



Module #3 - Agile and open ways of working

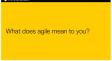
Digital Transformation Learning Modules

Time	Slide #	Script (text and actions)
Agile and open ways of working (161:30)		
Introduction (02:20)		
00:15	1 	Share screen. Welcome everyone, and thanks for attending today's session.
00:30	2 	I'd like to remind everyone of a few Zoom house rules: <ul style="list-style-type: none">• Make sure your name is displayed• Keep your video on unless you have connectivity issues• Mute if you're not speaking



		<ul style="list-style-type: none"> Don't hesitate to ask questions by raising your hand or using the chat. We want to make this session as interactive as possible to give you the opportunity to trigger conversations that you may wish to continue after this programme ends.
01:15	<p>3</p> 	<p>Today's module is a continuation of our last session where we explored the concept of human-centred design in the context of public services. We highlighted the importance of conducting user research at every step of the service development cycle to understand user needs and build services that efficiently answer these needs.</p> <p>In today's module, we'll go through the service development cycle in further detail, and try to understand how agile and open ways of working can help improve service delivery, and more generally digital transformation.</p> <p>Before we start, do you have any questions about our previous session?</p>
00:20	<p>4</p> 	<p>This session is divided into 2 parts: agile, and working in the open. They go together, but we'll explore them one by one.</p>
1. Agile working (97:40)		
a. The origins of agile (49:30)		
00:20	5	To unpack the concept of agile, we'll look at:



		<ul style="list-style-type: none"> the origins of the concept, what it means, and how it can be applied to the development of digital public services.
05:00	<p>6</p> 	<p>Group discussion</p> <p>What does ‘agile’ mean to you?</p>
01:00	<p>7</p> 	<p>Agile means many things to different people. For some, it means working from home and hot-desking. Or doing stand-up meetings (daily team meetings where everyone stands to keep them short). Some think that agile is an excuse to abandon planning. Others that it’s about doing whatever is necessary to get something done in the shortest time possible, or cutting corners.</p> <p>This varied interpretation of agile leads to lots of confusion across but also within organisations.</p>
00:35	<p>8</p> 	<p>But when you boil it all down, there’s a core set of principles that define what agile is. Once you understand those principles, you’ll have a pretty good idea of understanding what agile really is. This is what we’re going to explore in this session.</p>
03:20	<p>9</p> 	<p>Before we go further, I’d like to share a short story to illustrate what being agile means.</p> <p>On opening day, a ceramics teacher divided his class into 2 groups: 1 group would be graded solely on the quantity of work they</p>



		<p>produced, and another group solely on its quality.</p> <p>On the final day of class he would bring in his scales and weigh the work of the “quantity” group: 22kg of pots rated an “A”, 18kg a “B”, and so on. Those being graded on “quality”, however, needed to produce only one pot — albeit a perfect one — to get an “A”.</p> <p>Well, came grading time and a curious fact emerged: the works of highest quality were all produced by the group being graded for quantity. It seems that while the “quantity” group was busily churning out piles of work – and learning from their mistakes – the “quality” group had sat theorising about perfection, and in the end had little more to show for their efforts than grandiose theories and a pile of dead clay.</p> <p>The potter’s tale paints an interesting narrative that can be interpreted a number of ways. The way I see it, is:</p> <ul style="list-style-type: none"> • Those who knew they had space to fail, were likely to do more and faster • Those who had one shot, were likely to spend more time thinking before doing <p>The results are quite self-explanatory. Start small to fail fast, learn, and iterate. Not to build pots, but good government services which meet user needs. We’ll get back to this, but first, let’s explore the origins of the concept of agile.</p>
01:25	10 	<p>Where did it all start?</p> <p>At the start of the internet era, in the 90s, software developers began to challenge traditional project management methods as they found they lacked flexibility to work efficiently.</p>



		<p>In 2001, 17 software developers from different organisations decided to spend a weekend together, skiing and brainstorming in a wooden chalet on more lightweight development methods.</p> <p>While the participants didn't often agree, they managed to find consensus around 4 core values, which they summarised in what we now call the Agile Manifesto. This Manifesto was specifically written for software development.</p> <p>Let's look at these core values in further detail, and how we may build a parallel between software development and digital government.</p>
03:00	11 	<p>The first value of the Manifesto states that individuals and interactions should be put at the core of software development, rather than processes and tools. In other words, it says that people are teams' most valuable resource. Even with the best processes and tools in the world, a software development project is only successful when teams are free to do their best work, and interact with each other on a regular basis to share their ideas.</p> <p>Processes and tools are rigid, unless people transform them so that they can evolve. Conformity to rigid processes and tools sometimes makes sense, but not when it's to accommodate new ideas, discoveries and thinking, as software development requires. We can very easily become slaves to processes and tools. This is something the Agile Manifesto warns against.</p> <p>This does not apply to every aspect of the public sector of course. For example, there's no point in giving priority to people over processes in civil engineering. There's a clear millennium process to build a bridge, and the laws of physics are not going to change. But if you work on an innovation project, that has never been done before, it may be</p>



		worth giving the project team the flexibility they need to explore different options which outcomes can't be forecast.
02:00	12 	<p>The 2nd value gives priority to working software over comprehensive documentation. Software developers document the code they write. This documentation either explains how the software operates - which is useful to other software developers who may need to reuse the code, or update it - or how to use it.</p> <p>The Agile Manifesto does not say that developers should not create documentation. It says that they should not spend a huge amount of time on it, because their focus should be on getting this done, and achieving tangible results. It says that software developers should not waste their time on writing long comprehensive documentation that may be out of date in a few months time, as needs are likely to change quickly.</p> <p>This may resonate with some of you. Civil servants are often asked to write comprehensive strategic notes and reports. But their priority should remain serving citizens. A balance must be found between documenting what you do, and actually doing it.</p>
03:00	13 	<p>The 3rd value gives priority to customer collaboration over contract negotiation. In traditional project management methods, customers negotiate the solution requirements in great detail before any work starts. And at the very end, they do a box ticking exercise to check whether the solution actually meets the requirements they defined at the beginning of the project.</p> <p>In agile, the customer is an active stakeholder who's involved throughout the entire development process. This allows the customer to make sure their input is incorporated along the way.</p>



		<p>Let's take our example of bridge again. As a customer, your need is to link point A to point B, knowing that there's a river between points A and B. This is not likely to change as workers build the bridge.</p> <p>But let's say that you've hired a private company to build a citizen engagement platform. As a customer, your need is to encourage citizen participation in policy making. This has never been done in your country before. You may have looked at how other citizen engagement platforms work in other countries, but you're not sure whether they're adapted to the specific context of your country. What do you put as requirements in your contract with the private company? Agile is about giving you the flexibility to co-construct this citizen engagement platform with the service provider you've hired as you learn about user needs, and the challenges and opportunities that may arise during the project.</p>
02:00	<p>14</p> 	<p>The 4th value gives priority to responding to change over following a plan. It goes hand in hand with the 3rd one:</p> <ul style="list-style-type: none"> • The 3rd value was about having developers and customers working closely together to discuss project needs and requirements • The 4th value is about translating these discussions into actual changes to the project's plan <p>In agile, the project plan sets the direction, but the team is ready to adapt it based on incoming input and feedback. The best way to get constructive feedback is to test what you're developing, as you develop it. If software development teams wait too long before testing the product they're developing, then they take the risk of wasting time, money and efforts into developing something that's not relevant. They also take the risk of running out of time to bring changes to suit the needs of their customers and users.</p>



		This is why software developers build what we call MVPs - Minimum Viable Products. But before we introduce the concept of MVP, let's reflect on the 4 values we've just talked about.
10:00	15 <small>Do any of these 4 values resonate with you? Can you share an anecdote of a time when it would have made sense to apply one of them in your work, or a project you were involved in?</small>	<p>Group discussion</p> <p>Do any of these 4 values resonate with you? How so? Can you share an anecdote of a time when it would have made sense to apply one of them in your work, or a project you were involved in?</p> <p><i>Potential follow-up questions depending on participants' answers:</i> <i>What were the consequences of not applying this value?</i> <i>What may have prevented you from applying this value?</i></p>
01:35	16 <small>A minimum viable product (MVP) is a version of a product with just enough features to be usable by early customers who can then provide feedback for future product development.</small>	<p>A minimum viable product (MVP) is a version of a product with just enough features to be usable by early customers who can then provide feedback for future product development. It's not polished, it does not have all the features and details of a final version but it's good enough to test with some users and get constructive feedback. The whole point of using a minimal viable product is to learn, and to do it quickly without spending much money.</p> <p>The concept of MVP is not relevant for software development only. It has been applied to a range of industries, including healthcare, education and government, both for public services and policies. In the context of public services, a MVP is the simplest possible version of a service.</p>
02:50	17	Let's look at what an MVP looks like. Airbnb is now a key player in the hospitality business worldwide. But when they



		<p>started, their product was very different from what it is now.</p> <p>Airbnb’s co-founders Brian Chesky and Joe Gebbia were living in San Francisco at the time, and they had an extra room in their apartment. When they heard about a design conference happening in their city, they thought they could rent that extra room to make a bit of money, just during the time of the event, as lots of people would be looking for a place to stay, and may not be able to afford a hotel room. So they took pictures of their apartment which they put on a website. They got 3 customers.</p> <p>With time, Brian and Joe added a new feature on the website, allowing people to contact them in case they had an extra room available for rent as well. They would visit the person, take pictures of their apartment, and put them on the website.</p> <p>This is how it all started, with a website developed very quickly, offering a few rooms to the participants of an event happening in San Francisco. This MVP showed to Brian and Joe that there was a real demand and interest for what they were offering, that it was an idea worth exploring further.</p>
03:25	18 	<p>What does an MVP in government look like? In April 2020, the Government of Singapore announced new schemes and grants to help Singaporeans in need through this difficult time. GovTech Singapore (the country’s technology agency) noticed that many Singaporeans came together to share the information on these grants on social media. Some even created Google Sheets to collate, organise and disseminate the information. So what did they do?</p> <p>They rapidly developed an MVP for what was to become SupportGoWhere, an official, online portal that would allow citizens to discover and understand the different schemes and grants available in one place. But today’s</p>



		<p>SupportGoWhere website is much different than their first MVP.</p> <p>What did they start with?</p> <ul style="list-style-type: none">• First, they asked themselves about the grants that were available. To do so, they visited government web pages and reached out to government organisations driving the provision of schemes and grants, like the Ministry of Social and Family Development and the Ministry of Culture, Community and Youth.• They gathered, synthesised and collated the most relevant information, and then displayed it on a portal. They made a very conscious decision to simplify navigation by having all the applications within one page, split across 3 broad categories:<ul style="list-style-type: none">○ Means to find information and begin applying for grants○ Auto-inclusion grants that users automatically receive○ Other means of support that users can consider• They also added analytics and feedback tools to help them assess not only the site’s technical performance, but more importantly understand whether users were using the platform easily and whether they found it useful. <p>This took place in less than 2 weeks. It allowed GovTech Singapore to soon observe a good spike in site traffic which validated the benefits of the portal.</p> <p>https://www.developer.tech.gov.sg/communities/stack-x-meetups/past-webinars/supporting-the-community-part-2.html</p>
--	--	---



00:25	19 	With time, and user feedback they developed new features: <ul style="list-style-type: none">• Mandarin, Malay and Tamil options• Customer support team to provide information on specific schemes• An eligibility checker to help guide citizens to schemes which they were eligible for
10:00	20 	Activity <p>You think that a website for companies to access and bid for public tenders will allow more of them to compete. What would an MVP look like?</p> <p><i>Invite participants to answer.</i></p> <p><i>You may expect answers like websites with basic features (eg view tender opportunities and how to apply), a LinkedIn page advertising tenders, or even just a public google sheet on which different government organisations can list their open tenders.</i></p> <p><i>If the MVP shows there's a need for such a service, then the service team could design a website with further features (eg apply online, track application status, access history data, etc).</i></p>



b. Agile service delivery in government (48:10)		
00:50	21 	<p>To summarise, the concept of agile initially came from the software development world. It offered guidance for project management around a set of 4 values that highlighted the importance of being people-focused, collaborative, pragmatic and flexible.</p> <p>But the concept of agile is not relevant to software development only. The essence of the concept - its core values - can apply to many other disciplines, including service development and policy making in government.</p>
01:30	22 	<p>What does being agile mean for governments? The first thing to keep in mind is that agile is a way of thinking and acting. It is not a project management methodology like PRINCE2. It's a mindset.</p> <p>There are agile methodologies based on this mindset. Embracing the agile mindset is very much about the core values and principles and what the methodologies do is to put in a set of rules, and artefacts, that help to codify these values and principles into a way of working.</p> <p>It's important to say that you can very easily become slaves to these processes and tools and fall foul of the very first value statement in the Manifesto: individuals and interactions over processes and tools.</p>
02:05	23 	<p>Let's try to understand how the values of the agile manifesto can translate into principles for service delivery in government.</p> <p>The first principle, which we covered in our last session is to focus on user needs. I won't say much more about it.</p>



		<p>Let's focus on the next ones 'fail fast and learn quickly' and 'deliver iteratively'.</p> <p>Do you remember the potter's tale? Because they had to deliver a large volume of pots, team A very quickly put their hands in the dirt, and produced an enormous number of pots. Although they didn't know much about pottery at first, they quickly learned from their failures, and in the end, they managed to produce high-quality pots, much better pots than the ones produced by team B, which wasted most of their time thinking about how to make a beautiful pot.</p> <p>The same applies to digital services. By quickly delivering something, even something very small, you get to fail, learn and iterate.</p>
00:15	24 <small>Starting small = success</small>	<p>But this isn't a magic recipe. Starting small does not guarantee success. This is not what starting small is about.</p>
01:10	25 <small>Starting small = minimising the impact of failure</small>	<p>Starting small is a way to minimise the impact of failure.</p> <p>The reality is that for some types of projects, like innovation and digital projects, we can't avoid failure. Research shows that 70% of digital projects fail (McKinsey 2018, BCG 2020). So it's pointless being scared of failure, it will happen in some shape or form.</p> <p>By starting small, by doing the small things fast, we can minimise the impact it would have should it go wrong, as we've seen with the 'Register to your GP' example, or the biased exam grading algorithm.</p>
02:30	26	<p>We like to use the expression micro-failures to describe the types of failures that are useful and positive. This is what</p>



		<p>this chart tried to show.</p> <p>Let's say 2 teams are developing a new digital service in government. The Y-axis represents disruption and the X-axis time. The green team gets feedback about their service every fortnight, and the red team every quarter.</p> <p>Because the green team has very regular feedback, they have to deal with much less disruption. The red team on the other hand only has feedback every quarter, so when they find out what they've been doing for 3 months was wrong, they go through significant disruption.</p> <p>Both types of disruptions can be considered as failures, but I bet I don't need to ask you which one of these failures is more useful.</p> <p>Another example is if you were driving to some unfamiliar place and stopped by the roadside to check where you were on the map every 10 minutes, you would probably not be as lost as if you were checking every hour. Most importantly, the degree of course correction you have to make would be different for both of these scenarios.</p>
04:00	27 	<p>This is why MVPs - minimum viable products - are so important not only for tech startup companies, but for governments too. We looked at the example of the SupportGoWhere service in Singapore. In that case, the MVP showed that the team was on the right track, developing the right solution for a problem many people faced. But it's not always like this.</p> <p>Let's take another example of a digital service project from the UK health services organisation, the NHS. If you want to benefit from public health care in the UK, you need to register with a general practitioner (GP). To register with a GP practice, most people either visit the practice and pick up the necessary forms, or download them online and print</p>



		<p>them.They fill in the forms at home and bring them back to the GP practice. Patients have to provide contact information and their medical history. Each GP practice has its own form, although they very much look alike.</p> <p>The NHS wanted to make it easier for patients to register to a GP, by offering them the opportunity to do it online, whatever GP practice they'd be registering to.</p> <p>To test their idea, they built an MVP which consisted of an online registration form. They tested it with 6 volunteering GP practices. They found out that depending on the GP practice, patients were not always happy with the service. This is because even though they could go through the online application process online, they still had to visit the practice to give a proof of their identity. So the MVP didn't really bring significant advantages.</p> <p>Which is why they abandoned the project, and wrote a blog post about it, as you can see on the screen. Building an MVP allowed them to save time, money and effort in building a large-scale solution that would have failed to meet user needs.</p> <p>https://digital.nhs.uk/blog/transformation-blog/2018/winding-down-register-with-a-gp</p>
02:50	28 	<p>But MVPs don't only make economic sense, they're also a way to limit unintended consequences, which is particularly important when working on services or policies that vulnerable populations depend on. Small tests and pilots can help reduce unintended harm in public services and policies, by forcing a focus on frontline feedback and observed outcomes before scale.</p> <p>In 2020, because of the covid-19 crisis, the UK Department for Education cancelled the A-levels. In the UK, A-levels are the set of exams taken by students around the age of 18. They're the final exams taken before university, and they</p>



		<p>have a huge impact on which institution students attend. Universities make offers based on students' predicted A-level grades, and usually, a student will have to achieve certain grades to secure their place.</p> <p>So, to calculate grades, the government put together an algorithm that relied primarily on 2 pieces of information to calculate grades: the ranking of students within a school and their school's historical performance. The system was designed to generate what are, on a national level, broadly similar results to previous years.</p> <p>But instead, the algorithm gave higher grades to students from fee-paying schools, and decreased grades for students from poorer backgrounds. This led to complaints and protests that eventually led the government to abandon the algorithm, and rather ask teachers to estimate students' grades.</p> <p>https://www.theverge.com/2020/8/17/21372045/uk-a-level-results-algorithm-biased-coronavirus-covid-19-pandemic-university-applications</p>
02:15	29 	<p>Let's go back to our principles of agile and summarise what we've just said:</p> <ul style="list-style-type: none"> ● Fail fast and learn quickly: Every agile methodology is about helping service teams to identify where what they're doing might not work, and give them the chance to change it before they invest too much time and effort. It reduces the likelihood of big failures, and turns small failures into lessons. ● Deliver iteratively: big bangs don't work. The goal of service teams is to answer user needs, and because these can change over time, they need to keep iterating. There's no point in developing the perfect solution to the wrong problem.



		<ul style="list-style-type: none"> ● Keep planning: as said at the beginning of this session, there's a myth that agile means no planning. The opposite is the case. In agile, service teams are planning just enough to get going, and then they're continually improving and tweaking that plan based on the needs they're trying to meet. So agile actually often requires more planning than traditional project management. <p>Let's take a closer look at this.</p>
02:45	30 	<p>Traditional service or policy development usually follows this pattern, which is called the waterfall approach. It's a linear approach: the previous phase of the project must be completed before moving on to the next.</p> <p>Everything starts by someone, somewhere defining a thing they want to build or change. For this change to happen, or for the new service to exist, they capture a list of requirements. What does the thing need to do? This can take a while and leads to large specs or requirements documents.</p> <p>Then comes the design and implementation stage, which can either happen in-house, or be outsourced.</p> <p>Finally, once all the development is done, the service is launched to users. This is the moment where the team finds out whether the initial requirements captured everything correctly.</p> <p>The waterfall approach is often opposed to the agile approach. It's indeed radically different in several ways:</p> <ul style="list-style-type: none"> ● It's restrictive in the sense that it does not allow changes in the project plan. Once the design plan is fleshed out, waterfall does not allow for any changes.



		<ul style="list-style-type: none"> Users are not involved in the development of the project, and testing only happens at the very end. <p>This does not mean that one approach is better than the other one. It means that agile and waterfall approaches are suited to different types of projects.</p>
02:00	31	<p>When does it make sense to use the waterfall approach? We used the example of a bridge construction earlier.</p> <p>The requirements for a bridge to hold are well-known to engineers, and they're not ever going to change. Once the architects design a plan, the workers build the foundations, and cement the whole thing. There's no way this order can change. So it makes sense to use the waterfall method here.</p> <p>Waterfall methodologies don't apply to the infrastructure sector only. In digital, in order to maintain a stable product which does not require regular functionality changes, it can make sense to use the waterfall methodology.</p> <p>If there's absolutely no ambiguity in the service requirements, if all technologies involved are perfectly understood by all team members, and if the project does not take too long, then the waterfall methodology should do.</p> <p>But it's rare to find such a scenario when developing new digital services in government. The internet era has significantly changed the way we behave, and the pace of digital innovation keeps increasing. User needs change faster than they used to, and so do the digital technologies supporting government services.</p>
00:15	32	<p>What does the agile approach look like in the context of service development? This is the graph we used in our last session.</p>

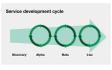


00:40	33 	Yet it would be more accurate to depict an approach in 4 stages: Discovery, Alpha, Beta, Live. I'll talk a bit about each of these in a moment, but what's key here to understand is that by the time a service 'goes live' it should have already been used by the people the service is for.
02:30	34 	<p>Discovery is a short, time boxed period to explore the needs of users. Discovery is about understanding the problem to solve. It's about understanding who the users are, and what they're trying to achieve. As seen in our last session, this is the phase where they do research, analyse data, and talk to users.</p> <p>The Discovery phase is also the opportunity to dig into the constraints and challenges the team may face trying to meet those needs. Whether that's tech, regulation, commercial, or skill.</p> <p>At the end of the Discovery, the team should know whether or not to continue to the next phase. If they think a new or updated service could make it easier for users to do the thing they need to do, then they can move on to the alpha phase. If they feel there's nothing they can do, or should do - taking into account their constraints - they can stop there. That's fine! It's not a failure to stop at the end of the discovery phase if research shows that's the best thing to do. In fact, the team will be saving time and money that could be better spent elsewhere.</p>
00:50	35 	<p>But let's take a case where it makes sense to continue to Alpha. Alpha is where the service team tries out different solutions to the problems they learnt about during discovery.</p> <p>This is the phase where they build prototypes and test different ideas. By the end of alpha, they should be in a</p>



		position to decide which of the ideas they've tested are worth taking forward to beta.
03:30	36 	<p>The beta phase is where the team takes their best idea from alpha and starts building it for real. It also involves thinking about how the service will integrate with existing services, to prepare for the transition to live.</p> <p>A beta version of the service should actually work. It needs to be connected to the right systems in the right places. It might not work for every edge case or have every feature, but it should be the smallest version of the service which meets a user need.</p> <p>Beta versions may be private or public. A private beta is a version that's only accessible to a group of test users who've been pre-identified and briefed. A public beta is accessible by anyone. It's a way to gather feedback at scale.</p> <p>You can see on the screen the public beta of the UK's companies house service. It's a website with open data on UK-based companies. It includes financial accounts, company filings, details and on directors, etc.</p> <p>You can see it's a beta from the top left corner orange sticker. It's important to let people know that they're on a beta version of the service to:</p> <ul style="list-style-type: none">● manage their expectations<ul style="list-style-type: none">○ it's work in progress, so users should not expect a perfect service. If everything goes well, great! But if they run into an error, they shouldn't be too harsh○ the service does not include all the functionalities it should yet. Here the team put in the bottom right corner a list of the features they're planning on developing, just to let users know it's not an oversight.



		<ul style="list-style-type: none">invite them to share their feedback <p>https://www.gov.uk/government/news/launch-of-the-new-companies-house-public-beta-service</p>
00:55	37 	<p>This is the live version of the UK's companies house service. You may have noticed a few changes from the beta, both in terms of design and features (eg advanced company research).</p> <p>The live phase of a service is about supporting the service in a sustainable way, which means continuing to iterate and make improvements.</p> <p>As you can see here, the service team invites users to tell them what they think of the website.</p>
00:50	38 	<p>This is where they land if they click on the link. It's a short form inviting them to share how they use the service, how often, what could be improved, etc.</p> <p>Of course changes to the service take place less often in the live phase than in the beta phase, and the service team is much smaller, but it does not mean everything's over.</p>
01:00	39 	<p>To recap, the agile approach to service development is divided into 4 phases. The first one - the Discovery phase- is about understanding user needs. The next 2 phases - Alpha and Beta - are about releasing things early, to test them with actual users. Based on their feedback, service teams iterate, add features, make improvements. Or delete things that don't work.</p>



		Working in this way reduces risk and avoids big failures. It turns little failures into lessons.
05:00	40 	<p>Activity</p> <p>Would you use agile or waterfall? Why?</p> <p>Share a poll, and invite participants to explain why they chose agile or waterfall.</p> <ul style="list-style-type: none"> • Build a solution to better match substitute teachers with vacancies - agile - this is a greenfield project, the requirements are unclear. It'd be worth interviewing substitute teachers and school headmasters first. • Add a text message option to an existing notification system - waterfall - this is very straightforward, and should not take long to do. • Tech companies regulation - agile - tech giants have emerged in the internet era, like Google, Facebook, Amazon. It's hard to grasp the full impact of this on the economy, and what the consequences of new regulation could be. An agile approach would make more sense here. • Transform the first floor of the Ministry of Economy into an open space - waterfall - there are not that many ways to go about it. First the plan, then the demolition stage.
00:30	41 	The last agile principle for us to explore is 'keep improving how your team works'. This is very intertwined with part 2 of this session on working in the open. But before we jump there, let's look at what a digital service team looks like.
04:20	42	The key message to remember here is that to be successful, teams need to be multidisciplinary from the start to the



end of projects. This applies both to policy and digital service teams.

A multidisciplinary team is a team that gathers people with different backgrounds and different expertises. This is important because they have a different knowledge, and they see things differently. The only thing that members of multidisciplinary teams have in common is the objective they're pursuing, as a project team.

What does a digital service team look like in government? To simplify things we can say there are 2 main types of experts: digital experts, and subject matter experts. Subject matter experts may be specialised in areas like social policy or competition law, or supporting functions like operations, public spending, and human resources.

There's a wide range of expertises in digital: from user research, design to software development and data science.

Traditionally government forms teams around agency structures. This usually works in a linear way. Many teams are responsible for different parts of the service. Specialists often don't get the chance to work together on creative solutions.

Multidisciplinary teams change this by forming around a problem or service. The team starts with all the right skills to deliver the service from the beginning to the end.

By bringing together people from different disciplines into the same team, working in a truly collaborative and open way, teams can better understand the end-to-journey, from the public that interacts with the service to the back office systems and the staff that use them. From there, they can design processes and systems that are more effective and efficient, and that meet user needs.

To summarise, the 2 main benefits of multidisciplinary teams are that:



		<ul style="list-style-type: none">• people that used to come in at the end (and often too late to do their best work) have input from the beginning, and• people with complementary capabilities work together at the same time on the same part of the service, so the team is never blocked, and can quickly make decisions.
02:15	43 	<p>This is a screenshot of a blog post written by a procurement expert - a suit and tie type of guy - who shares his experience of working in a multidisciplinary team in the UK government. He describes how surprised he is to arrive in an environment where everyone wears casual clothes, stands for meetings, don't wait for senior people to share their opinion to give theirs.</p> <p>He goes on to describe how they welcomed him in the team, and how efficient their collaboration was: "I've learnt a lot, I've shared a lot and I've also removed the need for the weekly 2-hour meeting. It's great. We talk instead of sending emails and I understand so much more about the project"</p> <p>His 4 takeaways are:</p> <ul style="list-style-type: none">- Being part of the multi-disciplinary team enables cooperation and teamwork- This in turn creates shared knowledge- Which results in less process duplication, less resource wasted, less management- Meaning more doing, more delivery, better public services. <p>I'll share with you the link in case you'd like to read the whole post.</p>



		Share the link: https://www.linkedin.com/pulse/procurement-needs-part-multi-disciplinary-team-kershaw-mcips-msc/
01:30	44 	<p>Digital service teams should not only be multidisciplinary, but diverse. It's one thing to have representatives from different departments, with different skill sets. It's another to have team members of different genders, ethnicity, social or academic background.</p> <p>There is lots of statistical evidence that shows that diverse teams make organisations more productive. But more importantly, people see the world differently. If everyone in a team looks alike, they're less likely to come up with challenging ideas and consider service users in their diversity. In a nutshell, diverse teams are more likely to build inclusive services.</p> <p>Diversity in teams is always key, but even more so in digital service teams because there's a strong gender gap in tech.</p>
01:55	45 	<p>This is an example of what happens when men design technology for men. In 2014, Apple released their health app, with a bunch of health indicators like heart rate, footsteps, weight, calories or even daily sodium intake. Can you think of anything missing on the Apple health app on the left? Apple completely ignored menstruation, a bodily function experienced by more than half of the world's human population at some point in their lives. It took a year for Apple to update their app with a menstrual cycle tracker. Had they had more women in their team, they would have certainly included a menstrual tracker early on.</p> <p>This wasn't the first time a tech product had prioritised men over women. The vast majority of technology companies are staffed by men, especially on the development side. The first artificial heart implants were designed to fit 80</p>



		percent of men but only 20 percent of women because of their size. Diversity in technology teams matters. https://www.theatlantic.com/technology/archive/2014/12/how-self-tracking-apps-exclude-women/383673/
Short break (05:00)		
2. Working in the open (64:15)		
a. Core principles of working in the open (09:15)		
00:50	46 	Let's move on to the 2 nd part of today's session: working in the open. We'll start with defining the core principles of working in the open, before looking at what it means practically. We'll look at this at 3 levels: the team, the organisation and the public sphere. And we'll try to understand how working in the open can create value.
05:00	47 	Group discussion What do you think working in the open is about?
01:30	48 	Working in the open can be summarised to 3 main principles: 1. Communicating in short, frequent updates, without 'big reveals' Working in the open is about removing barriers to knowledge. The best way to do this is to speak a little and often



		<p>rather than saving everything up for a ‘big reveal’.</p> <p>Working in the open starts with how – and how often – people talk about their work to their team. It includes how teams speak to other teams and to the wider organisation. How leaders speak to senior leaders. And of course, we can work in the open when we communicate with people outside the organisation too. These people could be service users, journalists, people looking for a job, etc.</p>
00:30	<p>49</p> 	<p>2. Speaking honestly, without spin</p> <p>Working in the open is also about honesty. It’s not about hiding things and it’s not about ‘putting spin’ on something to reframe it to try to reduce any negative impact it might have.</p>
00:45	<p>50</p> 	<p>3. Acknowledging that things will go wrong, admitting it when they do</p> <p>Working in the open is about getting comfortable with the fact that things do not always go to plan. And that that is ok. It’s about acknowledging that things will go wrong and talking about them when they do so that you and others can learn from them.</p>
00:10	<p>51</p> 	<p>In a nutshell, working in the open is not just about the destination.</p>
00:30	<p>52</p>	<p>It’s about people showing their work and processes. It’s about the “how they got there”. It’s about the story behind the thing. And why they made the decisions they did to get there.</p>



b. How and why working in the open (53:05)		
00:15	53 	Now let's look at what working in the open means practically, and how it can facilitate digital transformation in government.
i. At the team level (21:00)		
00:10	54 	First, what does it mean for a team to work in the open?
01:45	55 	<p>In traditional project management, governance usually takes the form of lengthy weekly, fortnightly or monthly meetings. They often require lots of preparation, and are centred around the most senior people at the table, as project members ask them for their approval on decisions they need to make.</p> <p>Governance is radically different in agile projects. Instead of long spaced-out meetings, agile teams meet much more often, usually on a daily basis. This allows them to make quick decisions, and deal with challenges as they arise. This helps to minimise the risk of project failure or delay.</p>



		<p>These regular short meetings (5 to 15 minutes) are called standups, because people are invited to stand, rather than sit. It's a way to remind people to keep the meeting short and to-the-point. Obviously this team has taken standups to the next level.</p>
02:25	56 	<p>Another type of meeting that agile teams organise to work in the open are 'show and shares'. They include product demos and project retrospectives. How these meetings are called is not very important. The point is that teams get together to share their work and what they are learning along the way.</p> <ul style="list-style-type: none">● Product demos - Teams can sometimes find it difficult to hold product demos where they show something that is not 'fully baked' because they fear it won't land well or that people will think it's the final product. In many organisations there's a cultural expectation to only show 'completed' work, which, unfortunately, gets in the way of early feedback!● Project retrospectives - A retrospective is a practice used by teams to reflect on their way of working, so they continuously improve how they work. When teams reflect this way, and regularly, they are able to detect when things might be going wrong in their team and not have to wait until things become toxic and affect delivery negatively. These retrospectives also help the team to detect and find solutions to problems that might be getting in the way of their effectiveness.
02:15	57 	<p>Another way to work in the open is to share weeknotes. This is something I invite all of you to do, whatever team or project you may be involved in. Weeknotes are reflections on your working week. You simply need to write about what you've done and how you feel about how it went.</p>



		<p>Weeknotes are flexible. They can be from one person to their team, from a leader to the people they lead, from teams to leaders, sponsors or stakeholders, from one team to other teams, from supplier to client, etc.</p> <p>At the team level, weeknotes allow people to learn about progress made, and the role of each team member in the project.</p> <p>Weeknotes should be fun to write. You can include jokes, pictures, anything really. There's not a defined structure nor an ideal format for weeknotes. Some weeks you may just want to write 4 to 5 lines, and some weeks 4 or 5 paragraphs. If there's not much to say, that's fine. Weeknotes are not about reporting. They're about communicating.</p> <p>You can see on the screen an article with good practices on how to write weeknotes. I'll share with you the link in the chat.</p> <p>Share the link in the chat: https://wow.how/to-/write-weeknotes</p>
02:30	58 	<p>Standups, Show & Shares, weeknotes - they're tools teams can use to work in the open. What benefits do they bring?</p> <p>The first - and maybe the most important one - is trust. Team members who trust each are much more likely to work efficiently together, and build good services. How can you build trust in a team? People tend to give their trust once to people who trust them. And there's no better way to show your trust than showing your vulnerabilities. Vulnerability is not about oversharing or talking about mistakes in an attempt to gain sympathy. This will likely achieve the opposite. It's about:</p>



		<ul style="list-style-type: none"> • asking for help on a project. If you are not an expert on a topic – or even if you are – seek input from your team members, • asking for feedback, • admitting to mistakes. <p>This might not seem so easy. In many hierarchical organisations, people are reluctant to share their mistakes or admit when they need help because they’re afraid they’ll appear weak, and give a bad image of their work and capacities to senior leaders. Leaders need to embrace the agile culture, and encourage their team members to work in the open, both to celebrate successes and learn from failures.</p>
01:05	59 <small>Why working in the open at the team level</small>	<p>Another reason it is good to work in the open is feedback. The more eyes there are on a service the better it gets - mistakes are spotted, better alternatives are pointed out, the bar is raised. I’m thinking particularly about project demos and retrospectives here, but it can also be just about sharing early drafts. Whether you’re working on a service, a policy, or a report, it’s always good to share your ideas early to get feedback. Getting new thoughts, different positions, and other ideas usually improves drafts.</p>
00:50	60 <small>Why working in the open at the team level</small>	<p>Working in the open helps tremendously with governance. Teams that are continuously providing information about how things are going, don’t need specific governance meetings, and preempt information requests even before people ask for them.</p> <p>There are some organisations that have more or less removed the need for formal reporting because they have a culture of working in the open and their leaders have embraced it.</p>



10:00	61 	Group discussion What might stop you from working in the open in your team?
ii. At the organisation level (04:40)		
00:30	62 	Working in the open should also take place at the organisation level. Weeknotes may be relevant here as well. But in general, it's harder to get people's attention on a project if they're not close stakeholders.
01:50	63 	Hence the concept of information radiators. The radiator is constantly emitting warmth. Those closest feel the most warmth but there is an attractive nature to it which makes everyone willing to get closer. Whiteboards can be considered as information radiators. If you happen to walk past them, they will catch your eyes and make you want to take a closer look at them. It's a way for teams to invite anyone within their organisation to find out more about what they do. Do you remember this picture? We saw it in our last session when talking about human-centred design. The team who designed this refugee settlement service printed information about its design, and displayed it on a wall, explaining what it is. It's a way to capture people's attention to deliver them information, or even asking them for feedback (eg by making post-its and pens available).
01:00	64	Obviously remote working makes this more of a challenge. But information radiators don't have to be physical, they



		<p>can be virtual too.</p> <p>This is the Trello board of London’s Smart City team. Trello is a collaboration tool that allows teams to ideate, plan and manage projects through online boards. Boards can be made private or public. The London team decided to make their board public, to let anyone know about their missions, and the actions they took to achieve them.</p> <p>https://trello.com/b/CloKi2mP/smarter-london-together-report-card</p>
01:20	65 	<p>When you are open about what you are working on, you are more likely to find others who might be working on a similar problem and with whom you can collaborate. There’s a famous quote that goes “if I give you my apple and you give me yours, we end up with an apple each. But if I give you my idea and you give me yours, we both end up with two ideas!”. A really good and topical example of this is how countries and pharmaceutical companies came together to develop the Covid vaccine. This cooperation was a huge aspect of the speed of delivery.</p>
iii. At the public level (30:00)		
00:10	66 	<p>Last but not least, service teams can communicate with the outside world.</p>
02:45	67 	<p>The most obvious way to do this is through blogging and social media. Teams can blog to share information and engage with people, either to crowdsource ideas or get feedback from a more diverse set of voices.</p>



		<p>You can see here extracts from 4 blog posts from digital service teams in Singapore. One's about how they built an inclusive voucher system for the government, another gives tips for conducting fieldwork with senior citizens, another is about the challenges of dealing with legacy systems and another one on someone's experience as a software developer working in government.</p> <p>By communicating openly about challenges they ran through, and good practices they've identified, they do a service to other teams, who don't have to learn all of this by themselves, but can reuse what has been already learnt.</p> <p>In some organisations, blogging and communication with the outside world more generally is the work of communication experts only. Everything made public needs to first go through a series of approval procedures, and not everything can be shared - especially failures. Change doesn't happen overnight. But slowly, governments need to embrace working in the open. Even banking institutions - which are known to be extremely cautious with everything they share because they're very sensitive to reputational risks - have started to do so.</p> <p>https://www.tech.gov.sg/media/blogs/listing</p>
01:50	68 	<p>Here's an example. Monzo is an online retail bank, it has had a great success story in the UK, but not without bumps. As you can see here, in a blog post they wrote in August 2019, they encountered some significant issues.</p> <p>For a while, an outage prevented users from logging into the app, sending or receiving payments, and even contacting anyone through the app.</p> <p>Two weeks later, they published this blog post. As opposed to just apologising and moving on, Monzo public blog</p>



		<p>detailed what the issue was and what they did to fix it.</p> <p>As a user, wouldn't you feel your money is safer in the hands of someone who recognises failure, and takes the time to explain to you why they failed and what they did to prevent this from happening ever again, than someone who just apologises?</p> <p>This also applies to government services.</p>
00:30	69	<p>Here are a few blogging good practices that you may find useful:</p> <ul style="list-style-type: none"> ● Be clear and brief. Assume that everybody is too busy, most of the time ● Write as you speak. Tell stories from humans to humans ● Focus on the work as it's happening
01:10	70	<p>The same applies to social media. Communicating through social media allows service teams to reach a wider audience than their usual blog followers. Depending on countries, the most popular social media may change. You can see here a screenshot from:</p> <ul style="list-style-type: none"> ● the LinkedIn page of the Singapore Smart national and Digital government Office, with a post putting forward the inclusivity of their services and the career path of one of their female engineers ● the Twitter account of the Jabar Digital Service, the digital service team of the West Java region in Indonesia about their efforts in strengthening digital literacy.



02:05	71 	<p>Digital service teams can publicly share blog posts, as well as code. Let's take a closer look at coding and software development. The computer you're using right now relies on 2 things: hardware (the actual physical object), and software. Software is a collection of instructions that tell your computer how to work. People in charge of developing software are called software developers, or developers. Their job is to write code - ie a human-readable text written in a specific programming language. The purpose of the code is to give the computer precise rules and specifications that it can translate into machine language.</p> <p>Developers who code in the open make their code available for others to view, copy, and modify for their own purposes. They often do this on a website called GitHub. As you can see on the screen, many public authorities already do this, at the national, regional or city level, like the governments of Australia, New Zealand, the Philippines, the Republic of Korea and the digital service team in West Java, Indonesia. Let's try to understand why they've decided to share the code that's behind their digital services.</p>
00:50	72 	<p>In 2020, at the beginning of the covid-19 crisis, the Government of Alberta in Canada created a covid-19 self-assessment tool. People didn't know much about the covid-19 symptoms at the time, and lots of people were worried and called the emergency services which quickly became overwhelmed. So they created an online form for people to self-assess their infection risk.</p>
00:40	73 	<p>Soon after this, the Government of Ontario contacted them, and used their code to build their own self-assessment tool. They brought some changes to the tool, to adapt it to their specific context. But by not starting from scratch, they saved time and efforts, during a period when every minute counted.</p>



01:40	74 	<p>Coding in the open can be about helping other departments and governments. But it's also about making your own services better, like the UK online petitions service. It's a website which allows UK citizens to organise and vote for petitions online. If a significant number of people sign a petition, then the UK Parliament has to include it to their meetings' agenda.</p> <p>The digital service team shared the code of the petitions service in the open. Someone outside of government - Leandro Alemao - looked at their code and suggested adding a feature that showed the date scheduled to debate the topic of the petition. As you can see on the right, the digital service team liked the idea, and merged the code that Leandro wrote to their code, to add the new feature.</p>
00:25	75 	<p>Working in the open at the public level, either through blogging or code sharing is a way to share knowledge and work to make others save time and efforts into building or learning similar things.</p>
02:35	76 	<p>It's also a way to own your narrative. If you don't tell your story, some others will. And you never know what they might say! We want to own our own narrative because it makes us more believable and more trustworthy.</p> <p>At some point Amerigo Vespucci - whose name was given to America - was one of the most hated explorers on earth. He had embarked on an exploration to America years after Christopher Columbus, and realised that Columbus had not found a new way to India, but what he called a 'new world'. Some people reinterpreted Vespucci's story, and published a book where it said that he had actually been the first one to lay foot on the continent, which should therefore bear his name - America. When the truth was later revealed, Amerigo Vespucci was highly criticised, people said he was jealous of Columbus, and a liar who only cared about fame.</p>



		<p>Had Vespucci had access to the internet at the time, he may have written a blog post with his true story, and shared the link to Columbus' twitter account for him to take the rightful credit of discovering America.</p> <p>Digital service teams, and government teams more generally, need to be transparent about what they do.</p>
10:00	<p>77</p>	<p>Group discussion</p> <p>What assumptions are made about your organisation? And what is the narrative you'd like to tell?</p>
00:10	<p>78</p>	<p>Being open with the outside world is also a way for digital teams to gain visibility, and legitimacy.</p>
02:10	<p>79</p>	<p>In 2020, Bolor-Erdene Battseengel was appointed the Chairwoman of Mongolia's Communications and Information Technology Authority, with the mandate to oversee the implementation of e-Mongolia, the online one-stop-shop for government services. At 27 years old, she was the youngest person and first female to hold this position.</p> <p>She didn't have many friends in government. People didn't trust her, and didn't want to lose control over the services that were about to be hosted on the e-Mongolia website.</p> <p>So she made the press her friend. She invited journalists to events where she showed them what e-Mongolia looked like. She was very transparent about the work they were doing. She didn't wait for e-Mongolia to be finished, or</p>



		<p>perfect. She welcomed their feedback, kept improving the website, in an open and transparent manner. e-Mongolia became very popular, both in the media, and in the population.</p> <p>Her colleagues in government ended up having no choice but to acknowledge her work, and support the development of e-Mongolia.</p>
01:10	80	<p>Another thing working in the open can do is to attract talent. It's not an easy task to attract digital experts in government, because they're in high demand, and private companies often offer them high salaries.</p> <p>But there are other aspects that can make a job position attractive: its purpose, the team, the environment, and their ways of working. But this is very hard to assess for applicants, unless they already know someone in the organisation.</p> <p>Speaking at public events or blogging about your team or your organisation can help attract talent.</p>
00:50	81	<p>This is an example of how the blog of the UK grocery retailer Co-op has attracted talent. If you look at the tweet on the right hand side, Caroline a software engineer says she used to read the blog back when she didn't work at CoopDigital but it was her dream job, she used to read the blog all the time and think "wow they do cool stuff, what a great place to work".</p>
01:00	82	<p>Last but not least, working in the open is a way to build trust, with citizens but also other actors, for example organisations from the local technology sector, who may be able to support governments in their digital transformation. Being open and honest about what service teams do, and how taxpayers' money is spent is the only way to build a good and long-lasting relationship between them and government organisations. Building trust in</p>



ASIA AND THE PACIFIC
**Regional
Innovation Centre**

**public
digital**

		government is the topic of our next module. I look forward to seeing you again then.
00:15	83 	Thank you all! I look forward to seeing you again for our next session!