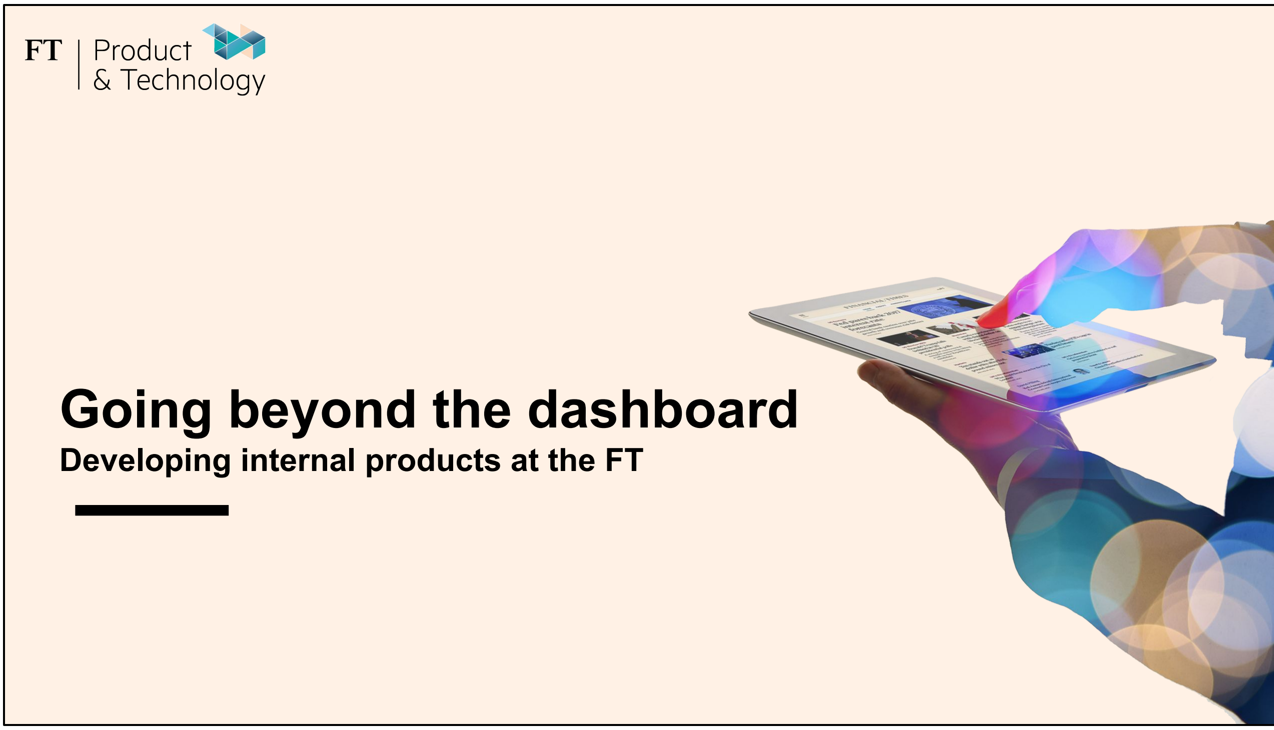


Good morning everybody and thank you for coming

My name is David and I am Head of Internal Products at the FT. This is Matt (Hello) who is the Technical Director for internal products.

At the FT Internal Products refers to any product that is used by an internal FT customer. This might be our editorial tools are employee apps or the tools that our sales and marketing teams use to attract, engage or retain our customers.

Our talk today is going to be about how we approach the development of these products and the journey we are going on.

[SLIDE]

And we are going to break it down into three sections

1. How we approach the development of Ft.com - our flagship digital product

2. Contrast this with how we have historically tackled the development of our internal products and the journey we are on to change that

3. And finally share with you an example of how we are putting this into practice and what we have learnt'

4. But first - a little bit of background about the FT

The Financial Times is one of the world's leading news organisations,.
Founded in 1888
Now part of Nikkei Inc
Global with over 2200 staff worldwide
Ft Content is read by 2m everyday

Probably still best known as a newspaper

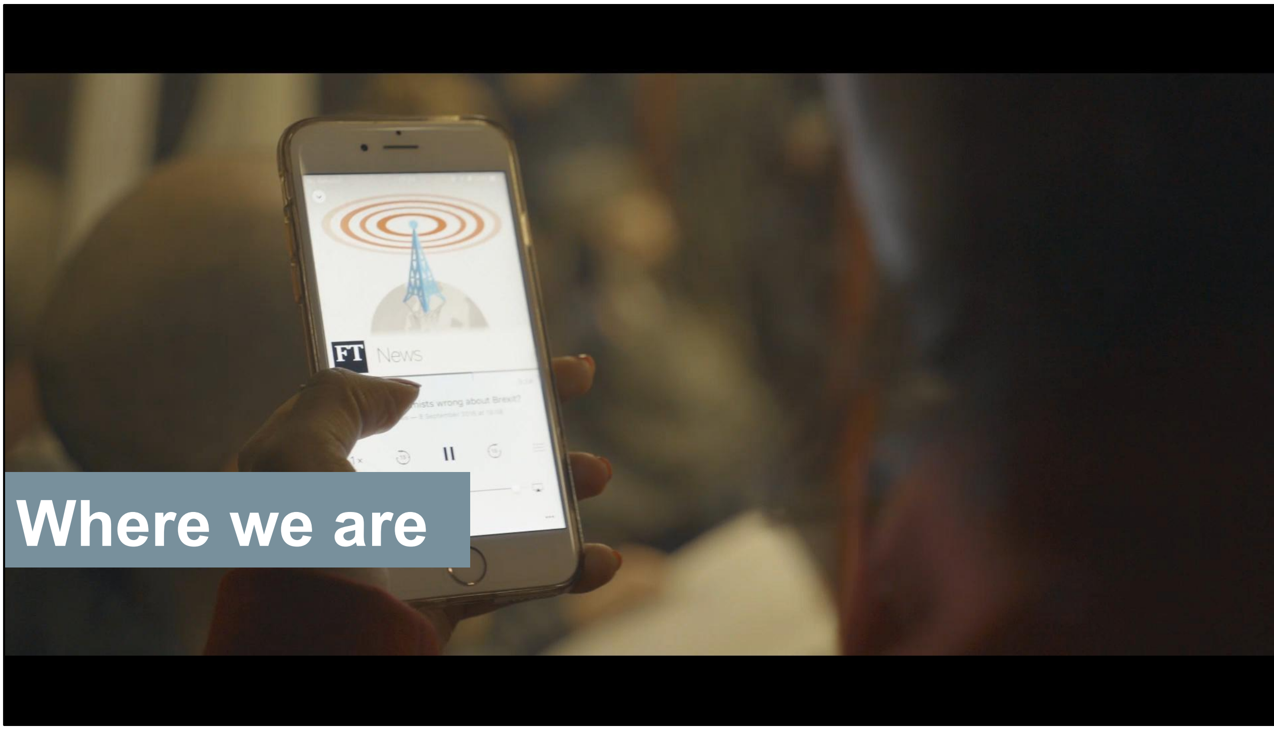
Our heritage and it is in our DNA . It is also a success product in it's own right with a global circulation of 200k subscribers .

But it is no longer our core product, like every other media organisation we have been through period of rapid and significant change.

Just 10 years ago - the typical FT reader would get their news in print - just like above.

5 years ago it may have been on a computer,

and now a typical ft reader might be consuming our content like this.. .



An FT podcast on a mobile device..

And perhaps in another 5 years time it might look something like this..

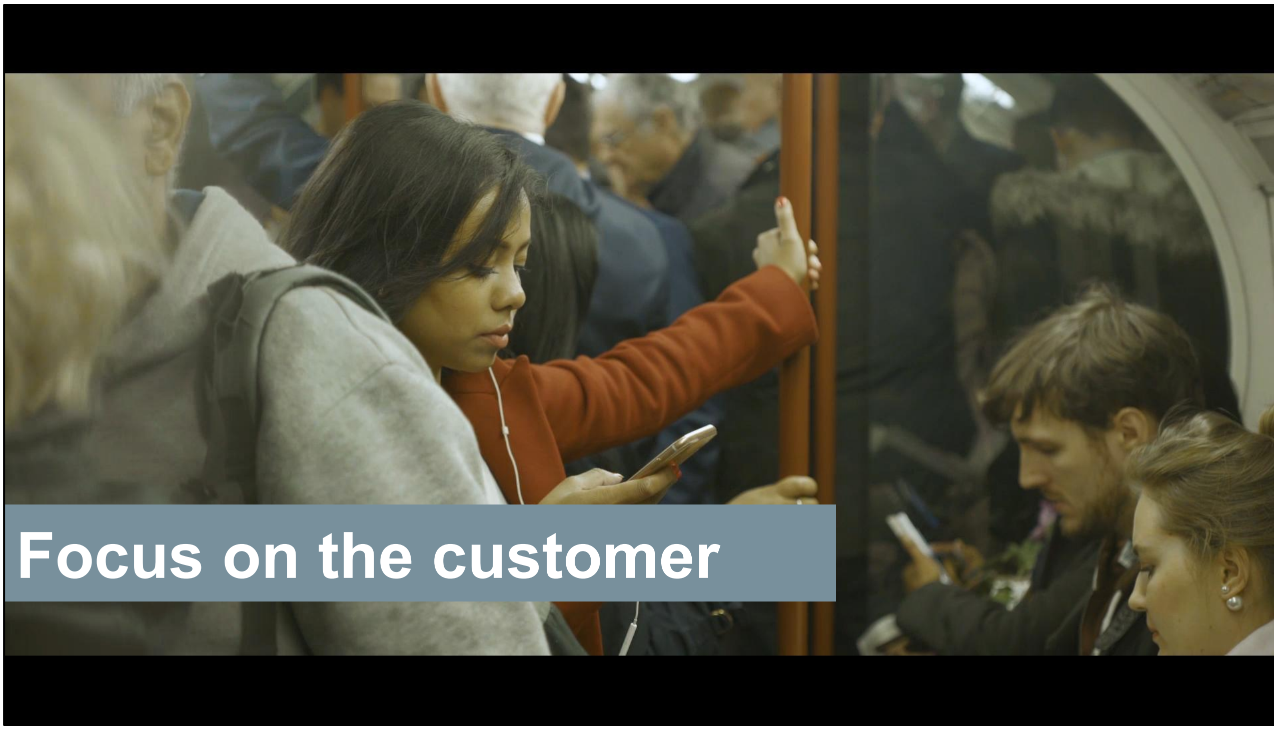
Or maybe not .

But there is a serious point here.

The pace of change shows no sign of abating . New forms, and new mediums will disrupt even further

Adapting to this new world isn't just about launching new products, it is about changing our approach to their **how** we develop them that take into account the changing consumption habits, competitive threats and the technology landscape

It's what we call a product centric approach, it is how we develop our customer facing products and we want to talk about some three key aspects



The first of these is a relentless focus on the needs of our customers

It’s somewhat of a truism - every business says they do this.

But I think that we do this well

We have a dedicated customer research team who sit within the Product function and they work with us to organise

We run weekly customer councils - bringing in customers and non-customers into our offices to share their feedback

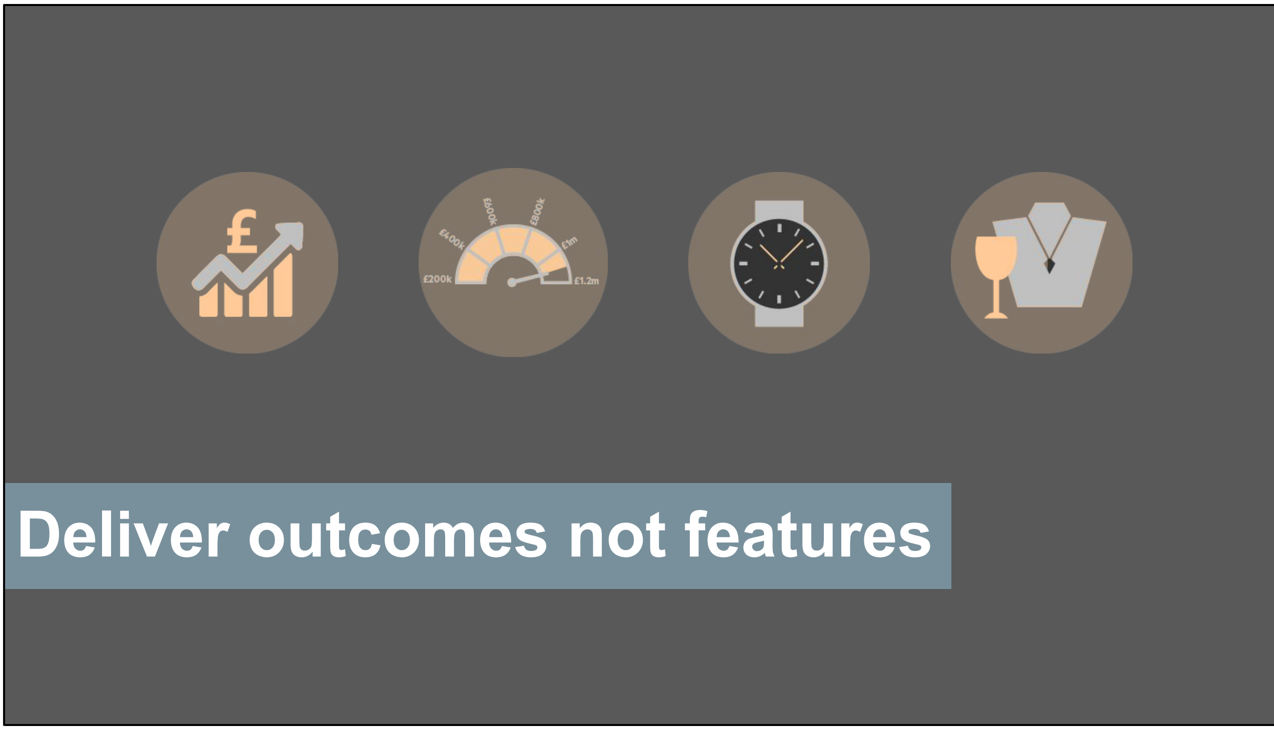
We run Gorilla user testing - camping out in coffee shops in the city to test new products or ellicit feedback

AND We run lots of qualitative surveys - NPS , Customer Service Satisfaction - On site feedback

Now, This isn’t about listening necessarily to what a customer wants

it is about listening to their problems, pain points, frustrations and goals.

And using this information to identify opportunities or problems that we can help solve for them



The second principle is around the mandate we give to our product development teams

Historically we used to ask teams to build features, or gave them “requirements” that were really solutions or details of what to build

Teams would then go away, build something and give it back.

And guess what, it never really achieved what was expected.

And we had our software engineers hated their jobs

And we wasted time, energy and money

So now, We banned (or discouraged) the term “REQUIREMENTS”

We give our teams outcomes that we want to achieve, engagement , acquisition, retention, speed. And we ask the teams to work in partnership with the business to solve problems

And we do it iteratively, we build, we measure, and we build again

And to do this ...

	Area	Metric	Benchmark	Goal	Change	Current State vs Goal
1	Engagement	**# of engaged users**	xxx,xxx	**xxx,xxx**	+12%	xxxxxx
2	Conversion	**# B2C Net subs** **Trials 17w survival rate**	xxx,xx 30.7%	**xxx,xx** **32.2%**	+8.8% +4.9%	xxxxxxx
3		**# B2B sales ready leads** **£ Trial opps closed**	xxxx £xxx	**xxxx** **£xxx**	+22% +37%	xxxxxxx
4	NPS	**NPS**	41	**47**	+7 points	40, Weighted '17 so far
5	Ads	**Viewability**	Desktop: xx% Mobile: xx%	**xx%**	+7%	xxxxxxxx

Measure everything

We measure everything .

On screen are our top level metrics for Ft.com

They are used by the product development team to track and report on their success.

They are available on demand to everyone across the business

And this is just one example

We have over 850 dashboards used by all teams at the FT

We have a data warehouse with several billion rows of usage data,

We A/B test all new functionality released on our sites, we measure performance and if successful roll out to our wider customers

So, alongside our customer research, this qualitative data is at the centre of our product development

Internal Products

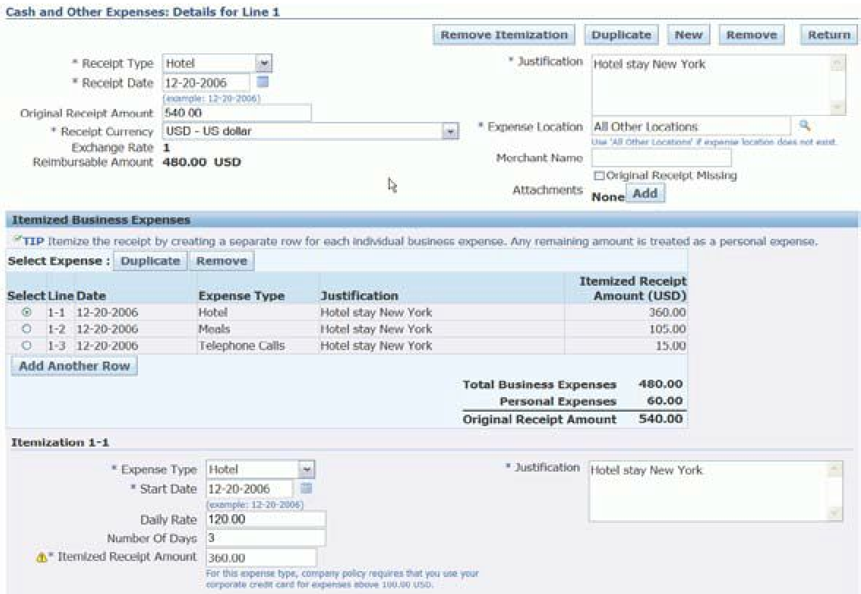
The principles that we apply to our customer facing products are equally applicable to the tools and software that our staff use.

This means our CRM systems, marketing tools, data products, the tools our journalists use to tell their stories.

Each of these has customers - (internal ones not external) - they all seek to achieve outcomes, they all exist a product lifecycle.

If the approach to FT.com feels relatively modern, lean… the approach over the last decade to our internal tools might be described as the opposite.

For the most part they've been neglected - not particularly though any malice, but they've typically had no overall direction.



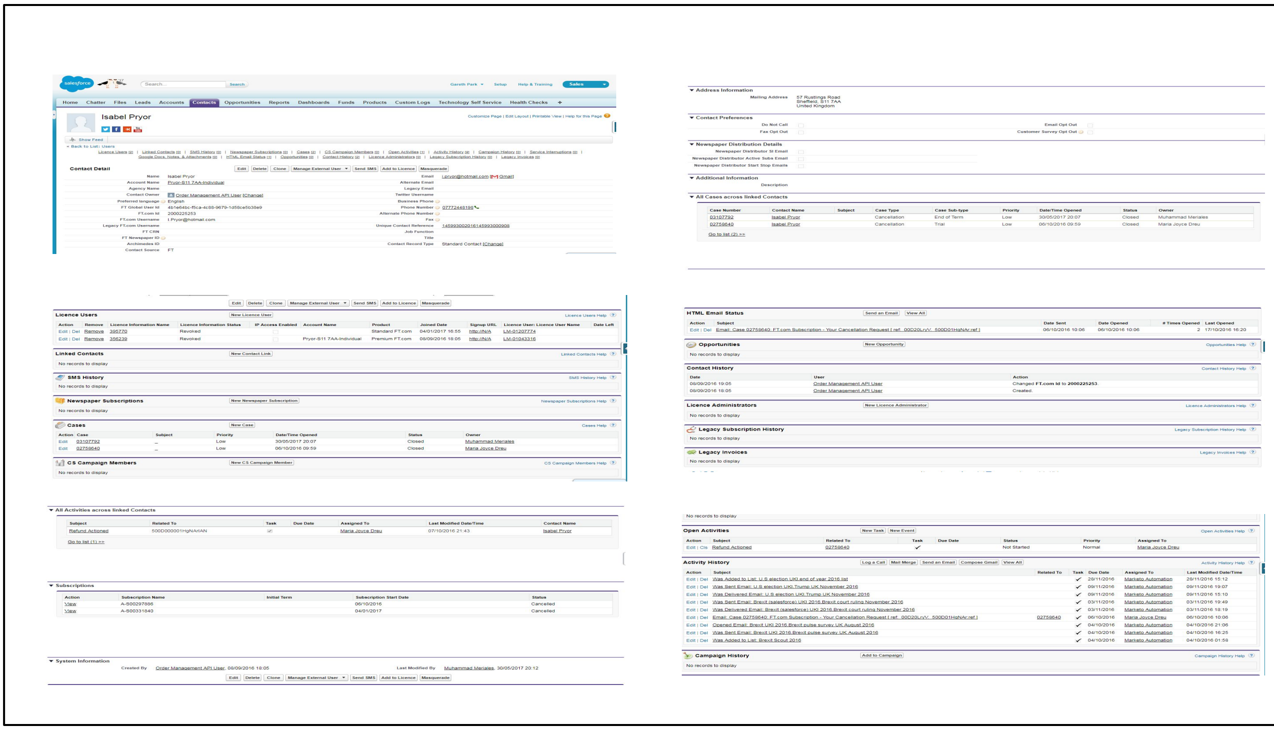
For example, this is our expenses system. Unless you are lucky, then your expenses system looks something like this too.

The form fields are scattered across the screen.

There's 11 shades of grey & blue, 8 different typeface styles.

When you type in 40 dollars it helpfully converts it to Sterling, but at a conversion rate that's a usually few months old, so you have to manually correct it yourself, unless it's in your favour.

The user experience is predominantly one of hacking your way through a thick forest to get to the end.



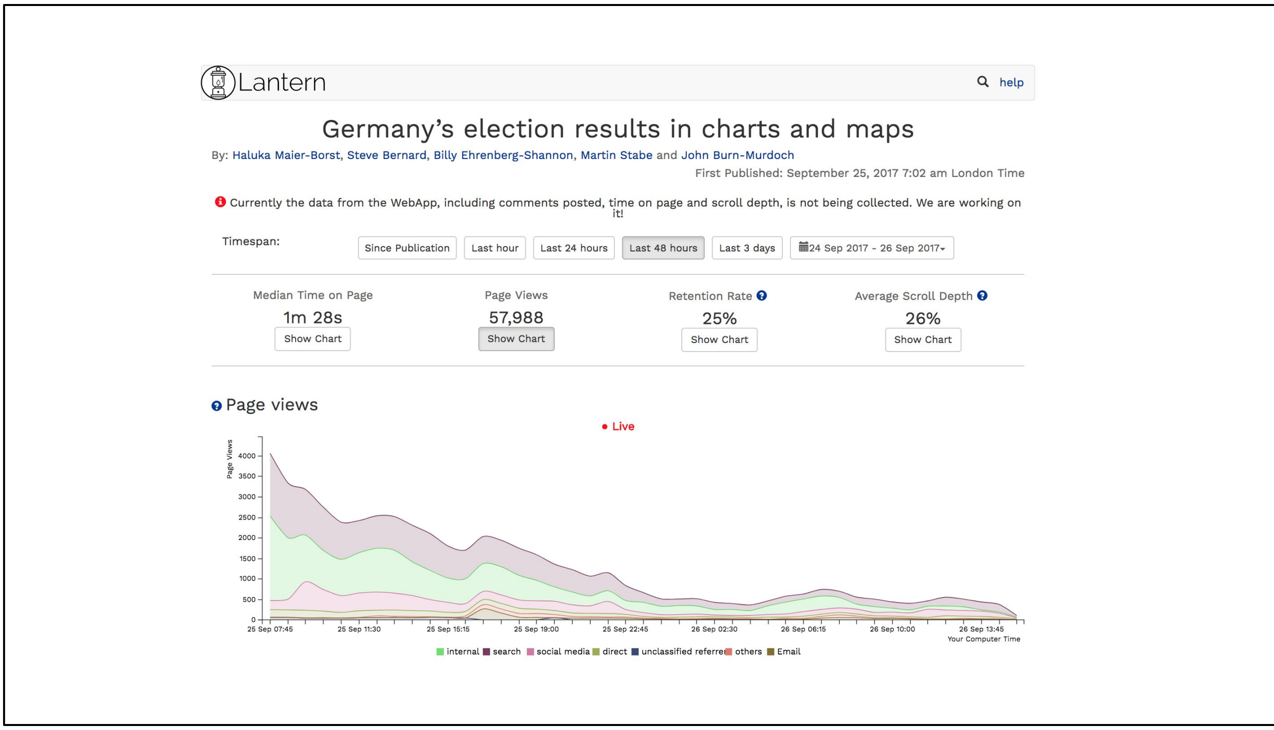
Sometimes these things aren't fun to use, and a little pain is perhaps inevitable and bearable.

But sometimes you feel like they are intentional acts of sabotage.

Here's our previous version of the customer service dashboards - top to bottom it just about fits on five computer screens.

When you phone the FT to speak to one of our customer service people on the phone, the phone system automatically opens up the customer's record on the screen so they can see who they are talking to.

So any conversation that they are having is hampered by having to scamper around trying to locate the information the customer wants or the series of actions to upgrade, downgrade the account or log a problem with the support team.



But it's not universally so.

This is a tool called Lantern - an internally developed tool to let to newsroom view the live behavioral data from FT.com.

It's a tool built by a product team - developed iteratively, over a number of months and years - there's a dedicated team that looks after it.

The information is presented clearly, detailed breakdowns are hidden beneath an intentionally empty looking surface, the data warehouse that powers it has been flexible to adapt over many iterations to lots of requests for new features, nobody really has to be trained to use it.

Well built tools are a discipline.

Step 1: Oversight

So in order to transform ourselves into a team that makes more useful tools we approached the problem in four different ways.

The first change, is that the situation we are in, as I said, isn't through malice rather lack of strategic oversight of our tools.

So in forming a single Internal Products team we can oversee the direction of everything - the cost of ownership and the technical platform and the approach to how we buy and build our software.

Traditionally, where budgets and decisions are devolved to lots of differents teams it's been hard to build a coherent ecosystem of tools, where each new feature or each investment builds on top of the previous ones.

Contrast this with many customer facing products. FT.com for example is currently around 8 product teams and 60 separate underlying components & services built by 100 engineers across different departments, but it's all orchestrated into to one seamless experience for the customer.

One person essentially has direction of what the customer sees, which leads to coherence.

Step 2: Metrics

As David alluded to earlier the second change we've made is to introduce is a focus on metrics.

In some ways this is harder than larger customer-facing projects I've worked on.

For example, you can't A/B test a product that only has five people using it. The volume of behavioural data isn't going to exist to be meaningful.

But conversely you can walk across the room and watch people do their jobs & talk to them, ask them about what they think.

You can conduct simple Time & Motion studies to look at efficiency and effectiveness.

So many of our metrics are around satisfaction and efficiency.

And ultimately investment in these things can help us meet other goals by freeing up people's time.

Step 3: People

Next we looked at the people.

In our view a more rounded team results in a better product.

At the start of 2017 we had a team of 40 software engineers, a collection of business analysts and project managers and QA.

We lacked UX people to help research and represent the voice of the user.

We lacked Product Managers to help us find, and then prioritise the most valuable problems that needed to be solved.

And we lacked software developers who were as interested in finessing the details of a user-interface as they were finessing, say, the performance of our databases.

Step 4: Process

These three new roles put a whole new emphasis on the way in which our internal tools are built.

We changed from a very traditional, subservient, requirements collection and delivery cycle, to a more exploratory approach.

We now focus on the speed to (our internal) market.

We focus on creating feedback loops.

We focus on figuring out where the non-obvious problems are to be solved.

It’s a much more inquisitive way to build things.

Much of the challenge we have re-educating our teams & stakeholders to work in this different way.

David is going to talk you through the recent project to redevelop of one of our b2b sales and marketing tools.

Putting it into practice

As Matt said, the story I am going to tell hopefully illustrates some of these principles in practice.

A bit of background first
Our corporate subscriptions (or B2b) business is one of the fastest growing parts of the FT
It is also one of the most innovative - it has a usage based pricing model where organisations only paid for engaged readers
And as such the B2b team, relies heavily on data and in particular a single tool - to help set pricing, drive usage and identify 'at risk' or ' upsell' customers.
This tool, essentially the Backbone of the business, is almost 7 years old, and has reached the end of it's life.
It needed to be replaced…
But rather than a looking for a 'like for like' swap, we saw it as an opportunity to re-imagine how the b2b team might use data more effectively

So In the same way we approach the development of our customer products we started with **internal** customer needs.

Our user experience team sat with end users - Sales, Marketing , Operations and observed what they did, how they did it and why

We then mapped their workflow end to end and between multiple tools and applications

And we used this to build personas just like the one shown here.

For those that don't know - a persona helps us understand and communicate holistic customer needs through building up a picture of their needs, goals, motivations and pain points.

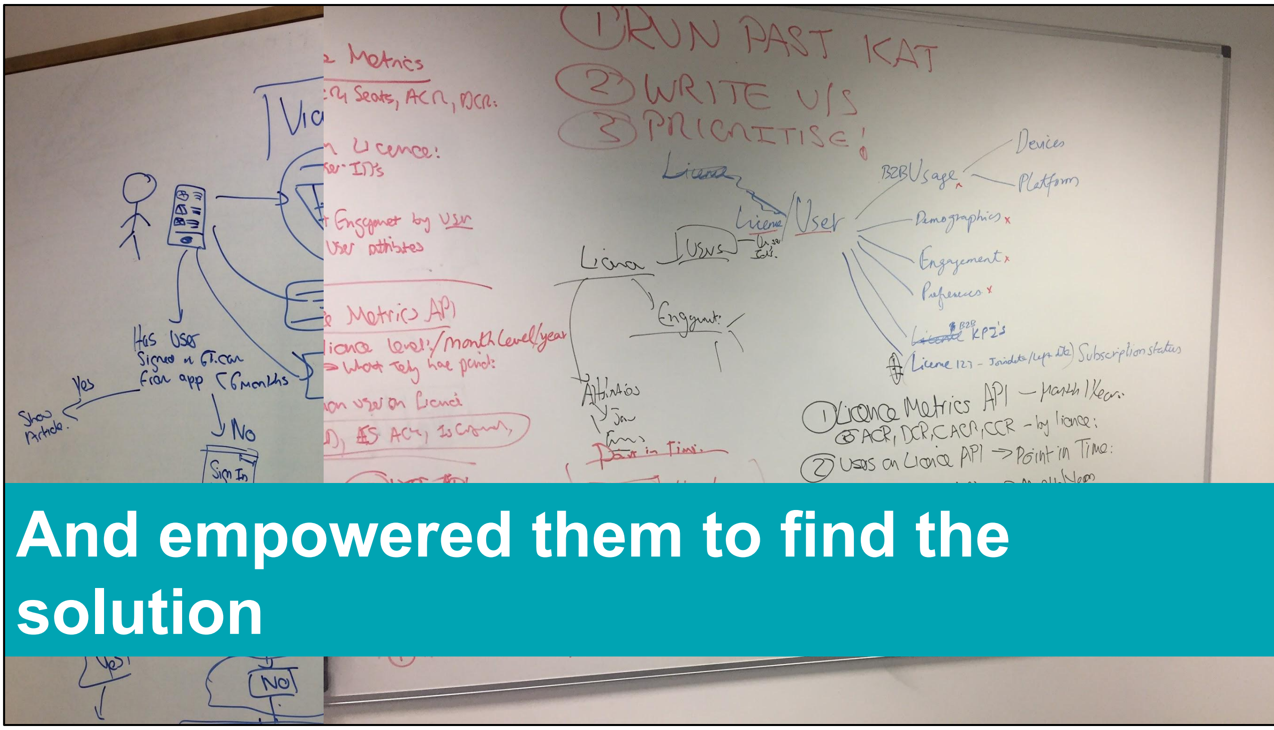
We use it through the development process as a core reference point - "Will this feature / capability help this customer achieve their goals?"

And we used this customer research, together with overall business goals, to set success metrics for the team

Crucially we look for metrics that the team can influence but that can be aligned back to overall business outcomes

For example one of the metrics is reducing the steps taken to retrieve key data points

Working with our senior mngt team we identified that reducing these steps would allow sales team to operate more efficiently, increasing the number of calls made and hence new revenue



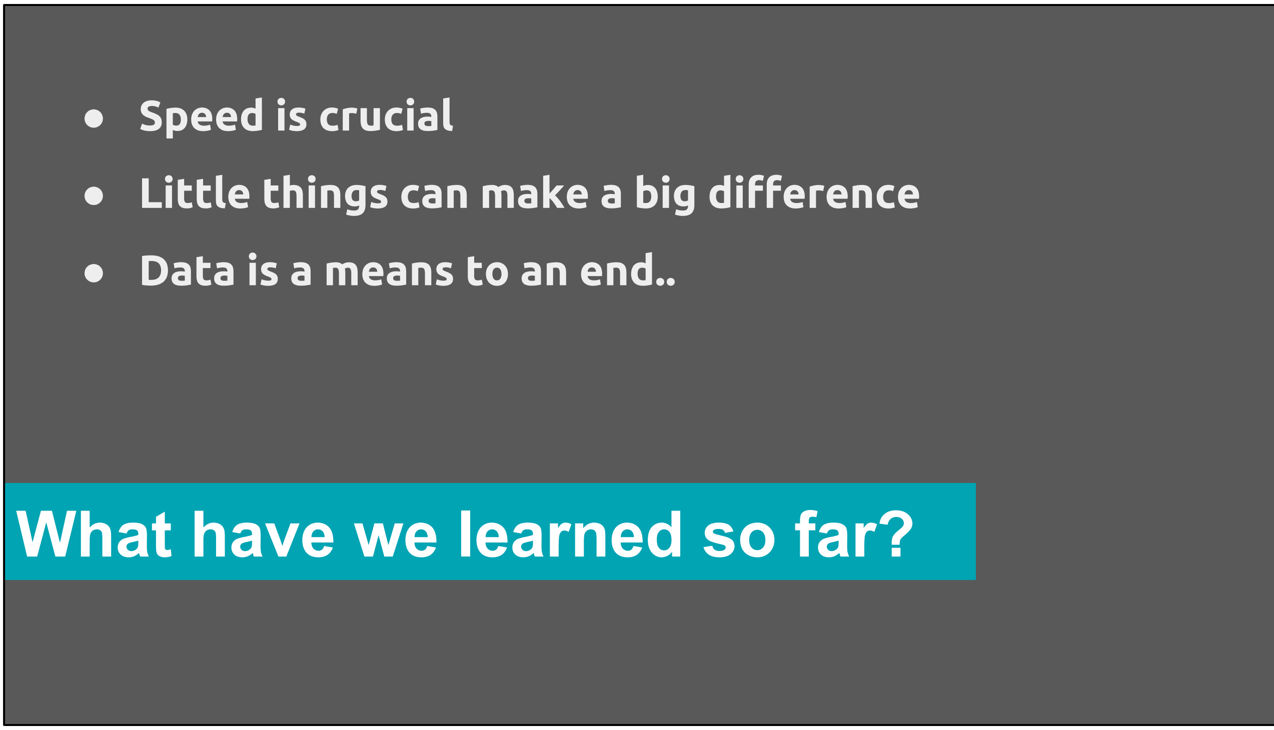
Finally once we had clear customer needs and clear metrics we let the team get on with solving the problem including finding the right solution.

Of course we have guard-rails in place where the team justified the decisions to senior management, but importantly they own the solution and any relationships with 3rd parties. They weren’t given a tool or asked to work with a particular vendor. They owned the decision.

It is worth saying that we are still mid way releasing to customers so it’s too early to tell if we are successful - but feedback so has been great

And importantly we are continuing to learn.

And I wanted to share 3 key learnings .



The first is that

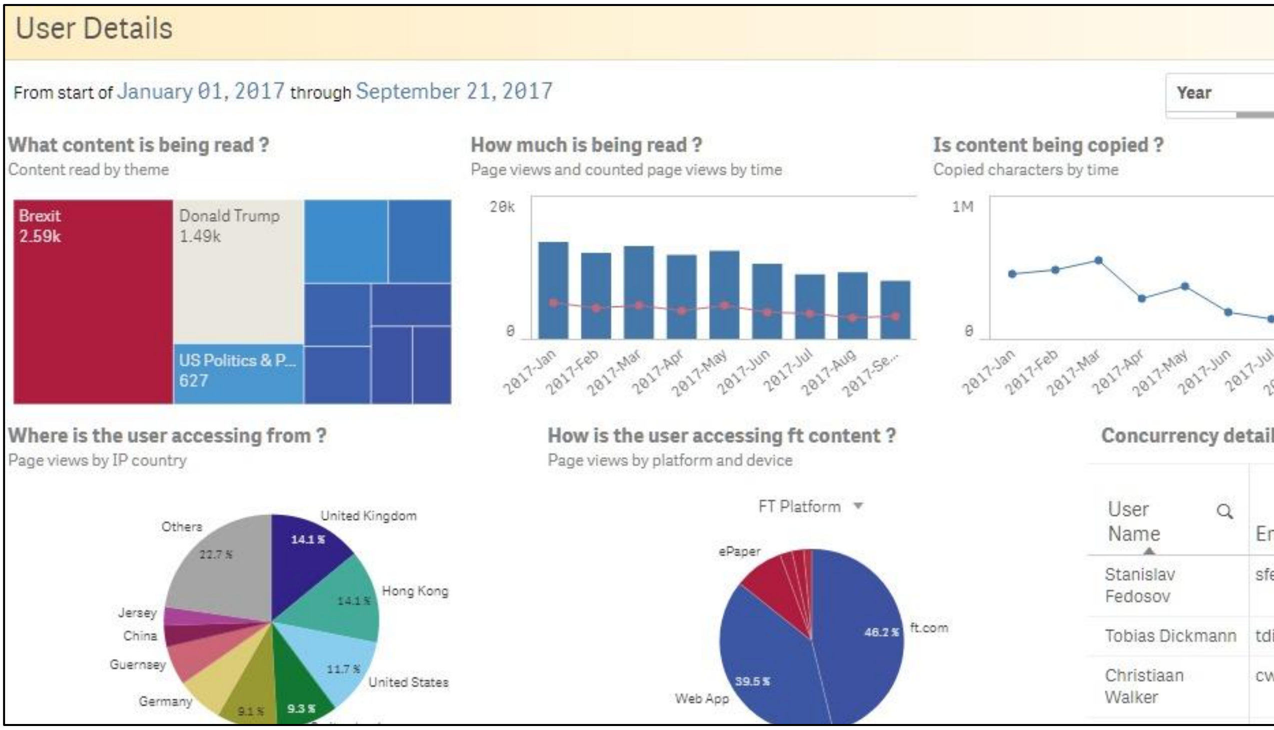
Speed (and performance) is crucial - our internal customers were frustrated if it took time to load - We didn't pay enough attention to performance at the outset and we have made to put more effort into this

Little things can make a big difference - We got a round of applause at one of our user testing sessions after adding what we thought was a very small feature (which took about 2 hours of engineering effort)

Finally and most importantly for most of our users : data is a means to an end. They aren't looking to deeply explore or slice and dice data.

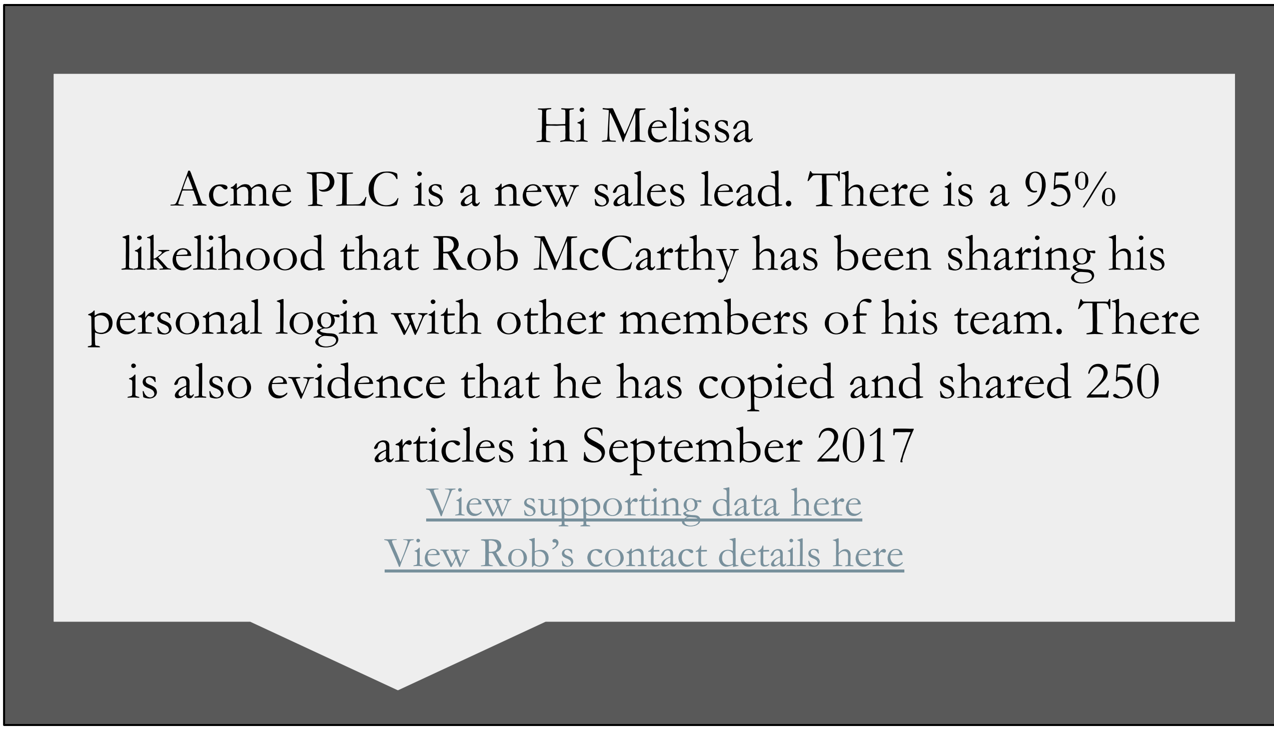
Most usage was transactional - "I just need this specific question answered "

And this learning has helped us re-imagine the solution



Because at the outset this what we imagined we would be building. Beautiful sets of dashboards that sales or marketing teams could explore / filter / drill down into

But in fact - for many of our customers what they were really looking for is this ..



A simple data driven answer to their question

That was easy to understand,

That Contained actionable insight, a clear call to action and backed up by data.

And even better that was pushed to them and available across multiple mediums

Effectively moving beyond the dashboard, to personalised data driven workflow

- Understand your internal customers needs
- Give your engineers problems not solutions
- Embrace iteration : V1 will never be “good enough”
- Decide metrics, execute, feedback, and repeat

In Summary

So 4 final summary points .

1. It is imperative that any product delivery team understands its customer’s needs. Internal customers are no exception . Make the effort to understanding how they work, what they are trying to achieve and what their pain points are. Align these to the outcomes that you are trying to achieve

2. Similarly customers / stakeholders of a project shouldn’t confuse needs with solutions. Present your problems to the engineering team and work with them to determine the solution. Good engineers will understand what is possible with technology

3. Embrace iteration and accept that v1 of any project will never be good enough - I have never seen a successful project where development stops after “Launch date”

4. Set your team’s clear metrics for success execute against them, solicit feedback and repeat - until the problem is no longer worth solving or something is more important.

So 4 final summary points .

1. It is imperative that any product delivery team understands it's customer's needs. Internal customers are no exception . Make the effort to understanding how they work, what they are trying to achieve and what their pain points are. Align these to the outcomes that you are trying to achieve

2. Similarly customers / stakeholders of a project shouldn't confuse needs with solutions. Present your problems to the engineering team and work with them to determine the solution. Good engineers will understand what is possible with technology

3. Embrace iteration and accept that v1 of any project will never be good enough - I have never seen a successful project where development stops after "Launch date"

4. Set your team's clear metrics for success execute against them, solicit feedback and repeat - until the problem is no longer worth solving or something is more important.