Adopting Business Agility at Moonpig: A Case Study

INTRODUCTION

In 2017 I had the exciting opportunity to introduce business agility at Moonpig, one of the UK's best known start-ups. Having achieved a measure of success adopting agile practices within our product engineering team, Moonpig's leadership were keen to see if the rest of the organisation could also benefit from agile adoption. For a number of reasons, I thought I should write about my experiences.

Firstly, I've told this story often at conferences and meet-ups, but I've found that many people crave a level of detail that cannot be captured in a 30 minute presentation. I hope this article will allow me to share that detail, explaining how I approached this project, what worked and what didn't.

Secondly, I have benefited enormously from the generosity of the lean and agile community. Learning from the community has enabled me to take on this challenge and achieve some success. I'd now like to return the favour and share my own experiences in the hope that others can benefit in turn.

And finally, I have become frustrated that achieving agility too often seems dominated by what Dan North eloquently describes as "religious methodology". We can be grateful to Scrum for popularising agility and turning it into a mainstream concept. However, Scrum, through no fault of the framework, is often misunderstood, misinterpreted and misapplied. Likewise its scaled counterparts might support agile delivery, but they won't necessarily achieve the full raft of benefits that are at the heart of lean and agile working.

I've been reflecting on why Scrum, LeSS and SAFe have become so dominant, and I believe it's because they appear to provide instructions on how to "be agile". Fundamentally lean and agile are about a set of principles, and translating principles into practice is hard. A set of principles with no guidelines is much like ingredients with no recipe. The Scrum family provides a recipe. But as with recipes, if you don't understand why you need to take each step, you start to skip steps. Scrum fails to deliver when people understand what they need to do but don't understand why.

How do I turn this in to an amazing cake??

In this article I am going to attempt to provide an alternative recipe for adopting and scaling agile. Using Moonpig as a case study I will attempt to provide a series of steps that provide a set of useful guidelines.

Sources of Inspiration

I've been lucky to learn from countless people and organisations, but I'd particularly like to thank Douglas Cook, Chris Downey and Lisa Venter of Skyscanner, Harsh Sinha of Transferwise, Jonathan Smart, Dan North, Barry O'Reilly, the Poppendiecks, John Cutler, Henrik Kniberg, Joakim Sunden, Sean Ellis, Brian Balfour and Jeff Gothelf—to name just a few!

STARTING WITH WHY

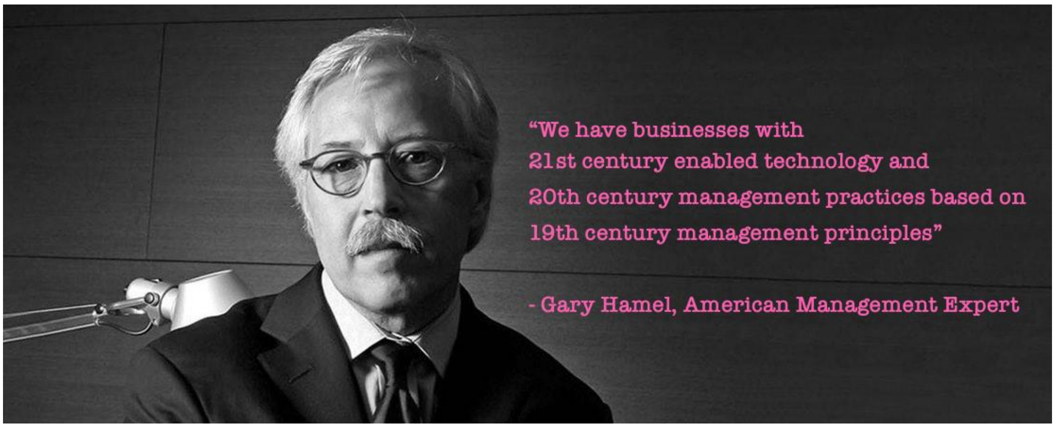
"Very few people or companies can clearly articulate WHY they do WHAT they do. By WHY I mean your purpose, cause or belief - WHY does your company exist? WHY do you get out of bed every morning? And WHY should anyone care?"

— Simon Sinek, Start with Why: How Great Leaders Inspire Everyone to Take Action

Introducing change is hard. Very hard. One way to make it slightly easier, is to take the time to communicate clearly *why* you need to change. At the time we began introducing business agility at Moonpig, most people outside of product engineering had little or no knowledge of lean or agile—and most would have believed them to be “tech things”.

As we started to introduce changes, I wanted to ensure people understood why we were making these changes, and what the benefits would be. I wanted our teams to understand that these changes were not borne of some executive whim, but were driven by changes across industry and were underpinned by solid principles and reasoning.

Why change?



“We have businesses with 21st century enabled technology and 20th century management practices based on 19th century management principles”

- Gary Hamel, American Management Expert

The external factors

As I mentioned in the introduction, Barry O'Reilly is someone that has strongly influenced my thinking. In his keynote talks, there are a couple of interesting facts Barry often quotes:

- 50% of companies that were in the Fortune 500 in 1995 had dropped off the list by 2015
- The average lifecycle of a company in the 1960s was 67 years—today it's 15 years, and it's falling

Having dropped these "bombshells", the point Barry goes on to make is that industry has changed. Whether you consider yourself a technology company or not, technology has fundamentally disrupted the business landscape. It has made it much tougher and more competitive.

Barry argues that to survive and thrive in this new world of business, companies need to change the way they work. They need to be able to move faster, and they need to be able to serve their customers better than their competitors. They need to innovate and they need to learn and fail fast.

While Moonpig retains much of its entrepreneurial, start-up spirit, it is in fact now 18 years old, and is bigger than many of its competitors. It is smaller than many assume, but still big enough to be cumbersome, and as with any other business, it is not immune to being challenged. It too needs to move faster and learn faster to keep succeeding. This is one reason change was necessary. Highlighting these external factors matters; it helps organisations and the people within them understand that change is vital.

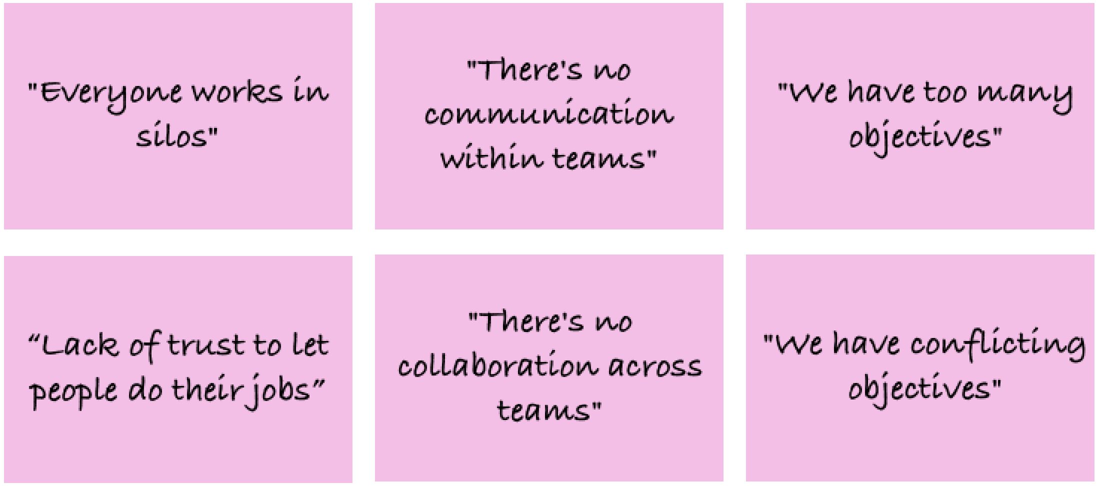
The internal factors

The story of Moonpig's agile journey started several years ago and as with many companies it focused mainly on software development. Within that space we adopted cross-functional teams early on, leveraged agile and devops practices to improve delivery capability and introduced a lean approach to product development—using data to form hypotheses and testing assumptions. This way of working drove big benefits—as well as improved efficiency we delivered better outcomes and healthy business growth. In staff surveys, teams leveraging lean agile practices also showed significantly higher levels of engagement.

At the time we were based in offices in Southwark, and our office happened to have a wall down the middle of it. I came to think of this wall as highly symbolic. On one side of the wall we had our product engineering teams, on the other side of the wall we had our Marketing and Commercial functions.

Having spent the first few years of my Moonpig life firmly on one side of the wall, I had started to take for granted the benefits that agile working had delivered. This is not to say we were perfect, but there were a lot of positives. People worked at a sustainable pace, they enjoyed what they did, they were highly collaborative and there was a hunger to learn.

As I began to explore the potential of business agility, I started to spend much more time on “the other side of the wall”. And it was a quite a contrast. These were some of the things I started to hear:



"Everyone works in silos"

"There's no communication within teams"

"We have too many objectives"

“Lack of trust to let people do their jobs”

"There's no collaboration across teams"

"We have conflicting objectives"

I’d imagine these sorts of comments would be familiar to most large scale organisations. The benefit of this feedback, however, was that it proved our current approach in those areas wasn’t optimal. It supported the case for change.

“Insanity is doing the same thing over and over and expecting different results.”

- Unknown (apparently this has been incorrectly attributed to Einstein!)

My experiences on both sides of the wall taught me that we had in some ways become a hybrid organisation. We had two different mindsets and two different ways of working—effectively we had two different cultures. The contrast between these two worlds suggested that agile working seemed to be a better system, and it inspired Moonpig’s leadership to adopt agility across the wider organisation.

WHAT AND HOW?

Having outlined why we needed to change, I'll now focus on the "what" and the "how". I will attempt to define my vision: what I was actually hoping to achieve. I'll also describe the strategy—how I planned to deliver that vision.

"The best person to change your organisation is you, and the best way is however you figure."

- Dan North, In Praise of Swarming: Scaling Without a Religious Methodology

What?

Defining the vision

In order to organise my own ideas, I realised it would be helpful to have a clear vision for what I wanted to achieve, together with measurable outcomes. I began with a mission statement:

"I want to design a tailored system of work that optimises the entire organisation, allowing Moonpig to innovate and move fast at scale, whilst still ensuring it is a place that people love to work."

How will I know if we have succeeded?

The outcomes I wanted to achieve could neatly be summed up in three words: better, faster and happier. For this I am indebted to Jonathan Smart. I first heard Jon speak at the excellent SEACON conference, when he described his approach to agility at Barclays. He explained how they no longer used the “A” word. Instead of talking about being Agile, they talked about being better, faster, cheaper and happier. I think this is a brilliantly succinct way of capturing the benefits of being lean and agile.

I have taken this a step further and reduced it to 3 words which represent the 3 outcomes I wanted to achieve:

- **Better** = better outcomes leading to increased ROI
- **Faster** = reduced cycle time across all value streams
- **Happier** = higher employee engagement

How?

Developing a plan

This is essentially what I thought of as my strategy—how I planned to go about delivering these outcomes.

Getting better

I believed we would get better by:

- Embedding a customer-focused, data-driven, experimental approach to minimise wasted investment.
- Understanding where we create value and reduce time wasted on low value output.
- Increasing innovation through the collaboration of people with different skills and expertise.

Getting faster

I believed we would get faster by:

- Aligning relevant people around key outcomes and removing conflicting priorities and dependencies.
- Leveraging lean working practices—visualising work, reducing work in progress and focusing on finishing.

- Championing a culture of collaboration and cross-functional working where team success comes before individual glory.
- Embedding a continuous improvement mindset, seeking to constantly optimise our working processes.
- Empowering and supporting teams to self-organise, removing dependencies and bottlenecks around around senior management.

Getting happier

I believed we would get happier by:

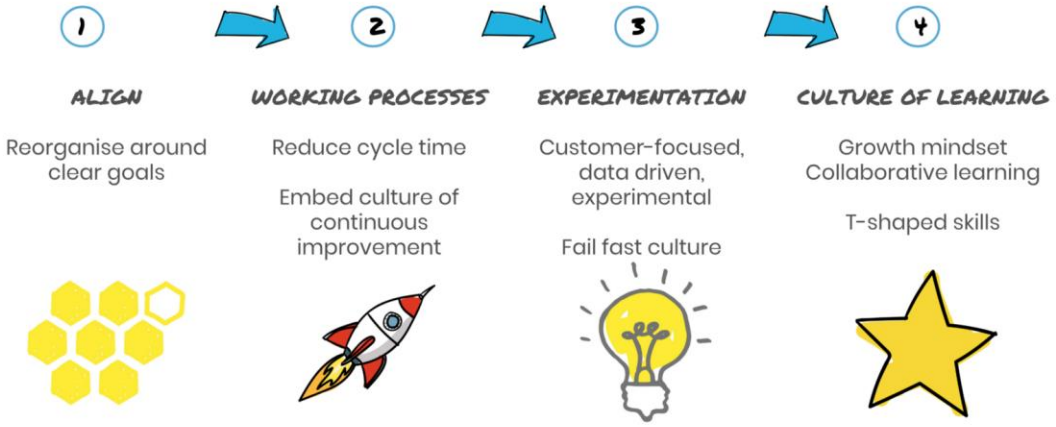
- Developing a safe-to-fail environment where people can take risks.
- Ensuring our teams have clear goals and are supported to achieve them.
- Creating a culture of autonomy where teams are empowered to use their collective skills to deliver outcomes.
- Encouraging a growth mindset and making learning a central part of our working life.

The “Roadmap”

A common misconception about agile is that it involves no planning. This is nonsense. Agile involves a lot of planning, but it advocates changing a plan as you gain more information. There are various reasons you should have a plan. Firstly, you’re more likely to succeed if you know how you’ll go about it. Secondly you will win more confidence and influence from those around you if you have a clear plan.

When I first started exploring the possibility of business agility, I spent a lot of time with our commercial and marketing functions. I realised fairly quickly that there were plenty of opportunities to introduce lean and agile working practices, but that it would be very difficult to make them work within the existing team structure.

I also realised from my experiences with product development teams, that there is a sensible order in which to introduce improvements.



1. Alignment comes first—aligning people around goals and outcomes reduces dependencies and conflicting priorities and thus immediately delivers increased speed. Secondly, it provides a much easier context in which to introduce lean and agile working practices because it enables better collaboration and communication.

2. Having aligned teams effectively, the next step is to support them in developing the practices and processes that will enable them to optimise the efficiency of their workflow and reduce time to deliver value. Introducing a healthy process also reduces chaos and provides structure and clarity which helps create a happier working environment. Reducing chaos buys a lot of faith and goodwill, and this paves the way for more change. Improved delivery capability also builds confidence and trust within the leadership team, and this in turn supports the safe-to-fail environment which is critical for bold experimentation.

3. Experimentation comes after speed simply because, in order to fail fast, you need to be able to deliver fast!

4. To a degree learning happens in parallel with all these steps—learning how to work differently, learning from experiments, learning about your customers and learning from one another. However, to really double down and build a culture of lifelong learning you need people first to understand the benefits of learning and developing a growth mindset, and secondly they need to have time to learn. Agile and lean working practices provide a way of working at a sustainable pace. A healthy system of work will have some slack and that provides time for self-learning. In addition, introducing the concept of continuous improvement and adopting an experimental approach helps to challenge the fixed mindset. At this point you can accelerate learning initiatives and start introducing the concept of t-shaped skills.

Sharing the vision and strategy

One regret I have is that, whilst I developed a clear vision and strategy, I didn’t share it with the leadership team. In hindsight I suspect that defining clear, measurable outcomes, and explaining how we would reach them would have helped de-mystify the process of “being agile”. It would have provided vital education and helped them understand how they could effectively support the changes. Whilst I was privileged to have great support from the leadership team, I think sharing the approach could have made the process of change much easier for them.

GETTING STARTED

Why align?

"Business and human endeavours are systems...we tend to focus on snapshots of isolated parts of the system. And wonder why our deepest problems never get solved."

- Peter Senge, Systems Scientist, Lecturer & Author

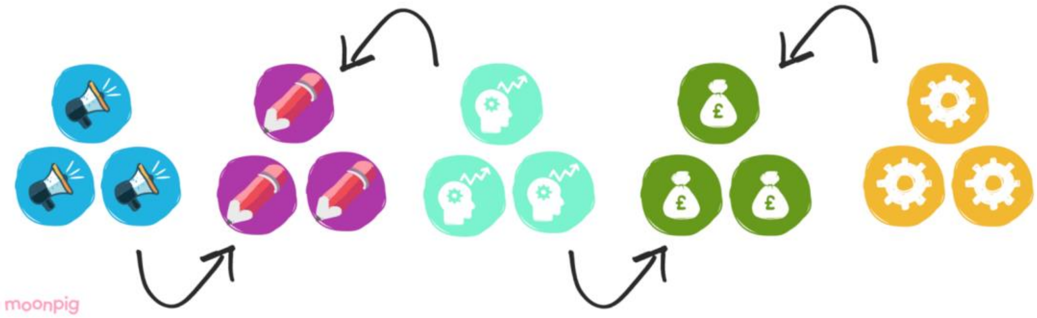
Beyond the obvious reasons why alignment is sensible, for me alignment was critical in order to promote speed. I believed this would help achieve one of our key outcomes - to get faster.

Outside of the world of agile software development, cross-functional teams are less common. I had one advantage at Moonpig in that we had introduced a flavour of cross-functional working through our home-grown Honeycomb framework. This meant the concept and benefits were familiar. However, it's worth clarifying these benefits, and this is how I explained them to our teams.

Why cross-functional teams?

Functional teams

Historically businesses have tended to organise themselves by function and to group people by skillset. The main drawback from functional teams is that they are liable to slow you down. Very often a single function is unable to achieve a strategic goal without being dependent on one or more other functions. This begins to create a network of dependencies, and necessitates a reliance on project management to manage those dependencies and the flow of information between teams. This tends to be exacerbated by conflicting goals which means functions don't share the same priorities—put simply they are not aligned. This leaves the different functions of your organisation pulling in different directions.

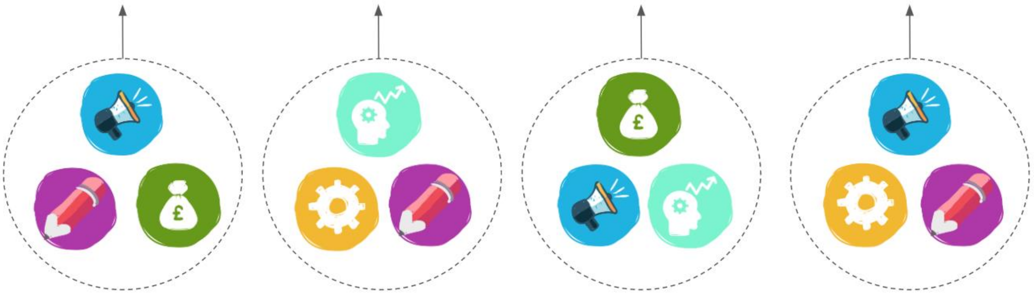


Functional teams often result in multiple cross team dependencies

The functional model focuses on resource efficiency—the ideal being that 100% of people are 100% busy 100% of the time. This is deemed cost efficient as it keeps staffing costs as low as possible. However, focusing on resource efficiency as your primary metric comes at the expense of flow efficiency. Flow efficiency can be simply understood as how long it takes to deliver value.

Cross-functional teams

The fundamental difference with cross-functional teams is that you organise people by what you want them to achieve rather than by what they do. In doing so you align people more effectively around your strategic goals and you reduce (and ideally eliminate) dependencies and conflicting priorities between teams. This liberates individual teams to move at speed.



Cross-functional teams ideally contain the necessary resource to enable them to operate independently

The cross-functional model focuses on flow efficiency—it prioritises time-to-value over resource efficiency. Indeed, in a healthy cross-functional system there will be some slack. People will *not* be 100% busy 100% of the time. This is a counter-intuitive concept but it is essential to making this model work successfully. As soon as you start to focus on resource efficiency you risk creating dependencies and introducing those bottlenecks that increase the time it takes to deliver value.

Slack time might seem wasteful, but it can prove invaluable. As well as ensuring a sustainable pace of work and avoiding burnout, it gives individuals time for personal development—time to learn. Companies that champion learning will be rewarded with a more effective, more engaged and more flexible workforce. Indeed learning provides one solution to managing the resourcing challenges that a cross-functional model presents by enabling the development of cross-functional, or T-shaped skills.

Understanding the context

Before attempting to make any changes you need to understand the problems that currently exist. As I mentioned earlier, I spent the best part of 6 months working with functions outside of technology. This gave me insights in to the way they worked and the problems they experienced. This was invaluable in understanding what changes we needed.

It became clear to me quite quickly that we could certainly extend a lean approach to these teams, but that their current structure would make it very difficult. We were set up in by functions, some of which were unable to deliver effectively because they were dependent on other teams or functions.

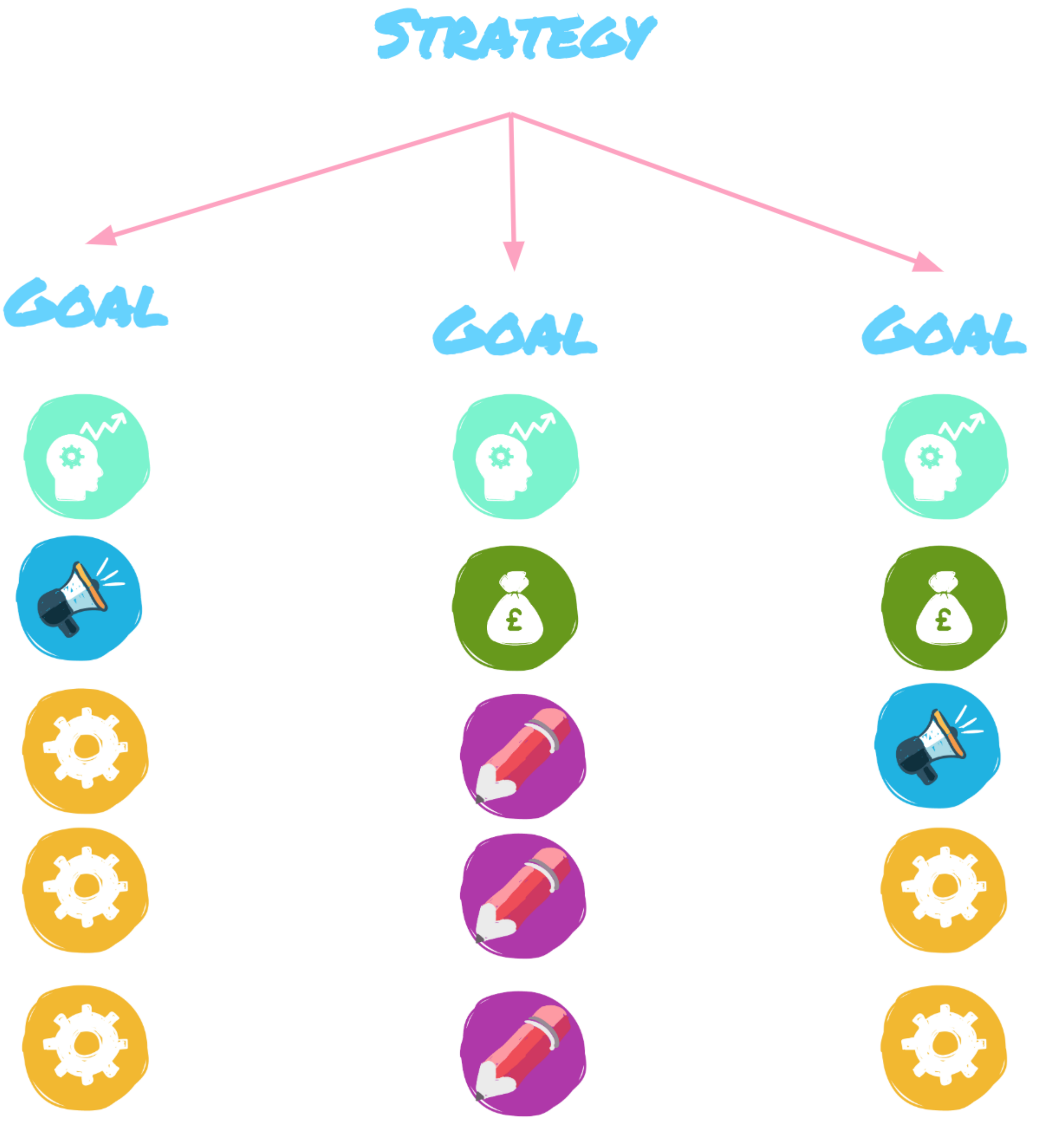
Identifying value streams and core metrics

My approach to reorganising our teams was to understand our core growth metrics and the key value streams that supported them. With this in place it was relatively easy to understand what our teams should look like.

I began by identifying what I believed were a set of long-lived metrics that captured our business value. This was very much about the “why” not the “what”. Outcomes like retention or acquisition rarely disappear unless your business model fundamentally changes. What you do to influence an outcome may change relatively frequently, but the outcome itself remains constant.

This was a key consideration because I wanted to introduce a model with some stability and longevity. There is a cost to commissioning and decommissioning teams. It takes time for a team to mature and become high performing; if you are constantly changing your teams you must reinvest this time over and over again. Not to mention which you introduce the risk of change fatigue.

There is often a tendency to organise teams around architecture or product. Unfortunately customer problems and business outcomes rarely conform neatly to these. Organising teams around outcomes does present challenges in terms of product and code ownership, but these are not insurmountable. Once you have teams aligned around outcomes, you can then start to work out a model of code and product ownership that will support it.



Once I had a set of metrics in place, it was relatively straightforward to understand the mix of skills we'd need to deliver against each one. I shared this with our leadership team, and we then spent some time refining first those metrics, and then figuring out which people we needed to support each metric.

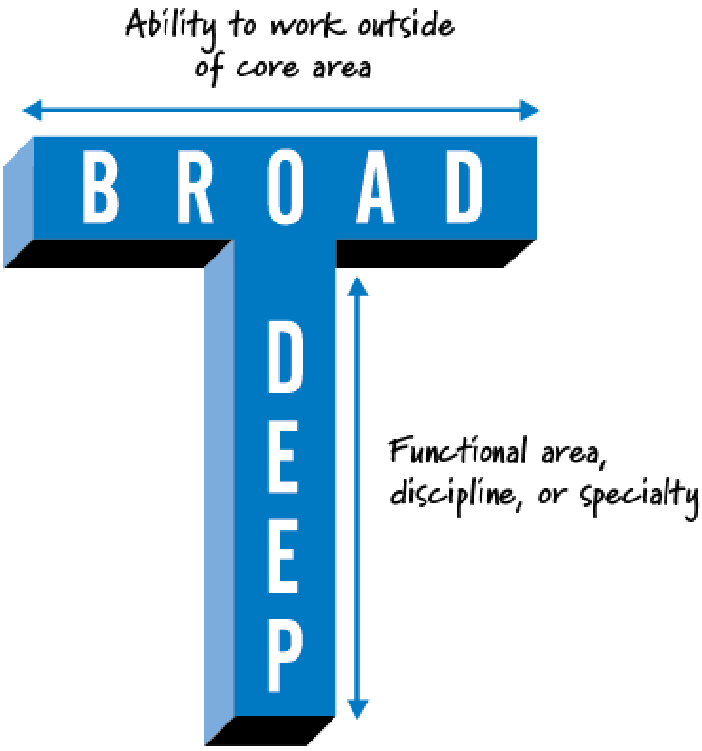
Resource Bottlenecks

One of the key challenges of devising the new teams was resource bottlenecks. We were not attempting to do more with the new teams—if anything we were much more streamlined. However, as we started to identify the skills each team needed, resource bottlenecks were brutally exposed. Whilst creating cross-functional teams didn't solve this problem, it made us aware of it—and once you are aware of a problem you can start to solve it. I'm often asked about this question of resource, so it's worth explaining how we tackled it.

Copywriting was one area in particular where we lacked resource. In some cases it was feasible for a single copywriter to work across multiple teams. There is no problem doing this if neither team involved needs a full time copywriter. However, as soon as the copywriter starts to become a bottleneck they will slow one or both teams down. At that point you have three immediate solutions:

1. Accept that either one or both of your team outcomes are going to be delayed
2. Hire additional resource
3. Do fewer things

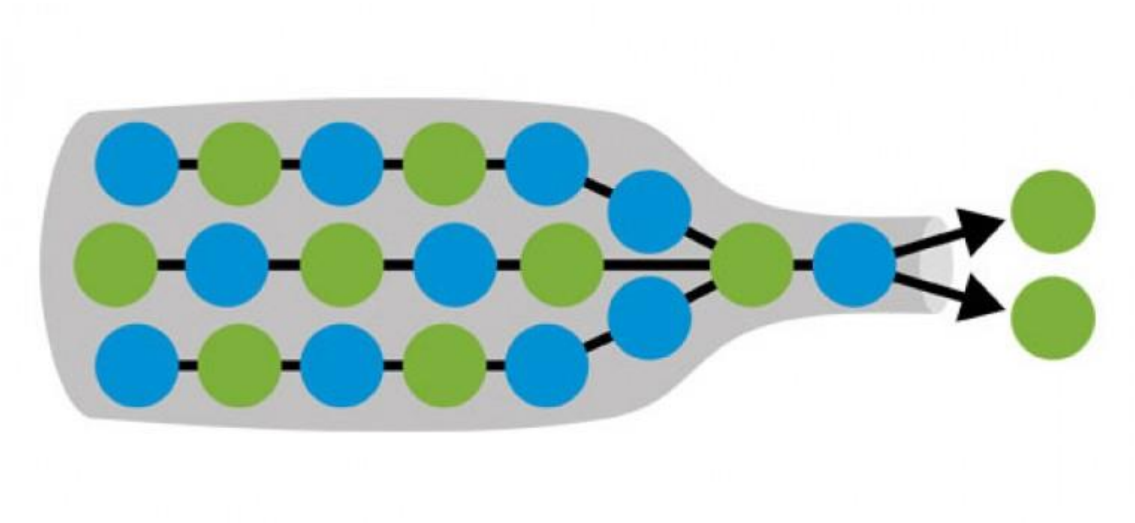
A longer-term solution is to invest in up-skilling your existing resource to develop more people with t-shaped skills. So in the case of copy, for example, a long-term solution would be to invest in training some of our marketers and UX designers to write copy themselves. This wouldn't necessarily remove the need for dedicated copywriting expertise, but simpler copy tasks could be handled by others.


Ability to work outside of core area
BROAD
DEEP
Functional area, discipline, or specialty

T-shaped skills involve developing a breadth of skills as well as a depth of expertise in one or more specific areas.

In the short term, however, your choices are limited, and none of the options are necessarily easy. In our case we did hire more copywriters. However, more people won't always be an option.

Product engineering has always been the biggest resource bottleneck at Moonpig. As an e-commerce company, the business is entirely driven by technology and engineering resource is therefore in high demand.



No matter how well resourced you are, the list of things you want to do will always be longer than the list of people to do them. The solution is always to focus and prioritise—the more things you try to do simultaneously, the slower you will move. Limiting your work in progress at the level of project or initiative is vital.

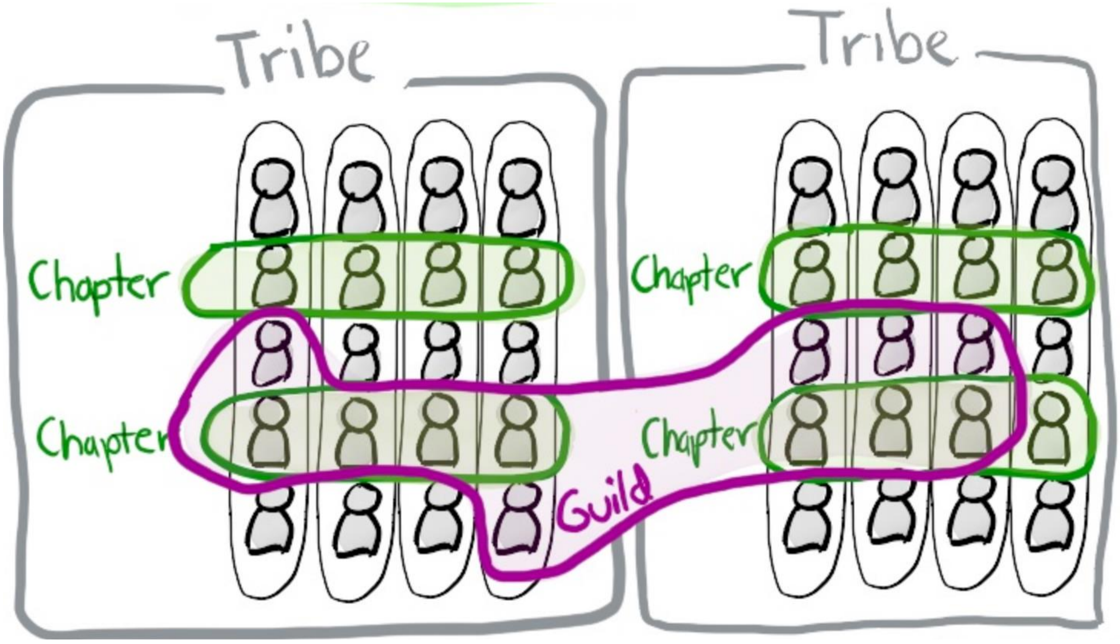
Resource your outcomes in the order of their priority; once you run out of resource, put the rest of your outcomes in a backlog. This will allow you to generate the most impact fast. The faster you deliver your prioritised outcomes, the faster you get to the next items on your backlog.

This is hard to do because it is counter-intuitive. The assumption is always that the sooner you start something, the sooner you will finish it. But in fact when you start too many things simultaneously you spread yourself too thin and you simply slow everything down. It takes discipline, but the key to moving faster will always be to focus on fewer things.

SQUAD PRINCIPLES & ROLES

Having talked about the process of moving from a functional to a cross-functional structure it's worth describing how we used Spotify's model of squads and tribes as a guide. What we ended up with was quite different from Spotify, but it relied on the same principles. I'll also outline the values and roles within our squad model.

The "Spotify Model"



The "Spotify Model" of tribes, squads, chapters and guilds

Spotify became the unwilling role models of how to scale agility when they published videos describing their engineering culture back in 2014. Since then many have sought to copy their approach with varying degrees of success. I was certainly influenced by Spotify's approach, but it's worth clarifying the nature of that influence.

Joakim Sunden, formerly an Agile Coach at Spotify, runs training courses on how Spotify approach agility, and I was lucky enough to attend one of these. The key point Joakim makes is that attempting to copy and paste Spotify's solution will likely result in failure. He urges people to take inspiration from Spotify, but to experiment and tailor their own solutions. And that's ultimately what I sought to do.

What we ended up with was very different from Spotify, but it shared the same principles. It's worth noting that Spotify are just one of a number of companies that influenced my thinking — Skyscanner were also a tremendous source of inspiration, particularly in how we organised teams focused on marketing. Learning from other companies is invaluable, and I am constantly researching how other companies approach agility and looking for ideas that can be adapted for Moonpig's particular context.

What’s in a name?

As any good software engineer will tell you, good naming is important. When we designed our new cross-functional teams we did choose to adopt Spotify’s tribes and squads terminology—and that proved to be a double edged sword. Both internally and externally there were certain perceptions of what a squad is — it’s an “engineering thing”, or a squad must contain engineers, or squads only work on creating new growth levers.

We did consider developing our own naming convention, but at the time I reasoned that, having come up with a new names, we’d simply end up by saying that “it was like a Spotify squad”. In other words, why reinvent the wheel? Given the confusion it caused, in hindsight I might have done that differently. As it was, I eventually came up with a clear definition of a Moonpig squad—something I should have done at the very start. Whilst it was clear to me, I hadn’t communicated it well enough to everyone else!

Definition of a Moonpig Squad

So in the interest of clarity, this is the definition of a Moonpig squad:

A Moonpig squad is organised around a value stream. It has a clear mission and is resourced to achieve that mission independently.

If a mission doesn’t require software engineers or product management, there won’t be any product engineering in the squad. And conversely, a mission which is product or tech lead may not need support from any business function. The mission and outcome determine the composition of the squad.

In addition to long-lived mission and independent resource, there was third key principle of a squad. We wanted squads to the have autonomy to decide how to achieve their mission.

“Move the authority to the information, don't move the information to the authority”

- Dan North, Organisational Change Specialist

Whilst leadership are responsible for devising strategy and defining outcomes, the squads themselves should be trusted self-organise to achieve those outcomes. There are a number of reasons why this matters.

1. Squads will be closer to the data and closer to the problems, which makes them better placed to solve them.

2. If there is a need to constantly refer back to management, management become a bottleneck that slows the squads down.

3. Autonomy is a key tenet of motivation. To achieve high engagement levels amongst staff, you need a high trust system that supports autonomy.

Squads in name only

For the most part, the squads we designed conformed to these principles. However, we had an added complication. Moonpig is part of the Photobox Group, and within the group there are group functions that serve multiple brands. Functions such as Shipping and Supply Chain, for example, operate at group level.

This was a challenge because we weren’t able to incorporate those group functions in to our cross-functional model. That meant some squads were squads in name only. Some were heavily dependent on group functions so we were unable to give them the full independence of a true squad. However, the reasoning that drove the squad model did encourage those teams to have a conversation with those group functions about how to better align and streamline their workflow. Not necessarily a perfect solution, but certainly an improvement.

Inspired by Moonpig, other brands within the Photobox group also started to adopt cross-functional teams. Should that trend continue, group functions might well adapt to become better aligned with individual brands.

Tribes

In our first iteration of squads, there was very little emphasis on the concept of tribes. We had a loose idea of tribes that helped us in identifying the various squads we needed, but they didn’t play much of a role beyond that. In our second iteration tribes became a much more important concept, and I’ll cover that evolution later in this article.

Squad Roles

Squad Leads

At the inception of the squads we identified a need for a squad leader—someone that could harness the talent and coordinate the work of the squad. The leader was responsible for helping the squad to decide how they would achieve their goals, as well as sharing learnings and progress with the wider organisation. You can read the full squad lead description here.

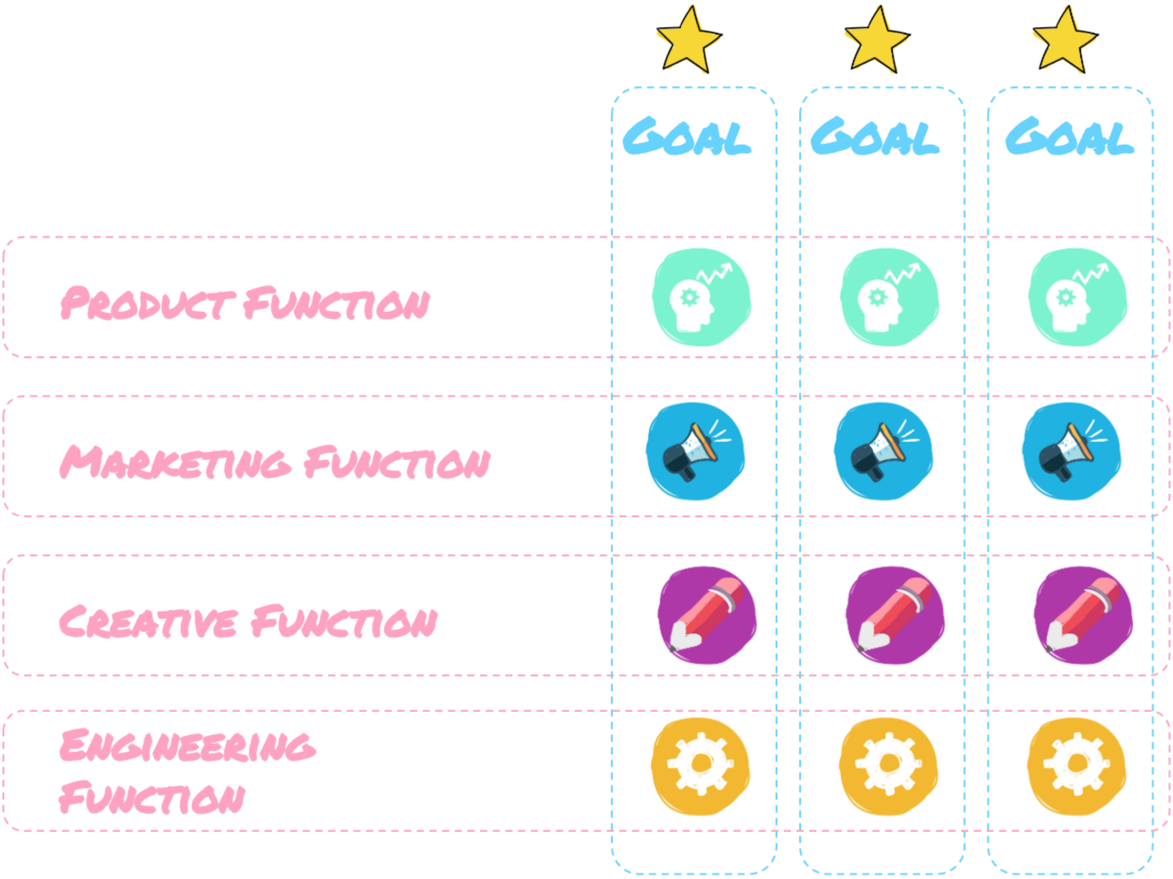
Squad Sponsors

In addition to a leader, each squad had a sponsor—sponsors later became tribe leads, but the role remained the same. The squad sponsor was a member of the leadership team. Their role was to support the squad by providing advice and guidance, as well as being the first port of call for problems such as blockers or resource issues. You can read the full description of the role here.

Functions & Function Leads

When introducing cross-functional teams, one common fear I encountered was that splitting functions across multiple teams would result in inconsistency and chaos. That is a legitimate concern and a key challenge of working in this way.

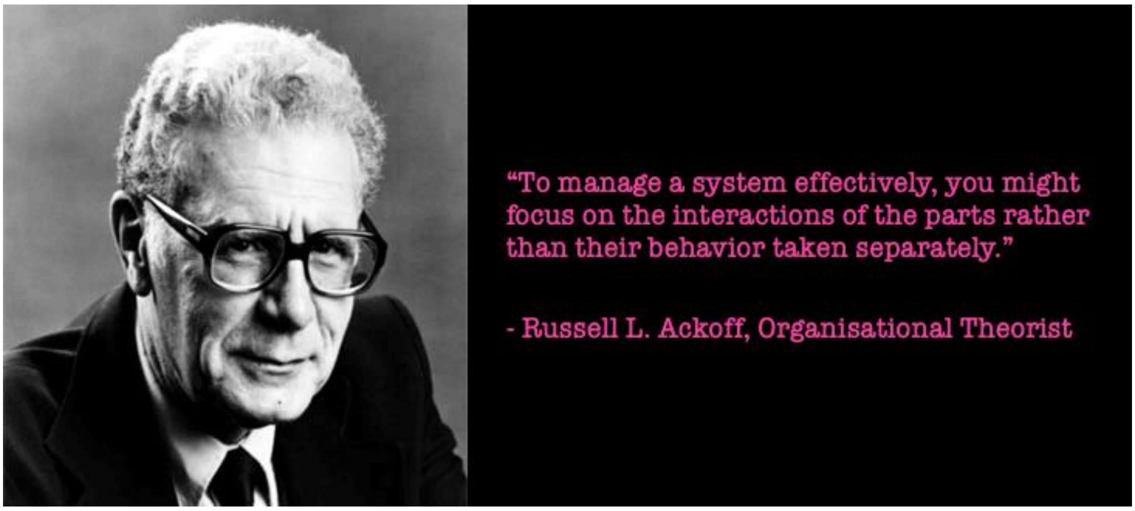
It's important to understand that in a cross-functional world, functions don't disappear. They still have a critical role to play. Functions provide the guidelines, frameworks and principles within which members of the function can operate autonomously across multiple squads. The engineering function, for example, will be responsible for defining preferred technology and coding standards. A creative or brand function will define clear brand guidelines. Functions provide the boundaries and constraints that enable consistent, high quality output—individuals within squads can then operate autonomously within these boundaries. To help embed this concept, we outlined the responsibility of a function lead, which you can read here.



The vertical represents a squad aligned around a goal. The horizontal represents the various functions to which individual squad members belong.

WORKING IN SQUADS

One intention of squads was to break down functional silos, but we wanted to ensure that individual squads themselves did not become silos! To that end we developed a broader framework in which the squads operated— this included ceremonies and practices designed to provide visibility and coordination between squads.



"To manage a system effectively, you might focus on the interactions of the parts rather than their behavior taken separately."

- Russell L. Ackoff, Organisational Theorist

The Kick-Off

At Moonpig we used the OKR framework, and at the time we were reviewing OKRs every 6 months, so the year was divided in to "H1" and "H2". To provide a broad overview of what the organisation was doing, we organised an "H kick-off".

Whilst it was important for everyone to understand what their squad was doing, and how it supported the company strategy, we also wanted them to have visibility of what other squads were doing. The event itself was fairly short and straightforward. We began by reiterating the strategy and then each squad spent 3–4 minutes explaining:

- Who they were
- What their long lived purpose was
- What their short term goal (OKR) was
- What they planned to do to achieve that goal

This helped give everyone a high level overview of each squad's plans and how they were aligned with the company strategy. The intention was to repeat this exercise at the beginning of each H.

The Showcase

A kick-off was useful, but we also wanted to provide regular updates and knowledge sharing to build on this.

In the pre-squad world we had a weekly tech demo in which the product engineering teams would give an update and demo what they had been working on. In the squad world we repurposed this to become a company wide showcase. This was held every two weeks, and alternated between squads. That meant we would have an update from each team every 4 weeks.

Everyone in the company would attend the showcase, and for teams presenting, the brief was to:

- Showcase anything new — new physical products, new features, new marketing content
- Share noteworthy learnings — for example an experiment that delivered unexpected results, positive or negative, and what was learned from it

Fundamentally this was about providing visibility and knowledge sharing. Even if teams didn't feel they had anything particularly exciting or visual to present they were still encouraged to talk about how they'd spent their time in the previous 4 weeks.

The Squad Leads Stand-up

This was a 10 minute weekly gathering of squad leads wherein each squad lead would provide a high level update on what their squad would be focused on for the next 1–2 weeks. It provided visibility and identified opportunities for cross-squad collaboration and highlighted potential conflicts.

The Monthly Check-In

The monthly check-in was intended as a "reporting and supporting" session. Each squad had a check-in once every 4 weeks, attended by all squad members and the Moonpig leadership team. The first half of the check-in covered the "reporting" element and this consisted of:

- Providing an update on progress towards the team's goal
- Highlights and lowlights of the past 4 weeks
- Celebrating successes, sharing learnings from failures
- Discussing forthcoming plans

The second half of the check-in was dedicated to "supporting". This provided the squad with the opportunity to seek guidance, advice and direction. It also provided an opportunity to raise problems and blockers—to raise problems external to the teams that they needed help addressing.

The Function Meeting

As mentioned earlier, it’s important to maintain strong functions within the cross-functional world and regular gatherings support this. The recommendation was that functions meet at least once every two weeks, if not more often. The purpose of these gatherings was to:

- Provide visibility of what individuals were working on across different squads
- Learn and knowledge share—this might be around driving business outcomes or around working practices and processes. Functional gatherings support cross-pollination of good ideas and help breed consistency.
- Continuous improvement—this was about driving quality, reviewing how excellence is achieved within a particular function

Functional gatherings could take different forms depending on the function. The UX function, for example, would run a weekly critique session, in which they’d peer review one another’s work and offer constructive feedback. In the case of the engineering function, demos were key—showcasing new technology and implementation.

Introducing Confluence

One key difference I had noticed between product engineering and the other functions was that the mindset around radiating information was very different. Moonpig uses Google Drive and whilst this works well as a way to store and share documents, it doesn’t support finding information easily. One of the most frequent complaints I’d hear was people not being able to find information.

To that end I started to encourage all teams to use Confluence. Confluence is a wiki that forms part of the Atlassian suite of tools. It was not necessarily the best choice of tool, but it was readily available and provided a quick solution.

The idea was to use Confluence as gateway to accessing information. Google Docs were not banned—far from it—but the stipulation was that information stored in Google Docs that was of wider interest was linked to in Confluence.

To get started I created a section in Confluence for each squad. This followed a loose template, and by default included:

- An overview page listing the squads’ long lived mission and OKRs

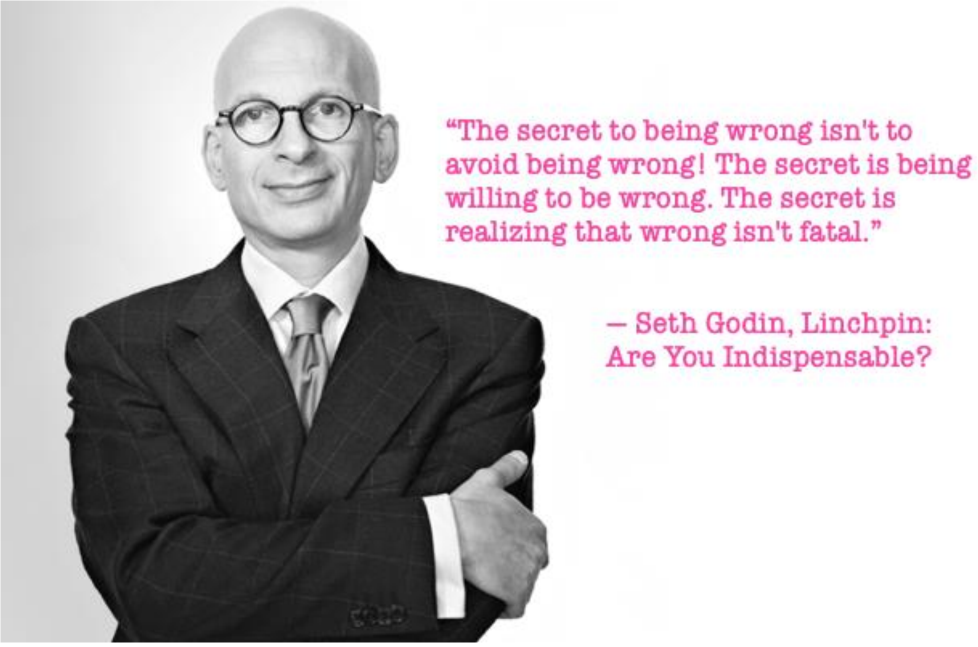
- A people page which listed the members of the team, their slack channel and email d-list
- A metrics page which was used to capture progress against OKRs for the monthly check-ins
- A tracking resources page which had links to the teams' Jira boards and dashboards

The purpose of Confluence was to provide information about every team— who they were, what they were trying to achieve and how they were progressing. I wanted everyone in the organisation to be able to find out exactly what any other team was doing.

We also used Confluence to capture information around user testing, analytics and AB tests. Essentially, any information that might be relevant to a wide range of people could be found and accessed via Confluence.

ITERATING ON THE SQUAD MODEL

Having put our cross-functional model in to operation, we were able to observe what had and hadn't worked and this enabled us to evolve and refine the model further.


"The secret to being wrong isn't to avoid being wrong! The secret is being willing to be wrong. The secret is realizing that wrong isn't fatal."
— Seth Godin, Linchpin: Are You Indispensable?

What wasn’t working?

By and large the first iteration had worked really well, but there were a couple of problems we observed:

1. Some squads were more about planning than execution
2. Some squads had more than one outcome to pursue

Both issues arose out of the gap between my knowledge about certain areas of the business, and the business leaders understanding of exactly what we were trying to achieve with the cross-functional model. We recognised them relatively early on and were able to make the necessary changes.

Despite your best efforts, it’s very unlikely you will get alignment right first time. There is only so much time you can spend planning, eventually you have to put a plan into action and see what works and what doesn’t. What’s important is that you react to what doesn’t work quickly—inspect and adapt.

It’s also important to manage expectations—before we rolled out the team changes I had been at pains to warn everyone that it was unlikely the new model would work perfectly, and that we would want feedback on what wasn’t working so we could respond to it. Preparing everyone for inevitable changes helps them to be accepted.

Beyond these relatively minor tweaks, there was a bigger challenge I needed to address. Some squads were too big to self-organise effectively and struggled to collaborate effectively. This led me to reconsider the structure we had, and how we could maintain alignment without compromising agility.

Tribes, Pods & Squads: Evolving the Model

Defining tribes

Previously tribes had been a concept we’d used purely for planning purposes. Yet over the first few months of seeing the squads in action I’d recognised potential benefit in more clearly defined tribes. Spotify, who developed the concept of tribes, used them as a way to make the organisation feel small as it started to grow. Tribes were used to develop small communities built around a shared purpose.

Whilst Moonpig is a much smaller organisation, it is still big enough to benefit from the concept of tribes. We had groups of squads that shared a common broad purpose, and I could see value in formally grouping those squads and articulating that shared purpose.

I recognised the need for three tribes:

Product & Service
This tribe was focused on developing and evolving our core customer value, both physical and digital. It was focused on customer missions—solving customer problems.

Growth
This tribe was essentially about marketing our customer proposition—getting more people to experience that core customer value. It was focused on business missions, leveraging our products and services to drive acquisition, activation, retention and revenue.

Foundations
This tribe was focused on support missions. We had some big re-platforming initiatives, so while this had high long term business value, it was not delivering new customer value in the short term.

The Product & Service tribe consisted of squads focused on designing and developing our card and gift ranges. This included creating the physical products and evolving the digital products that support their personalisation. Product & Service would also be the home of any new innovation around either physical or digital product.

The Growth tribe consisted of squads organised around the funnel, focusing on areas like acquisition, retention and revenue.

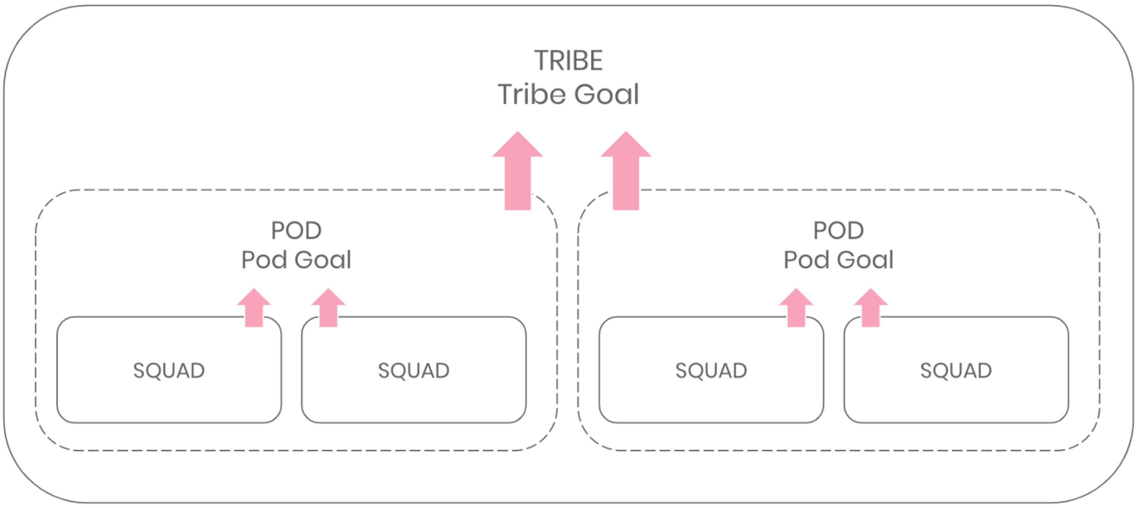
The Foundations squads focused on defining and building core elements of the new platform.

Size matters—introducing pods

It’s a well-known fact within the agile world that smaller teams are more effective—smaller size makes it easier to self-organise. Spotify advocate a team size of 7–9, Skyscanner believe 8 is the maximum effective size. My own experience confirms this, but it also left me with a problem. What happens when we need more than 8–10 people to deliver an outcome?

The solution to this was “pods”. Reflecting on the problem I reminded myself that the core purpose of squads was to align people. Having observed the squads in practice, it was clear that multiple work streams might be needed to support an outcome, and they didn’t have to live in a single squad, so long as the outcomes were shared between relevant squads and alignment was preserved.

A pod was simply two squads that shared the same goal—one team with two work streams. Crucially each pod had a single data analyst—this provided a single source of truth on which hypotheses and experiments could be based.



Squads worked towards a pod goal, which in turn supported a tribe goal

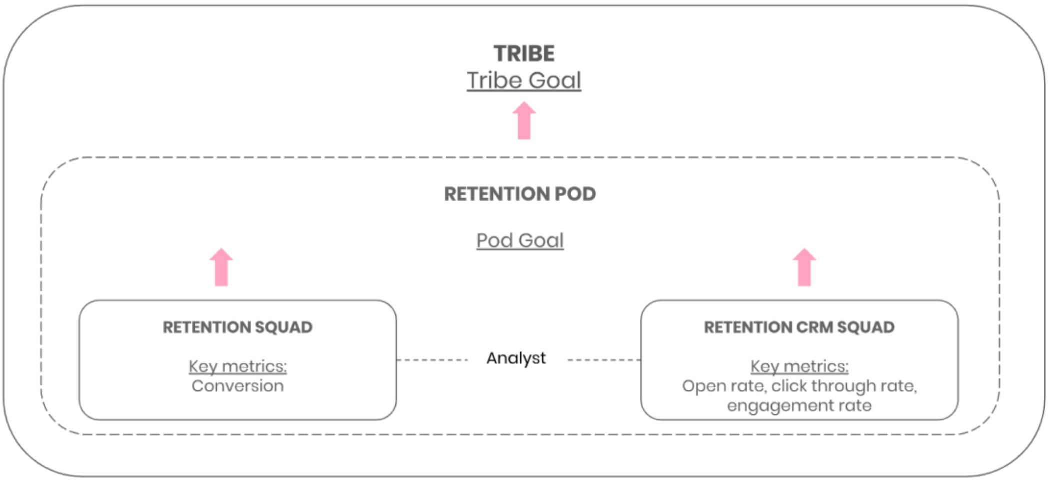
Sharing an outcome ensured the two squads would still collaborate, but were able to execute in smaller, more efficient units. Different pods worked in different ways depending on their goals and initiatives, but I'll describe how the Retention pod worked as an example.

Pods in action

The Retention pod was made up of two squads:

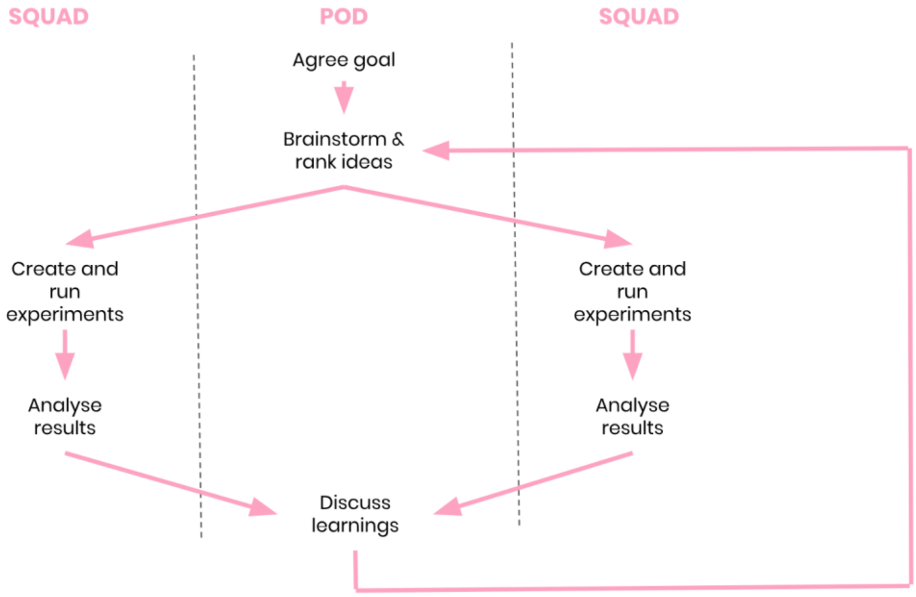
Retention Web — a product engineering squad consisting of a product manager, engineers and a UX designer

Retention CRM — a CRM squad consisting of CRM marketers, a creative designer and a copywriter.



The pod would meet each Monday for a weekly growth meeting. They'd review the tests they'd run previously and analyse the results. Based on the new learnings, they would then meet to brainstorm together and generate new ideas. If the ideas involved experiments around emails, the CRM squad would pick them up. If the ideas involved experiments on the website, they'd be picked up by the product engineering squad. The teams used shared tooling to capture and rank ideas and document analysis and learnings. Experiments themselves were executed within the individual squads, but they were generated by people across the pod.

Where coordination was needed the two squads could self-organise. There was also the freedom to share resource and expertise across the pod. For example, UX designers could help improve the layout of emails. Creative designers and copywriters could develop assets for landing pages on the website.



Aside from speed, alignment and knowledge sharing, the other benefit of this way of working was learning how to work more effectively. Whilst product engineering teams are familiar with the lean approach and the test and learn cycle, it was relatively new to the CRM team. Working closely together enabled the CRM team to learn from their product engineering colleagues and hone their own skills around experimentation. Likewise, the product engineering team were able to improve their understanding of email marketing.

GETTING FASTER

A key benefit of alignment is that it automatically delivers an increase in speed of delivery by dramatically reducing, or eliminating, the dependencies and bottlenecks which so often slow us down. By aligning people around goals and outcomes you essentially streamline your organisation and expose the bottlenecks in resource.

However, beyond alignment there is further opportunity to optimise speed by leveraging lean and agile working practices to optimise individual workflows and value streams. The practices I’ll describe will be familiar to many—anyone that has coached a software development team will recognise these tactics. However, as this article is about adopting business agility, I’m going to focus on how we introduced these practices to squads outside technology.

Flow Efficiency

“In product development, our greatest waste is not unproductive engineers, but work products sitting idle in process queues.”

— Donald G. Reinertsen,

The Principles of Product Development Flow: Second Generation Lean Product Development

Flow efficiency looks at how long an item of work takes to move through your workflow—in other words how long it takes you to create value. The more efficient your workflow, the quicker you can learn and the quicker you can deliver value. Cycle time is the measure we use to understand how efficient our workflow is.

From concept to customer, optimising flow efficiency helps us learn and deliver value faster

In the product engineering world this was nothing new—we'd been measuring cycle time and optimising work flows for some time. However, outside of technology, these were new concepts. I wanted to start optimising all our workflows, so irrespective of whether we were delivering physical products or marketing content, we'd be optimised to deliver as quickly as possible.

Minimum Viable Agility

In 2018 I had the opportunity of attending a course on how Spotify approach agility, facilitated by Joakim Sunden. One of the many great takeaways from the course was the concept of "minimum viable agility". Spotify seek to create an autonomous environment with just enough constraints to avoid chaos. One of these constraints is minimum viable agility.

Spotify don't advocate any particular methodology, but they do have certain things they expect every team to do. Inspired by this concept, I came up with my own version of minimum viable agility. I believed each team should:

- Visualise workflows
- Hold a daily stand-up
- Hold a retrospective at least every two weeks

These were to be the three cornerstones of agile working for any team at Moonpig. Over and above this, I planned to measure cycle time for each team, and to use data to identify bottlenecks in the workflow to help the teams optimise their processes.

Introducing agility outside product engineering

Workshops

As most people outside of product engineering had little or no experience of agile working practices, I ran workshops with various squads to introduce them to the key concepts and benefits.

I used various games that will be very familiar to agile practitioners. We played the coin game to understand the value of working in small batch sizes and we played the name game to demonstrate the importance of work in progress limits. The biggest part of the workshop was dedicated to playing getKanban—one of my favourite tools!

The getKanban board game

If you've not come across it, getKanban is a board game that simulates a kanban workflow. The main advantage of it is that it demonstrates cause and effect in a short space of time, which helps teams understand the consequences of the choices they make. I have found work in progress limits to be one of the most difficult concepts for people to grasp, and getKanban really helps people conceptualise the benefits. It is also a great way for people to start understanding the benefits of communication, collaboration and swarming.

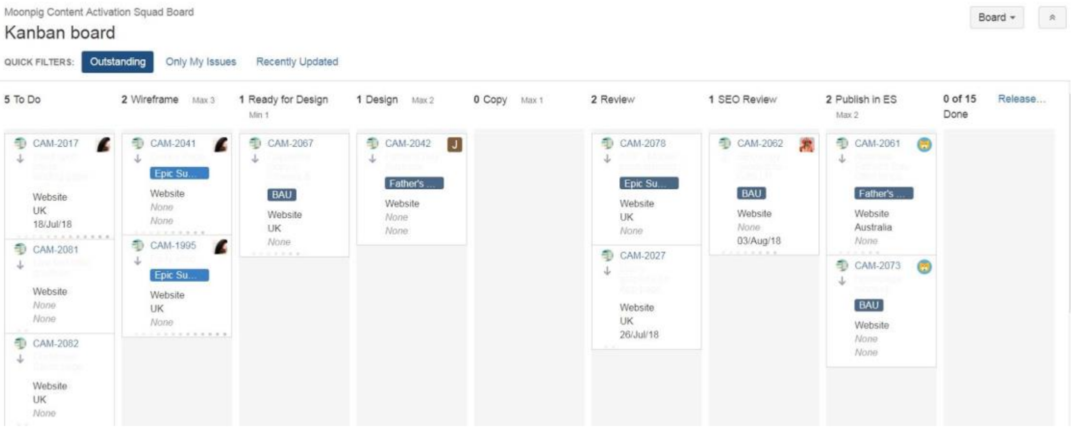
Visualising workflows

Having familiarised the teams with what agile working would look like, the next step was to start visualising workflows. Jira happened to be the tool we were licensed to use, so it made sense to create the workflows there.

I'm often asked why we didn't use physical boards instead, and there are two reasons. Firstly, lack of space! We simply didn't have enough free wall space to accommodate boards for all the different squads. Secondly, for me there is little point in visualising a workflow if you're not going to optimise it. Measuring cycle time and bottlenecks manually across 14 squads simply wasn't viable, so we needed an electronic solution.

The biggest resistance to using Jira was the perceived admin overhead of raising the tickets. However, the same people that were concerned about this were also the same people that cited lack of process and visibility as some of their main frustrations. I suggested that using Jira would provide the visibility and support a better process.

Most teams need a coordinated effort in order to complete a single task. Creating a website landing page, for example, needs a content marketer, a designer and a copywriter. A visualised workflow supports coordination because it provides clarity around the progress of each individual task—what has been done? What is left to do? Who is working on what?



Sample workflow for the Content Activation Squad

In reality, having persuaded the teams to try Jira, they realised the advantages very quickly!

Improving collaboration

With workflows up and running in Jira, I introduced stand-ups to the team. This ensured that at least once a day the whole team was reviewing their workflow and discussing priorities. This helped foster collaboration and encouraged the squads to start working as a team. In the pre-squad world, most people worked individually. Stand-ups helped cement the idea that delivering work was a team effort and that there were benefits in communicating regularly.

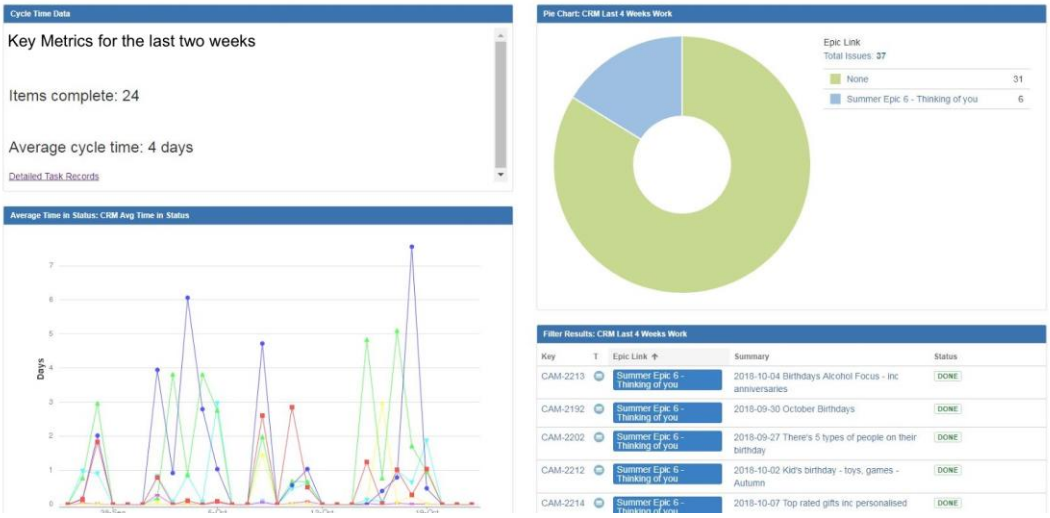
We also supported this collaboration by seating squad members together. Whilst stand-ups are a great way to have a regular team catch-up, putting people together automatically starts to build a rapport between team members and greater communication and collaboration are a natural side effect of this.

Optimising workflows

While daily stand-ups provide great visibility on day-to-day issues and blockers, and provide a regular opportunity to focus a team on finishing work, retrospectives are the cornerstone of process improvement. I'm a great fan of using data to focus retrospectives as it helps teams understand what's slowing them down—it bridges the gap between perception and reality!

We had long since created data dashboards for all our product engineering teams, and I now created equivalents for non-tech squads. The dashboards included:

- Average cycle time over the past two weeks
- Throughput (number of items completed)
- Workflow graph—the average time in status of each stage in the workflow
- Tasks completed in the previous two weeks



Cycle Time Data

Key Metrics for the last two weeks

Items complete: 24

Average cycle time: 4 days

Detailed Task Records

Average Time in Status: CRM Avg Time in Status

Pie Chart: CRM Last 4 Weeks Work

Epic Link
Total Issues: 37

🟩	None	31
🟦	Summer Epic 6 - Thinking of you	6

Filter Results: CRM Last 4 Weeks Work

Key	T	Epic Link ↑	Summary	Status
CAM-2213	○	Summer Epic 6 - Thinking of you	2018-10-04 Birthdays Alcohol Focus - inc anniversaries	DONE
CAM-2192	○	Summer Epic 6 - Thinking of you	2018-09-30 October Birthdays	DONE
CAM-2202	○	Summer Epic 6 - Thinking of you	2018-09-27 There's 5 types of people on their birthday	DONE
CAM-2212	○	Summer Epic 6 - Thinking of you	2018-10-02 Kid's birthday - toys, games - Autumn	DONE
CAM-2214	○	Summer Epic 6 - Thinking of you	2018-10-07 Top rated gifts inc personalised	DONE

Sample dashboard

We'd begin each retrospective by reviewing the dashboard, starting with cycle time. Calculating an ideal cycle time is not easy, but my rule of thumb is to get the team to estimate how long it takes them to a standard task, and then to compare that number to the actual cycle time. There will normally be quite a gap between the team's estimate and the actual cycle time, and this is good! It helps the teams realise that there is room for improvement, and they start to take the idea of cycle time seriously.

To deep dive in to cycle time we would then look at cycle time for individual tasks to understand which ones were pushing the average up.

In addition to cycle time, we'd look at the workflow graph to identify bottlenecks in the workflow.

Reviewing and discussing this data helped the team understand where improvements could be made, and this helped drive useful discussion and actions.

As I said earlier, the value of limiting work in progress is difficult to grasp. Using data to illustrate the impact of WIP on cycle time helps teams to understand the benefits and they are then much more likely to start respecting the limits set.

Sound familiar?

If you've ever introduced lean and agile working practices to a software development team, you may well have read this and thought "Well that's nothing new!" And that's precisely the point. Agile is too often perceived as a "technology thing", but in fact it's a set of principles that are agnostic of technology. Therefore, introducing agile working practices in to a content marketing team isn't any different to introducing it to a software development team.

The one additional challenge is around expectation and perception. Today agile is widely recognised as the "right" way to develop software, and most software developers will expect to be asked to adopt agile practices. Outside of technology there is much less knowledge of agility and its benefits, and there is no expectation that it should be adopted. To that end you do have to work that little bit harder to persuade people to give it a go.

I should also point out that what I have described here is merely the starting point. Most of the non-tech teams, while having adopted minimum viable agility, are a long way from having really optimised workflows. That's really down to a lack of coaching resource—as I was the only coach at Moonpig during this time, there simply wasn't enough of my time to give the requisite dedicated attention to the half dozen teams that needed it. However, the basics are in place and with the right support there is no reason those teams won't be able to improve their speed.

GETTING BETTER

Another key outcome we sought to achieve was "better". Having made some improvements in speed of delivery I'll talk about how we started to introduce experimentation to non-tech teams to optimise the value the squads created.

The case for experimentation

"All life is an experiment. The more experiments you make, the better."
— Jeff Gothelf, Lean UX: Applying Lean Principles to Improve User Experience

As with agile working practices, within the world of digital product development, experimentation is widely accepted as the optimal approach. Testing assumptions and creating minimum viable products is recognised as a sensible way to minimise risk and figure out exactly what your customers want. The growth marketing and growth hacking movement have also introduced experimentation to the world of marketing - digital marketing in particular.

However, beyond the world of product development, the concept of experimentation is still quite new. Yet as with any element of agility, it's a set or principles and processes that are agnostic of product and technology and can be applied in other contexts.

Using experimentation to increase retention

To describe how we started to introduce experimentation, I'll use the case study of the Retention Pod. Earlier I described the retention pod and their working process. The product engineering people in the retention pod had been experimenting for some time, but for their CRM colleagues, experimentation was something new. I'll explain how we set about introducing experimentation to the CRM Retention Squad.

The backstory

Historically Moonpig made great use of TV marketing and it had served them very well. Within the UK, Moonpig has huge brand awareness—75%+. However, the leadership were keen to boost our digital marketing efforts, both to maintain growth and improve ROI.

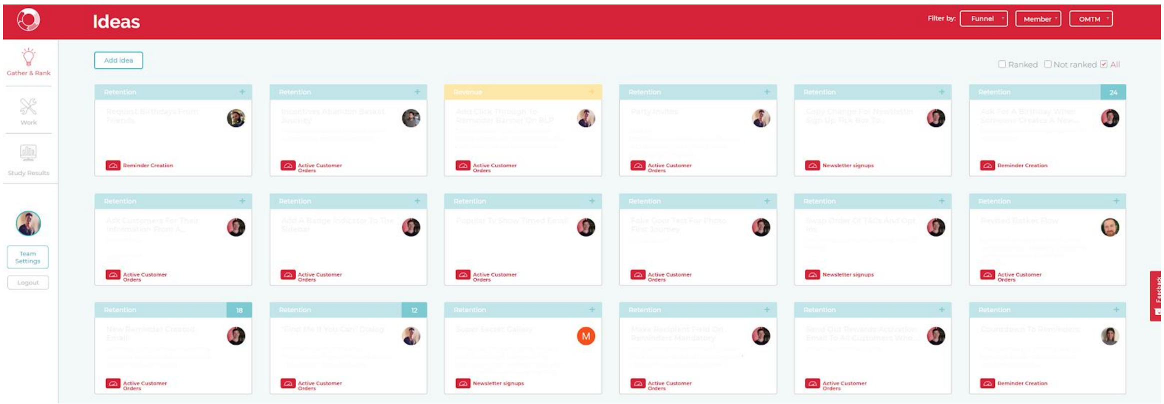
Skilling up

Within marketing the benefits of experimenting were quickly recognised, but what the teams needed was help to get started. To kick-start our lean marketing efforts we invested in some training with the excellent Growth Tribe Academy. We organised 3 days in-house training for 24 members of our growth tribe. Over the course of the training they learned how to use data to gain insights, how to gather and rank ideas, how to craft hypotheses and how to run experiments. This training provided the basic knowledge of how to operate in a lean way.

Experimenting in practice

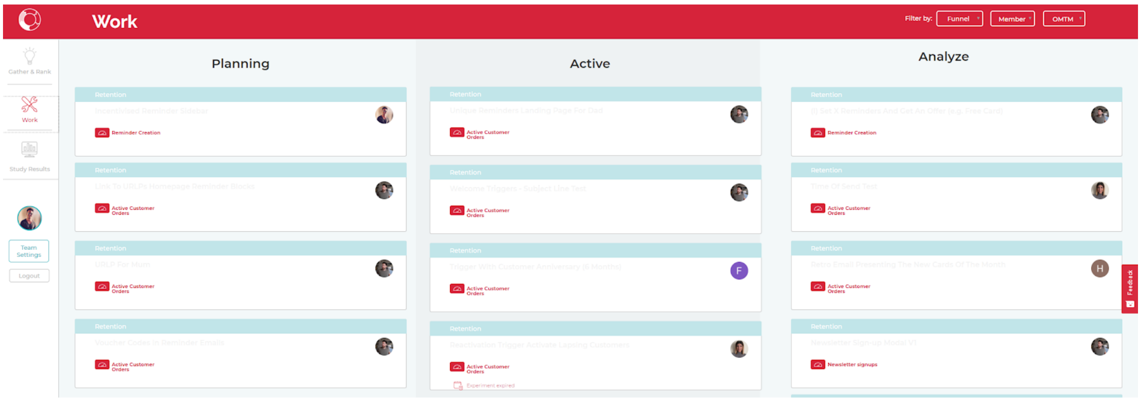
Within the Retention Pod the aspiration was to run experiments at high tempo. We adopted the growth process advocated by Sean Ellis in the excellent Hacking Growth, supplemented with some of the great tools we acquired from our Growth Tribe training.

To begin with the team needed to decide on the initial focus for experimentation. Having done this, they held a brainstorm to come up with ideas. They then ranked the ideas and kicked off the first few experiments.



Gathering and ranking ideas with Growth Tribe Academy's GrowsApp

Having got started, they were now able to adopt an on-going process. Each week the team held a growth meeting. They would review the results of the previous week's experiments, discuss learnings and insights, and pick which experiments to run next.



Planning, running and analysing experiments in the GrowsApp

In addition to the weekly growth meeting they would hold regular brainstorms to keep building a backlog of ideas.

In the case of our email marketing, we started with fairly simple experiments—testing subject line headings and email timings. Later they also started to test email content across both triggers and newsletters.

Managing experiments

One benefit of our training with Growth Tribe was access to their Grows App. This tool, developed by Growth Tribe, provides a way to gather and rank ideas, plan and visualise current and future experiments, and capture insights from past experiments. Everyone in the pod had access to it so it provided a great way to collaborate and share learnings.

Building a culture of experimentation

The pod model really helped drive the uptake of experimentation within the CRM squad, because they were able to learn from their more experienced product engineering colleagues.

Elsewhere experimentation was also being used to influence our product range development. Squads developing physical product ranges started to use data insights to influence the new ranges they developed.

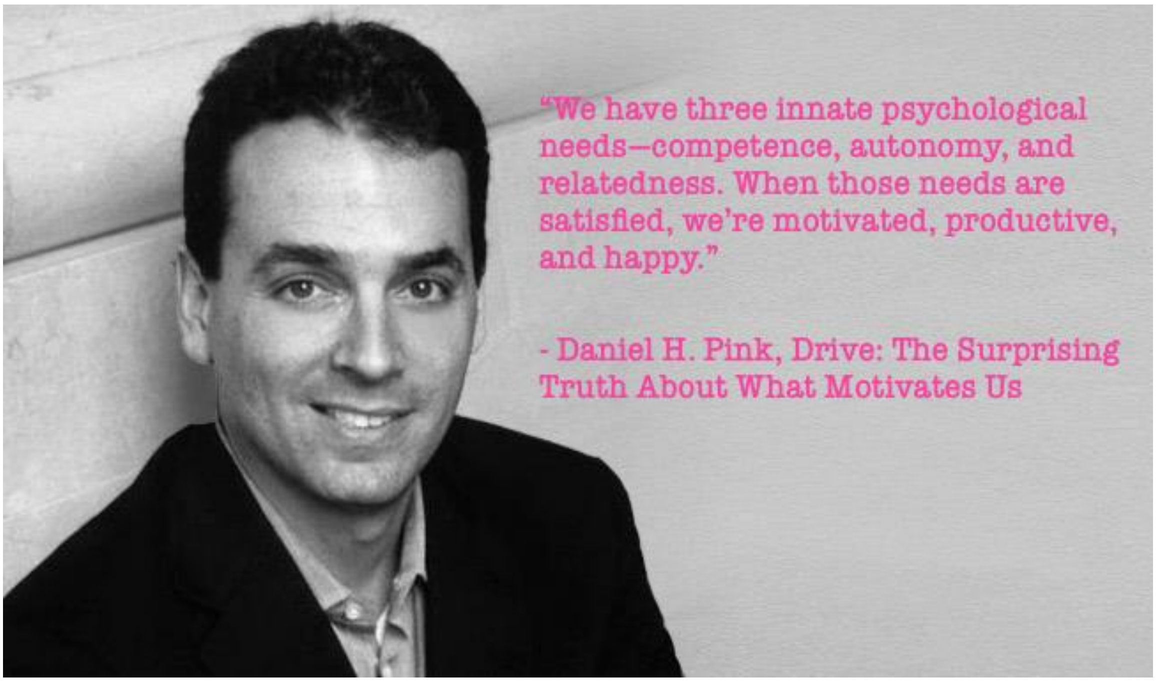
As with optimising speed, there is still plenty of room to increase experimentation, but the crucial factor is that the value of experimentation has been realised and it now happens across the organisation. It is no longer a “tech thing”!

MEASURING HAPPINESS

Thus far I’ve described how we sought to be faster and better. I’ll now focus on our third key outcome—getting happier. This was very much a collaborative effort and I was extremely fortunate in being able to work with Roopa Singh, then HR Director at Moonpig, and Lawrence Hay, Group Head of Talent for the Photobox Group (to which Moonpig belongs).

I’ll discuss firstly what we meant by being “happier” and how we planned to measure that, and I’ll then describe some of the actions we took to improve happiness.

What is happiness?


"We have three innate psychological needs—competence, autonomy, and relatedness. When those needs are satisfied, we're motivated, productive, and happy."

- Daniel H. Pink, Drive: The Surprising Truth About What Motivates Us

For us, being happier basically meant improving engagement. When it came to deciding how to improve engagement, Roopa, Lawrence and I shared the same vision. We were all very much inspired by the theories outlined in Dan Pink's *Drive*, namely that beyond a certain point extrinsic rewards no longer serve to motivate people. Instead people are motivated by purpose, mastery and autonomy. Clear purpose lets people know what is expected of them. Autonomy empowers people to use their skills to achieve that purpose—it is the antidote to micromanagement. Mastery is having the necessary skills to achieve that purpose.

We believed that by giving our squads clear goals, the autonomy to achieve them and the means for everyone to keep learning we would increase staff engagement.

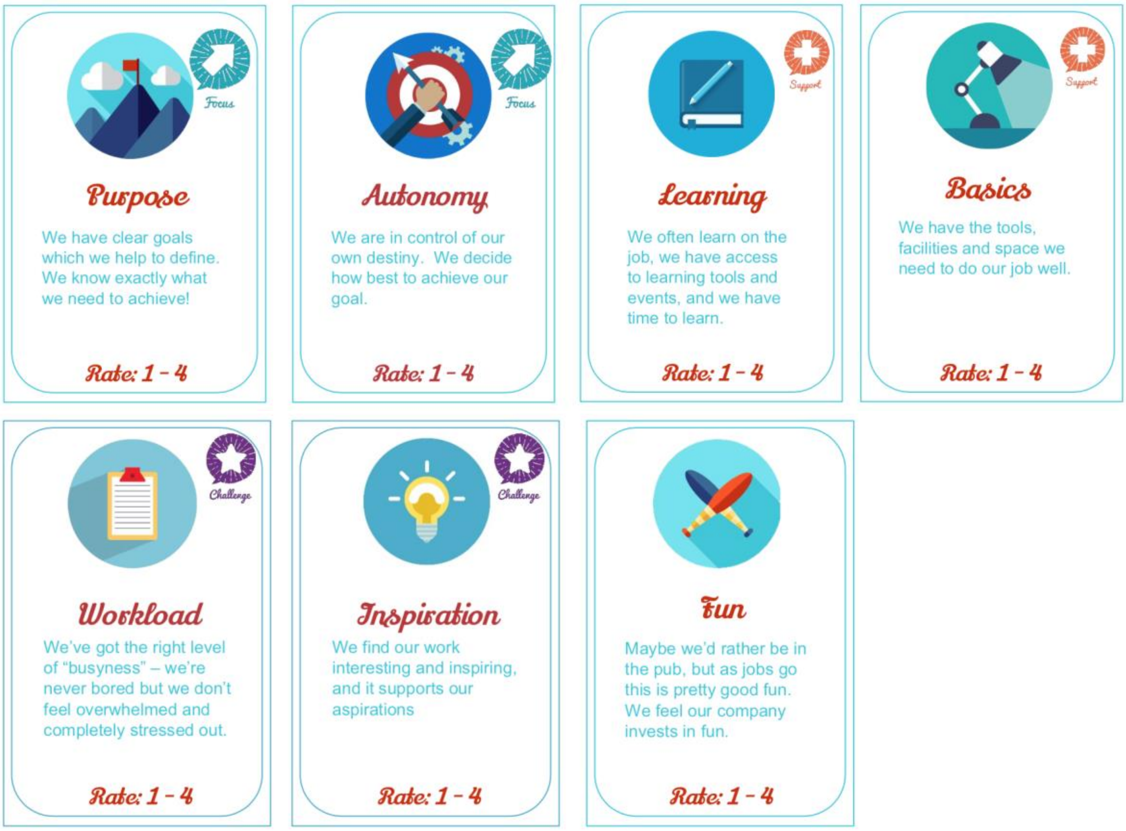
Measuring happiness

Engagement is a tricky area to measure, and to measure regularly. We had two methods. The first was the classic annual staff survey—an opportunity to gather very detailed information across the board. This always provided valuable insights, but an annual, or even 6 monthly survey, provides a very long feedback loop. We needed a way to capture actionable insights much more regularly.

"Appiness"

To that end Lawrence and I developed, "Appiness"—a Moonpig "happiness health check". This was very much inspired by Spotify's squad health check, but we modified the categories to be less software specific. The plan was to run happiness check-ups every couple of months in squad retrospectives. This would guarantee regularity and high levels of participation.

The health check focused on seven categories:



Purpose

We have clear goals which we help to define. We know exactly what we need to achieve!

Rate: 1 - 4

Autonomy

We are in control of our own destiny. We decide how best to achieve our goal.

Rate: 1 - 4

Learning

We often learn on the job, we have access to learning tools and events, and we have time to learn.

Rate: 1 - 4

Basics

We have the tools, facilities and space we need to do our job well.

Rate: 1 - 4

Workload

We've got the right level of "busyness" – we're never bored but we don't feel overwhelmed and completely stressed out.

Rate: 1 - 4

Inspiration

We find our work interesting and inspiring, and it supports our aspirations

Rate: 1 - 4

Fun

Maybe we'd rather be in the pub, but as jobs go this is pretty good fun. We feel our company invests in fun.

Rate: 1 - 4

Each category, except Fun, also aligned to the three key areas of the Be THAT Manager leadership training programme, which I'll come to later. Fun was an additional area we covered because it has always been an important element of Moonpig's culture and something people in the organisation valued.

People were asked to rate each category, and there were 4 levels of rating:

Awesome

Pretty Good

Meh...

This sucks!

Each rating had a score:
Awesome (3 points)

Pretty Good (2 points)

Meh (1 point)

This sucks (0 points)

By obtaining a rating from each person for each category we were able to extrapolate a score and work out as a percentage how each squad rated the individual categories.

In addition to gathering this quantitative data, we were able to discuss low ranking categories with the squads there and then. This provided qualitative insights that helped us understand what was causing the problems, which in turn helped identify actions to take.

You can download a copy of the categories and ratings cards here.

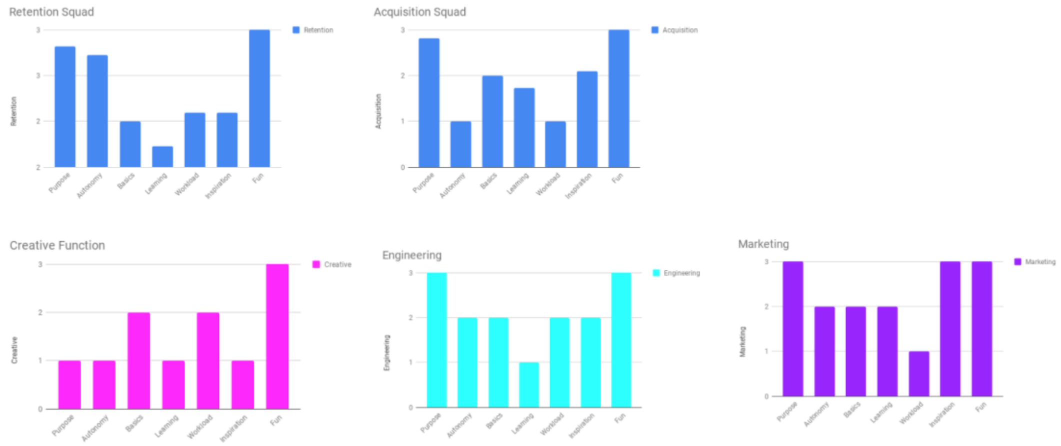
Scaling Appiness

To prove the concept, we ran a manual version of this with a few of the squads. We printed and laminated copies of the ratings cards and people would hold up the relevant card as they rated each category. I would then manually calculate scores and percentages for each category and translate that in to a graph. This was time consuming, but as a proof of concept it demonstrated the value it could provide.

However, in order to regularly measure and generate useful data from these checks, we needed a digital solution. Tech resource being scarce, we used our annual hackathon to build a digital version. This would enable people to rate categories directly through a mobile device, and we could then automatically generate scores.

We also wanted to supplement the app with a dashboard to display the data so we could see the latest scores, but also see how scores changed over a period of time—were we getting happier or not?

In addition we wanted to build in the ability to view results by both squad and function meaning we could identify trends at different levels—squad X doesn’t feel they have autonomy, function Y ranks learning very low etc.



This is mock-up shows how we wanted to be able to view the results—the data is not real.

Acting on the data

Whilst the awesome hackathon team completed the app during the hackathon, building the data dashboard proved too much to accomplish in 24 hours! There is still the aspiration to complete the dashboard, but it may take a while.

However, it’s worth outlining how I planned to make use of that data once we had the means to capture and display it. Ideally I would liked to have run a health check every two months to begin with—we could continue to review cadence. Once complete I wanted to go through the results with the leadership, talking through both the quantitative data and sharing the qualitative feedback.

The next step would be to work with the leadership to define clear actions to address the problems. Thereafter both the results and the actions would be shared with the entire organisation. Finally I wanted the leadership team to provide progress updates against those actions on a weekly basis—Moonpig holds weekly all-hands, which would provide an ideal time for this.

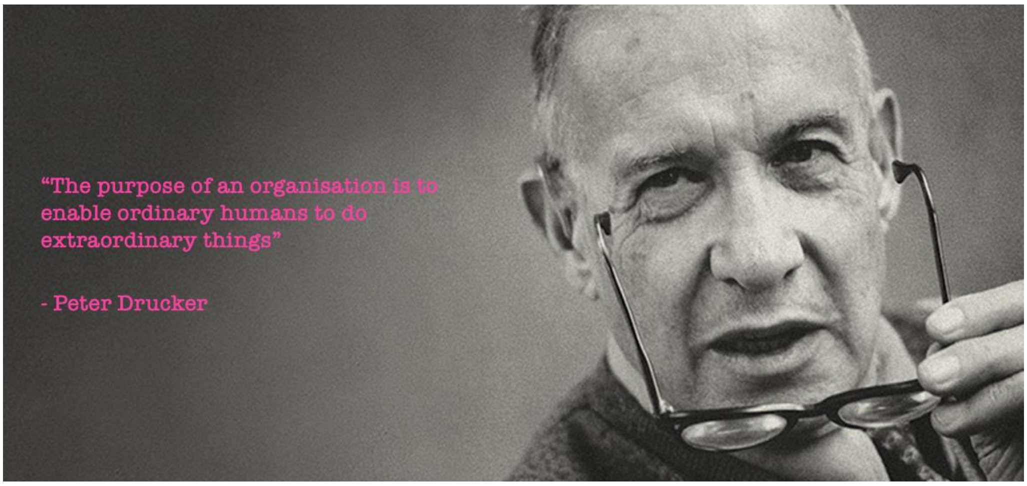
Translating real problems from anecdotal feedback in to hard data helps encourage leaders to take problems seriously. But for leaders, acknowledging the problems is not enough—they have to be seen to be acting on them. If people are to trust and believe in their leaders, they need to see that there is real commitment to their engagement and well-being.

GETTING HAPPIER

As I mentioned earlier, I was extremely fortunate in being able to collaborate with HR Director, Roopa Singh, and Group Head of Talent, Lawrence Hay. Many of the following initiatives were lead by Roopa and Lawrence and thus this provided one of the biggest areas of learning for me personally.

I discovered a huge overlap between the agile, HR and L&D functions, because fundamentally we are all working to improve the culture, and culture is very broad. Provided you share the same values and vision, which we certainly did, this collaboration can be very powerful and very rewarding.

Squad principles and happiness


"The purpose of an organisation is to enable ordinary humans to do extraordinary things"
- Peter Drucker

Earlier I described two of the key principles on which the squads were based. Firstly, that each squad should have a clear mission, and secondly that each squad should have the autonomy to achieve that goal independently. Thus the squads were very much based aligned with Dan Pink's theories around motivation.

In addition to these key principles, leveraging lean and agile working practices, which were introduced to increase speed, were also intended to promote a sustainable pace of work and better work life balance. As well as improved quality of work experience, it was also hoped that sustainable pace would provide people with more time to learn.
So there was an expectation that by working within this model we would naturally start to see improvements in engagement. However, over and above this, there were also several broader programmes launched to improve engagement. These are all in the early stages so I can't report on any outcomes, but it's worth sharing the ideas.

Values

Developing a healthy culture is intrinsically linked to engagement, so it’s not surprising that this was a key area of focus. Roopa and Lawrence began by running listening sessions, involving people across the organisation. The objective of these sessions was to understand both the healthy and the less healthy elements of our culture. From that we could begin to determine a set of company values that were widely supported and could drive behaviours that would improve the culture. An inclusive approach to developing the values meant they were more likely to be supported and embraced by the organisation.

In addition to the listening sessions, Roopa and Lawrence asked for voluntary culture champions, thereby creating a network of people that could promote our values and culture across the business.

Having run the listening sessions and started developing the values, a company offsite was used to involve the wider organisation in identifying specific initiatives that we could run to embed the culture and values. The culture champions ran workshops with groups of people across the organisation to brainstorm ideas that they could then take back and develop.

Leadership

Leadership drives culture, so focusing on developing good leaders is vital to developing a healthy culture.

To focus on improving leadership Lawrence and his team developed *Be THAT Manager*—a training programme that anyone involved in line management or squad leadership would attend.

As I said earlier, Lawrence firmly believed in Dan Pink's theory of purpose, mastery and autonomy being key motivators, and he sought to embed these in the training programme. The programme itself was built around the three key themes of Focus, Support and Challenge:

- **Focus**—providing inspiring vision and clear goals
- **Support**—adopting the coaching style of leadership which encourages managers to help others solve problems rather providing directive instruction
- **Challenge**— this focused on helping to develop and build people by encouraging them to push themselves beyond their comfort zone without over stretching them

The leadership programme was designed to develop great leaders that would excel at developing high performers while creating an environment where people remained highly motivated and engaged.

Career progression

One long running complaint we had was lack of career progression. This had been particularly problematic in technology but was also recognised as a problem elsewhere.

To address this we developed a competency matrix against which we could align roles and provide clear guidance for individuals to develop and progress. Whilst this was developed as an initial pilot within technology, it was written with a view to extending it across the organisation.

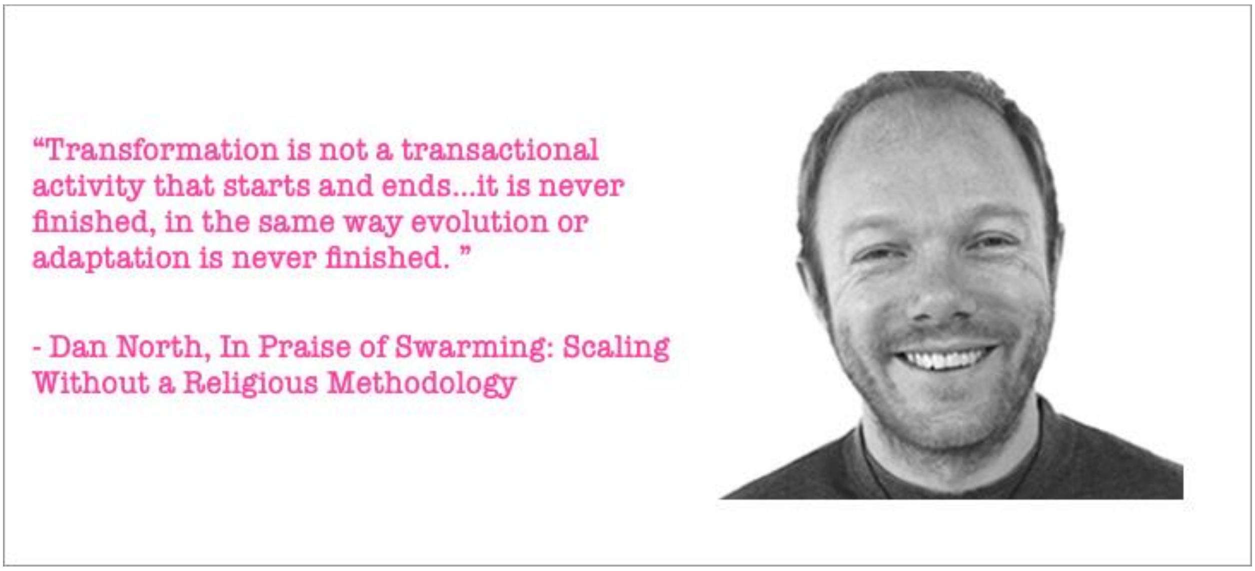
We focused on 12 competencies, 3 of which were specific to engineering. The others covered more general areas such as leadership skills, soft skills and ways of working.

The competencies very much reflected the values and behaviours we wanted to encourage. The leadership competencies aligned with the themes of our leadership training programme, and were based around the themes of focus, support and challenge. Likewise ways of working encouraged lean and agile working practices and softer skills emphasised areas such as communication and collaboration.

As well as providing a framework for career progression the competency matrix was also intended to support recruitment, helping us to acquire talent that would contribute to a healthy culture.

OUTCOMES & LESSONS LEARNED

Having attempted to tell the story of how Moonpig approached adopting business agility I'll now share some of the results—some of the benefits that we saw. I'll also gather together the key lessons I learned, and some of the considerations for anyone else that might be tempted to adopt business agility.



"Transformation is not a transactional activity that starts and ends...it is never finished, in the same way evolution or adaptation is never finished. "

- Dan North, In Praise of Swarming: Scaling Without a Religious Methodology

Outcomes

Before I describe some of the outcomes, it's worth putting them in context. The changes I've outlined took place over 6–8 months, so we are still at a very early stage and we have to consider the results in light of that. You might think of what we've done so far as an "MVP". If you consider the changes we made as a set of hypotheses against which we can measure success, what we're looking for is positive signals that we are moving in the right direction and that we should continue along this path.

When outlining our vision and measures of success, I described how we hoped to achieve 3 key outcomes:

- **Better** = better outcomes, increased ROI
- **Faster** = reduced cycle time across all value streams
- **Happier** = higher employee engagement

So did the changes we made positively affect those outcomes?

Did we get better?

To understand this I looked at improvements on ROI. I can't reveal actual numbers, but the squads delivered very healthy incremental revenue growth during the first 6 months. A less scientific, but no less revealing, indicator was that the leadership team were extremely pleased with the results being generated by the squads!

At this early stage I'd judge the gains as modest but very promising. With continued focus on experimentation I'd expect to see these gains continue to increase.

Did we get faster?

In terms of speed we saw some more dramatic gains. Realigning the teams improved cycle times dramatically in some areas, particularly in delivery of our marketing content where we saw cycle times reduced from months to days.

Again there is still a very long way to go and plenty of opportunity to continue to optimise workflows to further increase speed.

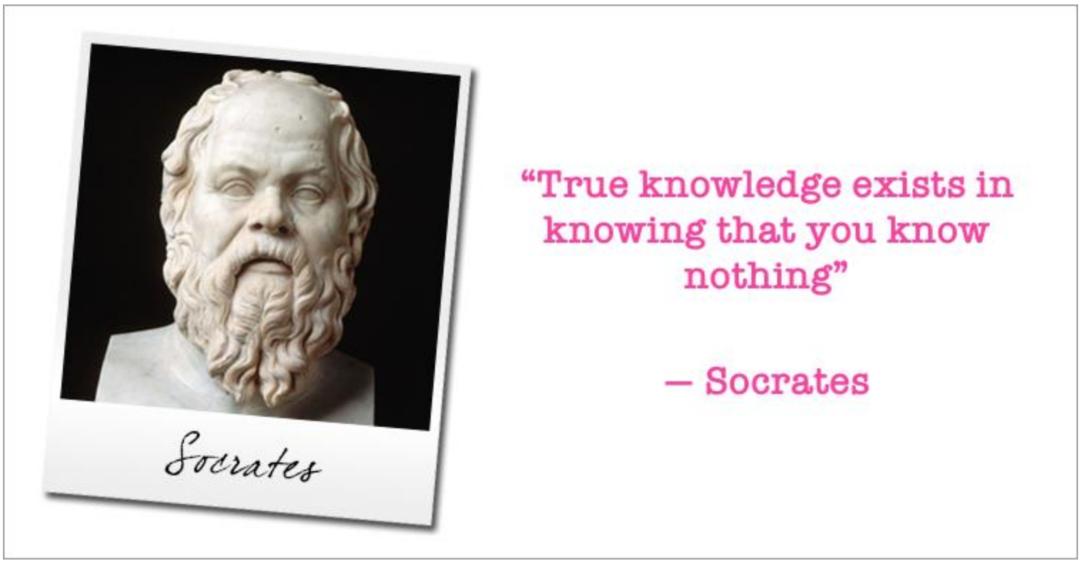
Did we get happier?

This was particularly difficult to measure, as we had no real baseline. However, I was able to extrapolate some results by comparing the results of annual staff surveys from before and after the change. These saw some marked improvements in areas such as alignment & involvement (+13%) and enablement (+21%).

There is still tremendous improvement to be made in all of these areas. This is not the end of the journey, or even the beginning of the end of the journey. Adopting a continuous improvement mindset pre-supposes that you never reach a state of perfection, but instead seek to constantly optimise and improve. As you make changes you reflect on whether they have had the desired improvement and you continue to inspect and adapt. However, the results thus far certainly gave me confidence that the changes we made were having a positive impact.

Beyond the quantitative measures I also had the benefit of adhoc qualitative feedback from the squads themselves. My impression from this was that, whilst there was acknowledgement that there was still a long way to go, people definitely seemed to feel there was an improvement. Some had benefited more than others and that was reflected in the feedback I got.

Lessons learned



"True knowledge exists in knowing that you know nothing"

— Socrates

My personal agile journey with Moonpig has now come to an end, but the experience has been invaluable. I think it would be hard to capture everything I've learned in the last few years, but here is some of the key advice I would offer based on my experiences:

1. **You need long term executive sponsorship**

It's widely accepted that executive sponsorship is vital for agile adoption, and my experience bears this out. You cannot make changes on the scale that we did without leadership support. However, you also need consistent, long-term support—a change in leadership can mean a change in strategy and approach, which can quickly derail your plans.

2. Start by coaching leadership

Executive sponsorship is great, but it's important that those executives know exactly what it is they are sponsoring! I described at the beginning of this article my vision and my "roadmap". In hindsight I wish I'd shared those with the leadership team and spent more time helping them understand my plans in detail and the principles behind them. Adopting business agility fundamentally changes the way you operate—you may alter your teams, the role of leadership is very different, working practices are different and the culture and behaviours you seek to drive are different.

These are wholesale changes and it's important that your leadership grasp what you plan to do and why, and what their role will be in supporting a successful transformation. There can be an expectation that everyone other than leadership needs to change—but arguably the greatest change required is actually with leadership.

3. Define your vision and outcomes

As I mentioned above, have a clear vision of what you're trying to achieve and define your measures of success. If you can, try and get some baseline measures so you can understand if you're improving. Successful transformation comes from experimenting, and as with all experiments you need to measure impact. Not only will this give you and the organisation confidence in what's working and what isn't, it will help keep momentum and support behind positive changes.

4. Collaborate

Successful organisational change is not the work of a single coaching function. You will need to collaborate with leadership to ensure they emulate the culture and model the behaviours you are trying to drive. HR and L&D will also be critical to your success and building strong partnerships with them will be invaluable in helping you succeed.

5. Have the right coaching resource

One of my biggest challenges was a lack of coaching support. I started hiring additional coaches too late, and struggled to find people with the right ideas. For much of the time this meant I was the only person available to support squads as they adopted new working practices.
With more coaching resource dedicated to each squad we could have made much faster progress both with addressing problems and with improving working practices and increasing experimentation which in turn could have lead to better outcomes sooner. Essentially my work in progress limit was exceeded and that impacted the speed at which I delivered improvements!

6. Be prepared to invest

The companies I read about that have had success in adopting agility don't treat it as a nice-to-have—they treat it as a strategic decision. It's a hard-headed business choice and they invest in it accordingly. This doesn't necessarily mean huge amounts of investment but there may well be areas where you do need to spend. It could be coaching resource, training and learning programmes or tooling. Agility can deliver huge benefits to your business; it’s worth investing in.

7. Preach the value of focus

I think the single biggest problem most companies have is they try to do too much. It’s a natural temptation to try and do everything—the expectation is always that the sooner you start something the sooner you will finish it. In fact the opposite happens—the more you take on, the thinner you spread your teams and the longer it takes them to accomplish anything.

It takes real discipline to prioritise and focus but those that manage this will feel the benefits. We achieved more focus by realigning our teams around goals, but we were still trying to do too much. Communicating the value of focus should form a key part of your leadership coaching.

Conclusion

My exploration of business agility began two years ago. Back then, I wasn’t even conscious of the term “business agility”—I was simply exploring the possibility of adopting agile practices and processes in different contexts. Fast-forward to November 2018 and I now cannot see past business agility! My experiences in the last few years have taught me that it has enormous potential, and anything less now feels sub-optimal.

As long as agile remains a “tech thing”, we consign ourselves to optimising a single corner of an organisation. Lean and agile comprise a set of principles which are ultimately agnostic of technology—every part of the organisation can benefit.

There is no one-size-fits-all. The basic principles may apply to all, but designing a better system of work for your organisation means experimenting to create a tailor-made system that works for you. Attempting to copy Moonpig’s approach won’t guarantee success any more than copying Spotify’s. I hope you have learned from what we tried at Moonpig, and you can take those learnings and apply them to fit your own business model, strategy, size and culture.

It’s a long journey, but it is incredibly rewarding. Start small, but start now.

About the Author
Amanda is a freelance coach and consultant helping organisations to adopt lean, agile and growth practices, enabling them to grow, innovate and deliver value at scale.