Kanban University

KANBAN AT NET-A-PORTER

Regaining the Control of Flow

KANBANCASESTUDYSERIES

They did not meet the commitment, just as they had not two weeks ago! Johnathan Swan was annoyed. Together with his teammates, he had been trying to deliver the promised number of work items in the two-week working timeframe known as an iteration. And just as they thought they had finally achieved that, something was stuck in testing. Johnathan disliked the pressure of the fixed iteration, but disliked that he failed on his promise even more. In that summer of 2012, the team shared the frustration.

Now, in late 2013, Johnathan Swan is busy with an improved stream of work. Johnathan does not have to think about whether he will fit the work for that stream—or anything else for that matter— into an iteration any longer. The team is working to improve their code and the logic of how systems respond, which makes a big difference in the amount of time stakeholders spend doing their jobs.

So far, Johnathan's team has saved dozens of hours for their colleagues by making the systems respond faster. With the galloping deadlines of iterations and release plans gone, Johnathan and his colleagues now feel that they are in control, and they actually are more efficient.

Working in iterations was derived from a concept that if work were time-boxed it would make the team better and more efficient. For Johnathan and his team, that came at a very high price. They took matters in their own hands and incrementally changed the way they work by following the teachings of the Kanban Method.

"It has revolutionized how the team works: enabling us to ship more, quicker, and give the team a higher level of purpose and energy," Paul Brennan, the product owner and Group Merchandising director says.

But this is much more a story of evolution than revolution.

Background

"Can you sell fashion online?" Natalie Massenet asked herself this question at the turn of the 21st century when she opened the e-commerce entity NET-A-PORTER.COM.

Thirteen years later, NET-A-PORTER still sells high-end fashion exclusively online through its three sub brands: NET-A- PORTER, MR PORTER, and THE OUTNET.

Everything displayed on the pages of those three websites is selected and curated by the buyers employed by the London-based company. They constantly scout hundreds of leading designers and make purchasing decisions that are afterward displayed on www.net-a-porter.com, www.mrporter.com, and www.theoutnet.com to millions of customers in more than 170 countries.

To make an item purchasable, though, there are many steps that have to be performed. Johnathan's team is responsible for part of the backend systems that are involved. Becoming a technology company that sells fashion has been a slow process, as the IT systems in use were built internally.

"Back in 2006, when I came to the company, the IT team was 20 people," Kam Chovet, now head of Service Delivery, says.

There were no business analysts, project managers, or testers. They were developers who got requests for what to work on via e-mail and released to production whenever they could.

"Those in the business who shouted the loudest got their things done by the developers," Kam continues.

As NET-A-PORTER grew and the technology team increased, the impact of the lack of structure and processes began to show. Kam was among the people who wanted to change that. She wrote specification documents, helped establish teams by vocation, and created delivery cycles.

"In 2009, I attended a conference about Agile practices, where I met an agile coach (Sally Ann Freudenberg). Having had experience [with] Agile in a previous company, I was keen to get it working at NET-A-PORTER and recognized the need to get an expert in to help us implement it here."

By 2010, Agile methodologies like Scrum[1], which aimed to change the way software was produced and delivered, were gaining a lot of popularity. When Kam returned from the conference, she immediately wanted to experiment with this new agile way of working and got in touch with Sally Ann to help with the rollout. The trendiness of Agile, and Scrum in particular, excited the developers.

The project management office was not as excited, though.

"What is this Agile thing? How would we ever start a project without having a full-on specification, and why would the business be involved from the start; this is our role," they would say to Kam.

Educating everyone about the changes and how they would affect them took substantial time and the presence of many experienced external coaches.

By mid-2010, all IT teams switched to the ceremonies[2] and roles[3] that Scrum prescribed. The work cycle was split into two weekly iterations. In the beginning of the iteration they were assigned a certain amount of work by a designated product owner, which they had to deliver by the end of the Sprint. By the end of 2010, official release plans and dates were introduced—the IT teams would not only work in fixed time periods, but they would also release in fixed, three-week time slots.

In this new manner of working, the dependencies between teams became apparent. It was difficult to coordinate smooth delivery in fixed iterations between separate teams regardless of how well they collaborated.

In 2011, NET-A-PORTER went through a major organizational change to address that. Instead of having teams split by type of work, such as development or testing, people were reorganized into squads with designations such as front end, backend, and application development. The idea was that those squads would include people with various roles so that they could independently deliver a requirement.

One of the newly established squads that autumn was the Product Management Systems team, whose responsibilities included part of the databases and backend solutions used by the main web sites on a daily basis. Johnathan joined this squad in January 2012. The team's work varied from placing and managing purchase orders and pricing tools to integration of product data.

As the IT department continued to grow, and the squads were increasing in size, the need for coaching and mentorship to help agility and collaboration became more evident. David Lowe was one of several internal agile coaches on the permanent staff. He was assigned two teams, one of which was Johnathan's.

[1] An agile software development methodology developed in the mid-1990s. Scrum is based on a "Sprint," which is a set period for delivering a working part of the system. Each Sprint starts with a two- to three-hour planning session that includes predefined roles: "the customer" (product owner), "the facilitator" (Scrum master), and the cross-functional team. The customer describes the highest priority in the backlog, and, after the team agrees on how much of it to do and commits to that, the team is left alone to do it in the duration of the Sprint.

[2] Process actions that help keep the content of work on track. In the context of Scrum, those actions are: Sprint Planning, in which the development team forecasts the stories it can complete in the upcoming sprint and the Sprint backlog is created; Daily meeting, in which the team discusses what has been achieved since the last such 15-minute meeting and what the expectations are for achievements until the next; and Sprint review, which acts as the feedback loop to evaluate the work done so far.

[3] Product owner, who has responsibility for deciding what work will be done; Scrum Master, who helps the rest of the Scrum team follow the process; and Development team, whose members do the work of delivering the product increment.

Problem

"When I arrived at THE NET-A-PORTER GROUP, it was the first time I did not have to sell Agile from scratch. It was already in place through the framework of Scrum, and initially I was very happy about that. And then I saw how people struggled with it," David says.

The iterations, a centerpiece of Scrum, had been a big help in getting the technology teams to deliver smaller chunks of work faster and meet the expectations of the business. Compared to a few years back, the overhead of redoing things had subsided and collaborative behavior had sprung up in the teams.

As intended by the methodology's philosophy, the fixed and reoccurring end point that required a certain bit of completed code submitted each time proved helpful as a motivation, and it made the team more efficient.

And yet, the sprints—and their galloping deadlines fortnight after fortnight—were creating a strain on the people from the Product Management team. Completing all the stories they committed to deliver in the sprint seemed to be an unattainable task. They usually came in just short of the amount of story points they had originally thought they could attain.

The logic behind story point measurement in Scrum is that the points are "achieved" only if the story is completed. If a bit of the work for a task—such as some testing (which might, in reality, equate to a single point)—remains, no story points are given to the team for that task.

"They knew they needed just a little more time, but they felt frustrated because in the context of Scrum, they considered they had failed to deliver what they committed to," David says.

The team kept pushing hard to achieve the commitment in the Sprint goal. "Sometimes we took shortcuts just so we could make the iteration [commitment] and that affected the quality of our production. We wanted to have quality as our main purpose, not a deadline," Johnathan says.

Figure 1 - Johnathan's avatar for the Kanban board.

An added problem was the company-wide release taking place every three weeks. Each stream was responsible for packaging all of the tested code and providing it to the merging team. After everything had been uploaded, the teams had to do a final test of their code, but now in the context of everybody else's changes. This practice interrupted the teams on a few occasions during the iterations, which contributed to not meeting the commitment, adding to their frustration.

"I was confident that extending the length of iterations so they matched the cycle of releases would not help the situation," David says.

He felt that a larger time span for an iteration would only result in more commitment from the team, even less focus, and consequently an even worse committed-to-delivered ratio. It would have sacrificed the benefits of having an iteration—the small steps of doing a bit of work and then stopping for a reality check.

"We were not mature or regimented enough to go there," David says.

"I think our particular interpretation of Scrum was wasteful and inflexible, and we were all doing it the same way. We could not adapt it to our needs," Johnathan says.

The frustration had transferred to other aspects of their process. Pressured for time, they doubted the need to have so many meetings (daily stand-up, retrospectives, planning). They missed information about what was coming next.

"The burn-down[4] charts we had were not of much help for us to work out how to improve our situation. If anything, they were hurting the team." David says.

To an extent, that was because burn-down charts measured days and hours, while stories used points.

Eventually, together, the team came to the conclusion that keeping that commitment could become an end in itself and would not help the business sell more fashion. They also realized that not keeping the commitment was not that big of a problem.

So they decided not to worry about breaking the rigid rules in place, and to start seeing the commitment of the iteration as a flexible goal. Whether they achieved it or not would not matter as much. But they still felt they were losing too much time on forecasting the two-week iterations, especially if most of the time they would probably miss them.

"The team knew that they wanted something significant to change," David says.

[4] Burn-down charts show work remaining over time. The measurement used in them could be story points or days or hours.

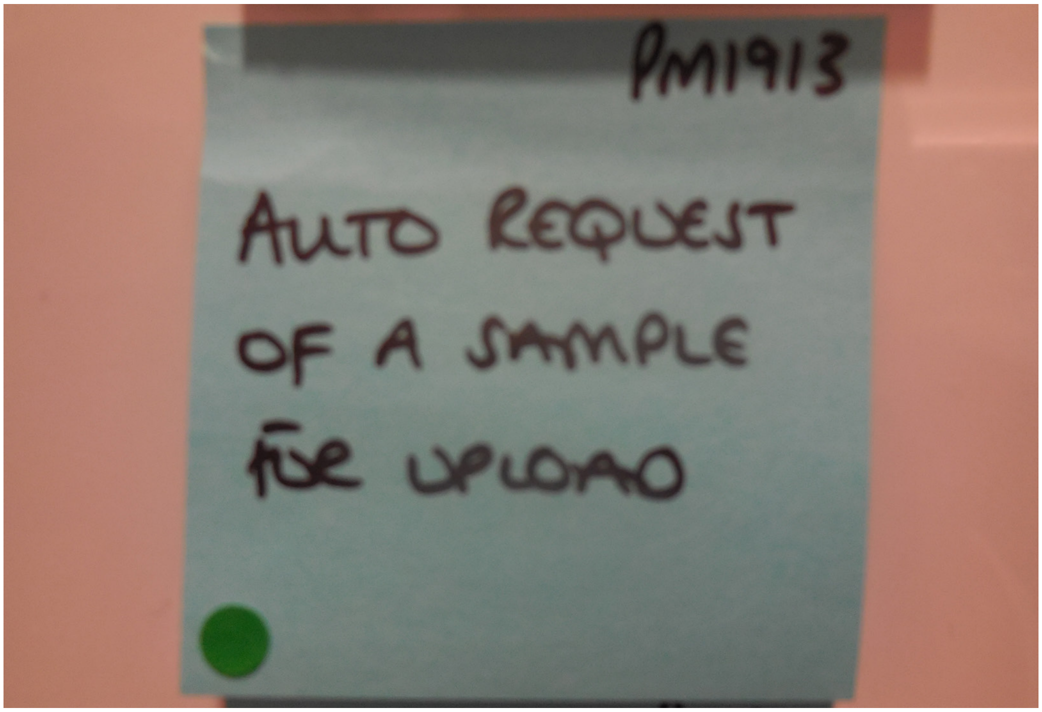


Figure 2 - A sample ticket.

Kanban is Coming to Town

"The first time I heard about Kanban was back in 2006. I knew it smoothed the flow of work, which I attributed to the limitation of multitasking. I had even suggested it back in the spring of 2012, but it did not seem to gather much interest at the time," Johnathan recalls.

As Scrum was not working for the team any longer, and they were breaking the rules anyway, by the winter of the same year Johnathan reintroduced his suggestion to David and the team.

"I thought that Kanban was Scrum with added WIP limits. It is very rare you see a developer getting excited over processes and ways of working, so I offered to read more about Kanban to understand it better," the coach admits.

He was directed to David J. Anderson's book about Kanban by the agile community that he is deeply involved with. David promised to read it and give the team his opinion on the method's appropriateness.

"As I read the Kanban book over Christmas I saw a beauty about the Kanban Method—its lack of instructions. It sounded like the team would be able to apply improvements and ideas in a way that would best suit their situation," David says.

In the meantime, Johnathan decided to read the book, too. On those pages he read further about negotiating priorities and timescales as well as about changes to the way of working inspired and executed by the team itself. He saw it as the salvation from the one-way-to-do-it framework that Scrum had become and from the consequent rebellion against those very rules.

Together, they brought the Kanban Method and its principles and teachings to the team, believing it could improve the team's delivery.

"Nothing would change initially, but with time we would be able to figure out together what works for us as a team," the two explained.

Less forced rigidity and more flexibility—which is within the rules—sounded good. The team was happy to try to move gradually to a way of working that would eventually make sense.

A Journey for Relief

The first thing the team did in January 2013 was to map the process steps for each work item and to visualize that on a whiteboard, referred to as a Kanban board. Their first board had 10 columns, one for each major activity in the workflow. They all thought that this was a mirror of how they currently worked. The only new column that was added was "In Analysis," which served to let everyone know what was coming up, something they had complained was missing before. The work items were written on tickets (Fig.2), and the measurement of story points was kept. Keeping them, as permitted by the first principle of Kanban—start with what you do now—helped David feel safe that, if for some reason Kanban did not work, the team would be able to return to its old ways.

It was also important to keep story points because they felt that the process a size enabled them to ensure a joint understanding of the requirements.

"The process was certainly more important than the resulting point size," David says.

As soon as the Kanban board was built, the next thing David initiated was figuring out what work-in-progress (WIP) limits to put on each column. Limiting the WIP is a guiding principal in Kanban, suggested because of the understanding that focusing on a single task is likely to increase its quality while decreasing the overall time spent on it.

Initially, they put limits that ranged between 2 and 8, which they perceived as low. David was very clear with the team from the start how important it was that the board should reflect reality completely. As tempting as it was, it should never paint an idealistic image of where the team wanted to be; it needed to show where the team actually was. David was afraid that people might be working on more items than appeared on the board.

"In order to make the commitment for the iteration during Scrum, team members used to work on many things at the same time, hoping that doing so would deliver work faster. In fact, I think the time incurred switching between things had the opposite, negative effect," David says. It was a habit that would take time to change.

During the very first stand-up meeting in front of the board, the team looked at all the tickets—from more completed to less completed, or from right to left—across the board.

"All of a sudden, we reached a ticket that nobody seemed to have a clue about. It was something completely forgotten and abandoned," David says.

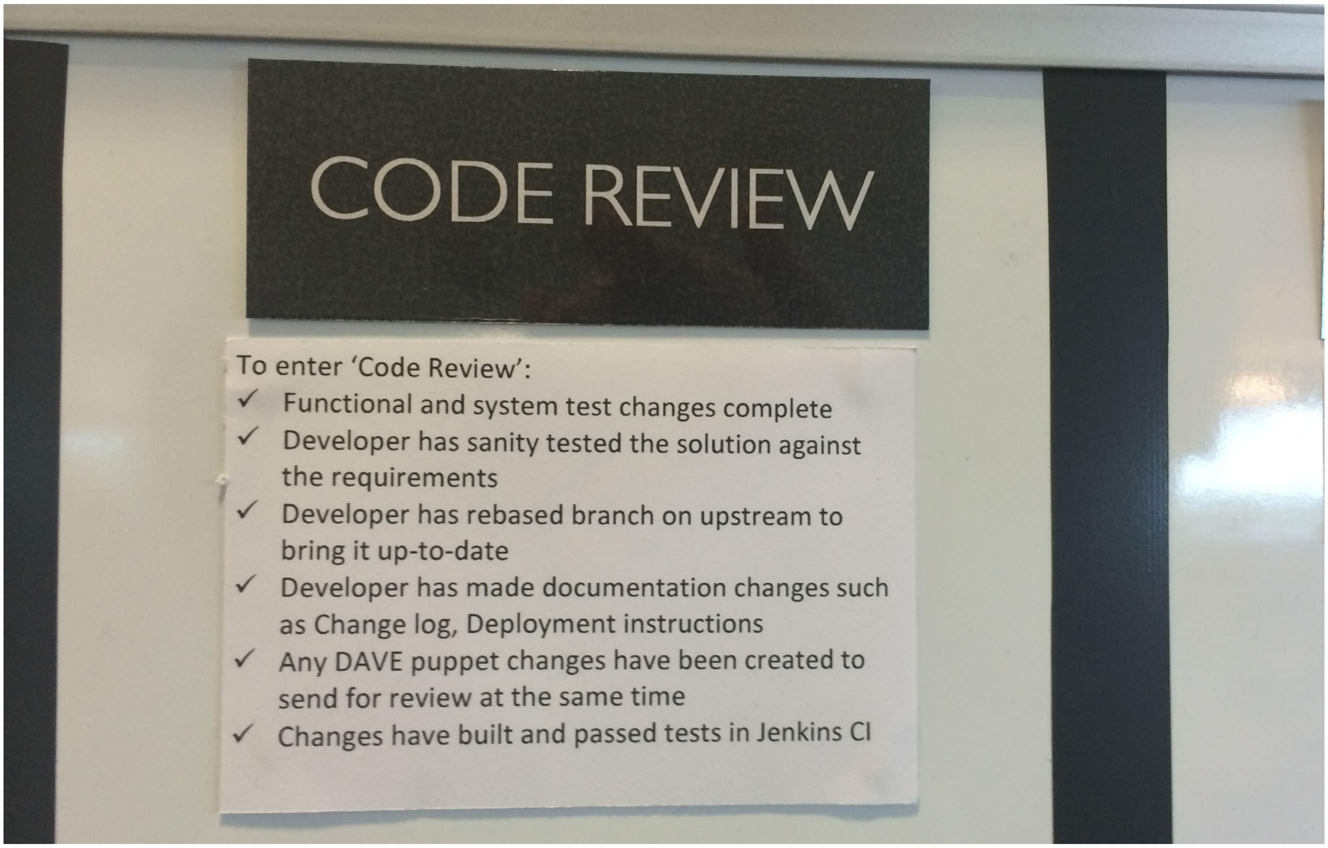
The stand-up meetings used to be about what each person had been working on the previous day and what was planned for the day.

"I think these fifteen minutes a day made everyone feel defensive, having to prove they were working hard, especially in the context of iterations with unfinished work items. When we changed the focus from individuals to the tasks on the tickets, people were... relaxed [enough] to mention problematic issues. Their own performance was not in the spotlight any longer, and we immediately saw the improvement," David says.

"We kept experiencing bottlenecks in the stages just before testing; we kept on shuffling the limits, but we could not avoid being stuck with many tickets. To achieve flow was beyond simply not multitasking," David says.

After many discussions and pondering what was causing the bottlenecks, the team realized that their board had been missing a column from the very beginning - the one that stood for merging the tasks into the common environment where they would be tested. It was an early lesson: They learned just how important it is to allow enough time to critique the initial mapping of the process. With an open mind and a critical eye, the board changed many times.

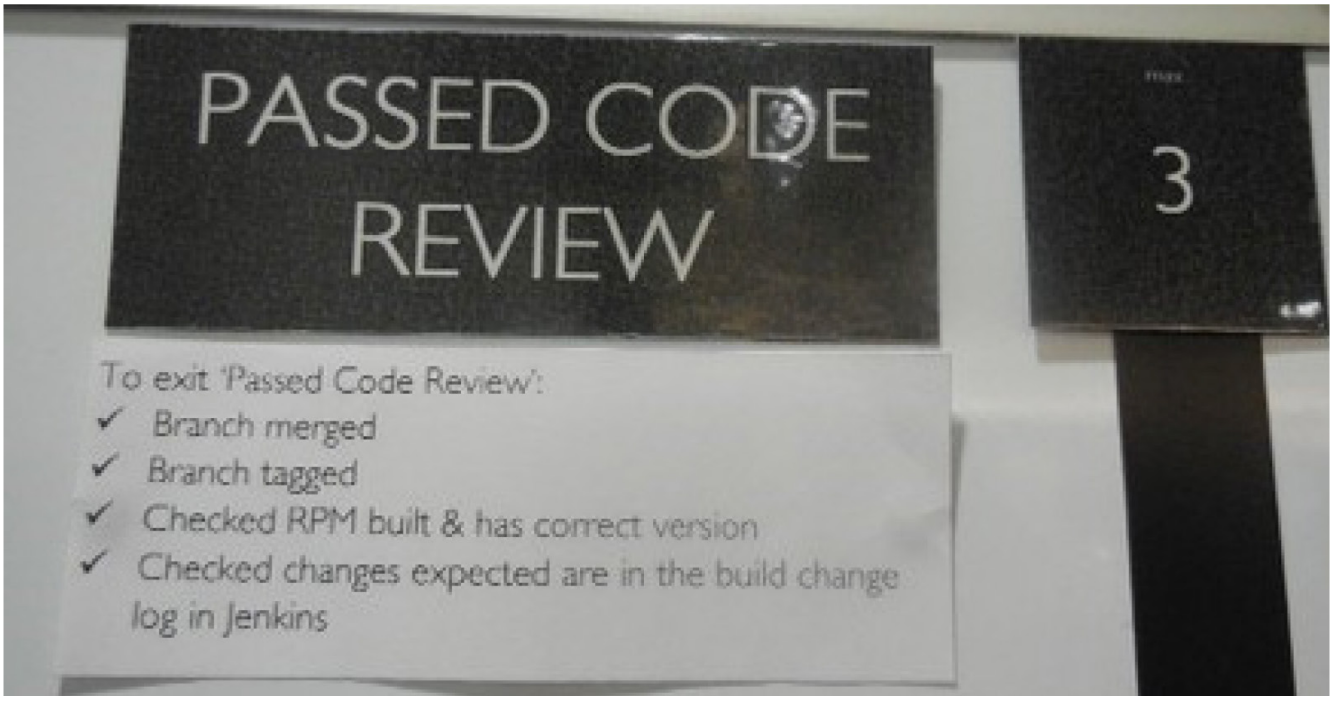
The work-in-progress limits that were initially set proved high. Auditing their processes carefully, they figured out that they could share work-in-progress limits across multiple columns. The team introduced the concept of columns that act as buffers, such as Passed Code Review (Fig.3), where items could be parked until a person could pick them up for the next step.



CODE REVIEW

To enter 'Code Review':

✓ Functional and system test changes complete
✓ Developer has sanity tested the solution against the requirements
✓ Developer has rebased branch on upstream to bring it up-to-date
✓ Developer has made documentation changes such as Change log, Deployment instructions
✓ Any DAVE puppet changes have been created to send for review at the same time
✓ Changes have built and passed tests in Jenkins CI



PASSED CODE REVIEW

max

3

To exit 'Passed Code Review':

✓ Branch merged
✓ Branch tagged
✓ Checked RPM built & has correct version
✓ Checked changes expected are in the build change log in Jenkins

Figure 3 - The Passed Code Review column has very specific definition of done and has a shared WIP limit of three with neighbouring column Merged to Branch.

The team's Kanban board eventually had 11 columns (Fig.4).

"Colleagues from other streams still think we are silly to have such a detailed board. But we appreciate this level of detail because we never forget something or leave any item lingering," David says. "Of course, we might well change this in the future as our use of Kanban matures," he adds.

Stand-up meetings became more relaxed and productive occasions. That gave David the confidence to initiate another change he had wanted to implement for a while—the format of the retrospective meetings. They used to last for an hour after each iteration, which, in his mind, was not enough time to allow for the team to agree on concrete action points with realistic timescales. David felt that the team needed to dedicate adequate time and effort to really drive improvements.

Only by doing this would they be able to improve continually. He suggested that the team have a half-day retrospective once a month with an informal get-together for pizza and drinks afterward. The team liked that idea.

During one of the first retrospectives, the team members addressed the issue of what happens when a person is idle and the column for his activity is full. Inspired by the conversation, Johnathan decided to explore the issue further and come up with suggestions for what a developer could do to improve—beyond some testing.

Little by little, the team began to deliver faster and with improved quality. Fewer bugs lingered on the board. Consequently, they began feeling better about themselves. They tried to have stories of only one size, avoiding ones that were either too big or too small.

After attending a certified Kanban class, David got familiar with the various ways to measure and evaluate the performance of completing work items. Finally having data about how long work items took and what happened to them on the way to completion, he felt confident that the team had better insight into reality.

"We knew approximately how long something would take, and we also saw just how approximate that picture was because there was massive variation in tasks that supposedly were the same size. We could finally see and address that," David says.

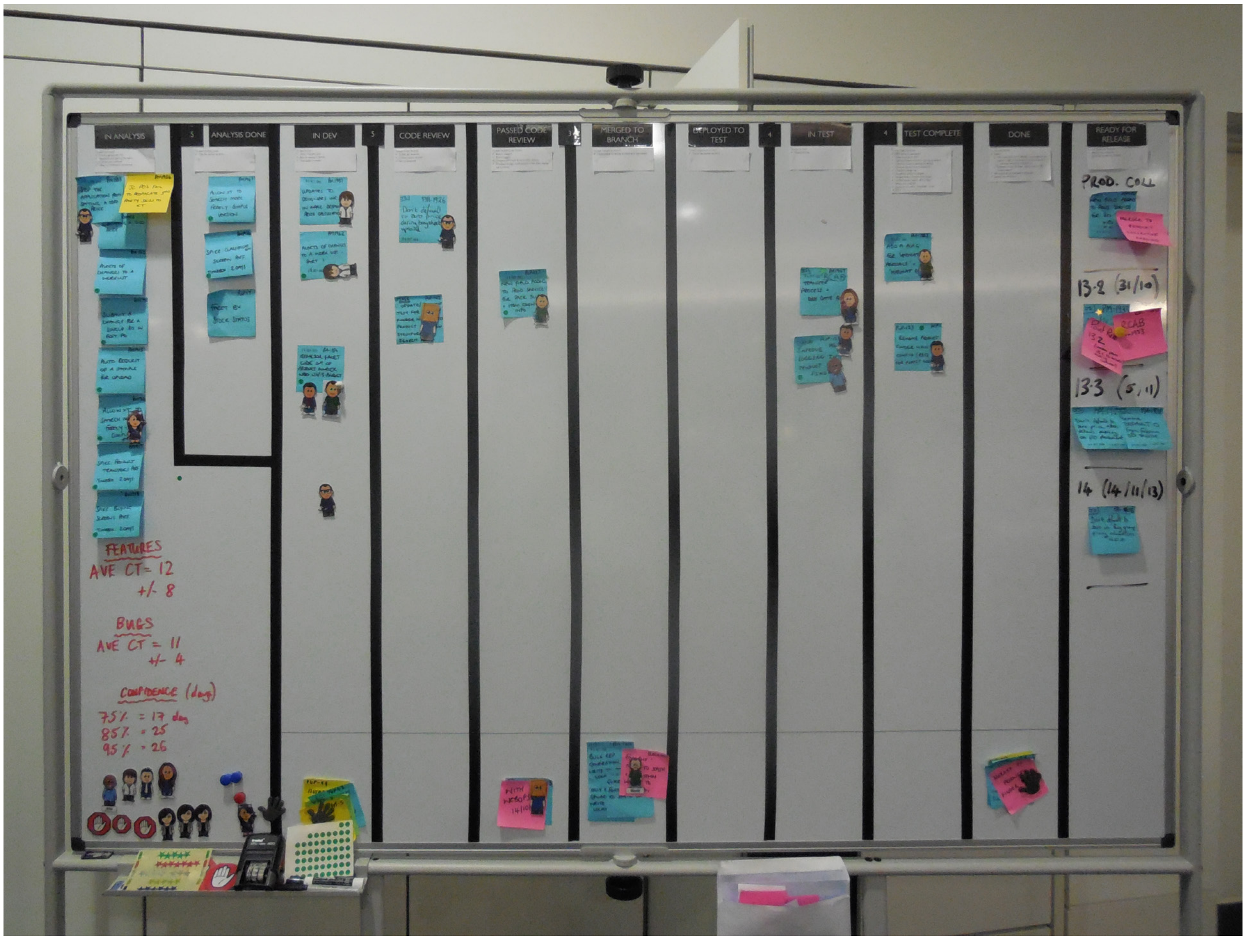


Figure 4 - The Kanban board of the team as of October 2013.

Beyond Self Improvement

One day, in April 2013, Paul Brennan, the product owner for the team, sat down to discuss a new stream of work. Along with the teams for buying, studio, warehouse, and others, he was keen for Product Management to lead an initiative that focused on improving the flow of new products being uploaded to the websites.

The systems were built for many fewer users than they were currently supporting. As they had grown, these legacy systems needed more and more maintenance and support. This affected many people, and it resulted in many small requests hitting the team on an ongoing basis.

The opportunity was huge, and Paul was convinced that with their improved delivery ability, the Product Management team could undertake this project.

Accustomed to an Agile and collaborative mindset, the first thing the team did was talk directly to its stakeholders and ask them what their major difficulties were.

"The time it takes for image and content production; the time it takes to convert and store new product images in multiple sizes; the time it takes to build a page of an internal system" were the sort of answers they got during the bootcamps they held.

In the beginning, the team thought that they would need to devote half of their available bandwidth to the improvement stream, so they split the board in half.

"I was looking at the board and realized that we did not actually need a manual split. The board visualizes what is going on anyway. As long as we indicate the improvement items in some way, and everybody picks from them, we would know at any moment how much we were contributing." David says. Everyone had the freedom to choose to work on improving the system or on another project.

Reviewing and rewriting bits and pieces, the improvements began to show. The stakeholders began to feel the relief.

Emily Kindness, the business analyst of the Product Management team, decided to measure the stories the team was working on not by the amount of time it took them but by the amount of time they saved stakeholders. That idea turned into the "thermometer of time saved."

The time it took for image and content production—which includes styling the products, photography and video, writing the product description, and providing the appropriate sizing guide—was decreased by 25%.

Consequently, the time it takes to convert and store new product images in multiple sizes, which vary from thumbnails to product pages and full size, was decreased from a range of 20 to 60 minutes down to 5 to 10 minutes.

Over the course of a few months, the four developers, two testers, and one business analyst saved up to 20 working hours per week for the page-build time of internal systems.

Marmite

"I have always been excited about data. Beyond the histograms or cumulative flow diagrams that a kanban system provides the metrics for, I was curious [about] how people felt. But not just as a one-off statement, rather, quantitatively," David says.

In November 2012 he came up with what he calls the Marmite survey. He named it after the sticky, dark brown food paste that has such a powerful flavor that people either love it or hate it.

Similarly, at the end of each month, David has been asking the team if they love or hate their job, their team, their processes, and other metrics. According to the survey, from November 2012 until November 2013, the team's happiness has been going up—they are 6% happier with their jobs, 8% happier with the stuff they are working on, and 12% happier with each other.

The Road to Continuous Deployment

"We are in a great position to take control of our own deployment and see our efforts live without waiting for the release cycles," David says.

Having recognized the importance of the Product Management team and their ability to deliver valuable individual stories, the executives of THE NET-A-PORTER GROUP are giving them the green light to deploy their code on a continual basis, independently from the rest of the squads. It is a risky endeavor and it requires a lot of responsibility, as the newly introduced code might conflict with existing systems and crash. But unless they try, they will never know...

Prior to Kanban, the team was in the habit of tying items together that could be delivered only when everything was completed. Most of the interlacing was done early on in the product backlogs.

But the new way of working was a cultural change from the beginning of the process because the only thing that really mattered was an agile delivery of small bits, regardless of when the release was. Now, this new habit could allow them never to have to think of a release date again, and to ship production to their stakeholders independently in a continuous flow.

"One thing we still see is variability in our lead times," David says.

Stories they try to make roughly the same size take varying amounts of time—between five and eighteen days.

"We are hoping to see a drop in the variability once we begin to deploy continuously and not stop our flow for the regression test," David says.

The continuous deployment is set to be initiated in early 2014.

Conclusion

The road traveled by the Product Management team has not been an easy one. It has taken a substantial effort. Kanban has allowed them to figure out a better way to work. There have been no rigid rules, but a lot of thinking and experimentation over what their optimal manner of delivery is.

Simply wanting to be good and be of help to their internal clients has been enough motivation for them to play around with the process and keep on improving. The effect has been not simply that they feel happier with their performance, but as a result, they have clearly affected the performance of many other people in the company.

This team's improved ability to deliver their services has caught the attention of other teams. David is worried that others will want to copy their success, thinking only of its positive effect without realizing the exertion it took. David formed a Kanban Steering Group with people who have used Kanban. Its purpose is to question teams about their motivation for using Kanban and to offer support.

“We do not want Kanban to turn into the shiny, new, one-way to do work. We want to be sure that they really want it for the right reasons. Only then do we commit to helping them make it work for their unique case,” David says.

So far, a total of seven teams are using kanban systems to help their service delivery at THE NET-A-PORTER GROUP. To varying degrees, the Kanban approach is helping each of them.

The evolution continues.

About Kanban University

Kanban University works to assure the highest quality coaching and certified training in Kanban for knowledge work and service work worldwide. Our Accredited Kanban Trainers™ and Kanban Coaching Professionals™ follow the Kanban Method for evolutionary organizational change.

Kanban University offers accreditation for Kanban trainers, a professional designation for Kanban coaches, and certification for Kanban practitioners.

Kanban University
https://www.kanban.university