the project and team levels.

But please, be warned! Do not think of a Project Manager's Guide to Kanban as a framework. We encourage you to start where you are and gradually find out what works best in your own context.

Take the ideas presented in the book, experiment and learn from your mistakes. This is the only recipe we are ready to give.

There is no better time than now to start with Kanban Project management!

Try Kanbanize for free