



HM Revenue  
& Customs



# Digital delivery unpackaged

A practical guide to establishing a digital delivery capability

28 January 2016

# Contents

## i. FOREWORD

## ii. ACKNOWLEDGEMENTS

## 1. EXECUTIVE SUMMARY

- 1.1 The digital delivery vision
- 1.2 Principles for successful digital delivery
- 1.3 The evolution of digital delivery
- 1.4 Conclusion

## 2. INTRODUCTION

## 3. THE VALUE OF DIGITAL DELIVERY

- 3.1 The UK Government digital agenda
- 3.2 The digital delivery vision and culture
- 3.3 Agile delivery
- 3.4 Digital delivery progress in HMRC

## 4. MOBILISING A DIGITAL DELIVERY CENTRE

- 4.1 Leadership
- 4.2 Culture
- 4.3 Operating model functions
- 4.4 Process
- 4.5 Governance
- 4.6 Metrics
- 4.7 Staffing a digital delivery centre
- 4.8 People and capability development
- 4.9 Interfaces
- 4.10 Tools
- 4.11 DevOps
- 4.12 Physical location
- 4.13 Communications
- 4.14 Work Prioritisation

## 5. OPERATING AND CONTINUALLY IMPROVING A DIGITAL DELIVERY CENTRE

- 5.1 Continuous improvement
- 5.2 User experience
- 5.3 Integration between digital and traditional delivery methods
- 5.4 Supplier management
- 5.5 Scaled platform vs. service model
- 5.6 Live support model
- 5.7 Open source technology
- 5.8 Security of digital services

## 6. MANAGING MULTIPLE DIGITAL DELIVERY CENTRES

- 6.1 Communications
- 6.2 Centralised Continuous improvement Function
- 6.3 DevOps
- 6.4 Work Prioritisation
- 6.5 Governance
- 6.6 International Expansion of Digital Delivery Centres

## 7. CONCLUSION

## 8. GLOSSARY

## i. FOREWORD

### *Foreword by Mark Dearnley*

Since the first dotcom revolution in the late '90s the challenge for most organisations has been to deliver amazing digital services that integrate with, and transform, existing business processes and systems. Doing this at scale, with pace, over multiple geographies, and in an environment where products are continually changing just adds to the complexity. Technology that hardly existed five years ago is now making it much easier and cheaper to deliver digital solutions. But this is only part of the answer. People are the key to delivering all transformation. How they work, what skills and experience they have, and the environment in which they operate makes the difference between delivering interesting pieces of technology, and the amazing digital services we all love to use.

Government is no exception. The Government Digital Strategy, published in 2013, set out challenging plans for the redesign of government services. HM Revenue and Customs (HMRC) has over 800 years of legacy processes and systems. It is responsible for over 70% of all government transactions. We have an ambitious vision to make tax digital with probably the biggest programme of transformation in the Department's history.

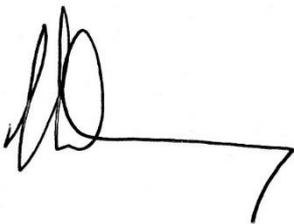
Our journey over the last two years has been significant. We now have 450 digital specialists working in five state-of-the-art digital delivery centres across the UK and a healthy pipeline of digital talent, helped by award-winning graduate and apprentice schemes. In the last year alone we've brought to life 17 new digital services, and in 2016 we will complete the rollout of digital tax accounts to five million UK businesses and 50 million individuals.

The purpose of this manual - or 'the book' as we've come to refer to it - is to share what we've learned so far about setting up a digital delivery capability. When we started our journey under three years ago this document would have been immensely helpful to us. So we hope it helps you.

Our digital journey in HMRC is far from over. There's lots still to achieve and some big challenges to overcome, particularly given the size and scale at which we operate, but we have a growing confidence and belief in our ability borne out of the journey we've been on. We are always open to new ideas and improvements on the way we do things, so please give us feedback so this 'book' can keep evolving.

A number of individuals and organisations have helped us along the way and made this manual possible. I'd like to thank everyone involved, particularly Accenture who co-authored it with us.

I hope you find something in here that helps you on your journey!

A handwritten signature in black ink, appearing to be 'MD', with a long horizontal line extending to the right and ending in a small hook.

**Mark Dearnley**  
Chief Digital & Information Officer

## ii. ACKNOWLEDGEMENTS

HMRC has partnered with Accenture, Equal Experts and Capgemini to build its digital delivery capability. The development of this paper, co-authored by HMRC and Accenture, involved contributions from HMRC and each of its digital delivery partners. Key contributors are listed below.

<b>Name</b>	<b>Current Position</b>
<b>Michael Potter</b>	HMRC Digital Transformation Director
<b>Brigid McBride</b>	Deputy Director, HMRC Digital Transformation Team
<b>Antony Collard</b>	Deputy Director, HMRC Digital Delivery Centres
<b>Steve Rowlands</b>	Deputy Director, HMRC Multichannel Digital Tax Platform
<b>Jason Kay</b>	HMRC Digital Transformation Team
<b>Richard James</b>	Digital Service Manager, HMRC Personal Tax Account
<b>Jane Andrews</b>	Digital Service Manager, HMRC Business Tax Account
<b>Joyce Robertshaw</b>	Digital Service Manager, HMRC Benefits and Credits
<b>Multiple contributors</b>	Government Digital Service
<b>Dom Pia</b>	Accenture
<b>Vish Papineni</b>	Accenture
<b>Rebecca Hudson</b>	Capgemini
<b>Alun Coppack</b>	Equal Experts

## 1.0 EXECUTIVE SUMMARY

### 1.1 *The digital delivery vision*

The UK Government is committed to becoming “digital by default”. This means delivering “*digital services that are so straightforward and convenient that all those who can use them will choose to do so, whilst those who can’t are not excluded*”.<sup>1</sup>

The government digital agenda aims to create digital services that are more efficient, convenient, secure and accessible than the manual services they replace. The outcome will be a more positive customer experience - by making transactions quicker and more reliable, digital services save time and money for members of the public, as well as government departments. The increased transparency and accuracy provided by digital services will also help the public sector work more effectively, getting things right first time. HMRC is embedding digital delivery at the heart of its business, so it can achieve its three strategic objectives of improving customer experience, reducing cost and maximising yield.

Public sector organisations are working with Government Digital Service (GDS) to achieve their digital goals. In particular, they are transforming their organisations to enable them to deliver digital services, often by setting up large-scale Digital Delivery Centres (DDCs). Creating a DDC is more complex than staffing a new IT Delivery Team and involves enterprise-level change.

It’s clear that digital services radically change the way that public services are delivered. They push the **primary** focus towards meeting **the needs of the end user**, rather than the business. In practice, this means that an organisation using digital services must aim to align these two needs more closely, so it can deliver customer-led services that benefit both the public and the business. Every aspect of a DDC needs to support this aim: the people, culture, processes, governance, ways of working, physical location and technology should all be designed to enable the delivery of high-quality, user-focused services.

DDCs use agile delivery methods. Agile is a way of working that focuses on building and releasing critical functionality quickly, then iterating and continuously improving the service. Successful DDCs develop working environments and cultures that are highly collaborative and focus on the continuous improvement of both the digital services they produce and their own ways of working. This represents a challenging, yet exciting transformation of public sector IT delivery.

The purpose of this document is to provide clear principles, activities and considerations for the successful development and operation of a large-scale digital delivery capability. The content is primarily based on HMRC’s experience since 2013 in establishing and operating large-scale digital delivery across its five DDCs in London, Newcastle, Telford, Yorkshire and Worthing.

---

<sup>1</sup> <https://www.gov.uk/government/publications/government-digital-strategy>

## 1.2 Principles for successful digital delivery

There are 10 principles that organisations should follow in order to successfully mobilise, operate and continuously improve a large-scale digital delivery capability.



Figure 1: Principles for successful digital delivery



### Start small, learn from doing

You should use agile delivery to encourage your team to focus on delivering the minimum functionality required for a product to start delivering value. This is known as delivering the Minimum Viable Product (MVP). The product can then be iterated further to add more features as needed. You should use the same approach when setting up the DDC. Start with a focused period of mobilisation that aims to deliver the DDC MVP i.e. the minimum DDC components you need in order to start small-scale digital service delivery. You don't need to set up a new DDC location to get started. You can make practical changes quickly by seating teams together, rearranging existing desks to encourage collaboration, printing out and using agile metrics, and using agile tools and meeting styles, such as sprint planning, and daily stand-ups (see glossary).

Once the DDC MVP is in place, establish one or two experienced 'Scrum teams' (the term used for a Service Delivery Team) and learn from doing. Simultaneously, continue with your priority mobilisation activities and use these to prepare your DDC for scaled agile delivery, whilst at the same time ramping up the teams. As you learn new lessons, you can continually improve the DDC - and as you achieve real and demonstrable progress, this will underline your commitment and build momentum.



### A culture of continuous improvement

Successful DDCs instil a culture of continuous improvement. To achieve the agile benefits of risk reduction, shorter time-to-market, and fit-for-purpose delivery, it is important that the Service Delivery Team iterate the digital service, rather than merely adding new functionality upon each release. GDS specifically asks for "evidence of iteration" - in other words, evidence that the product or service has changed direction by applying learning from previous releases. Projects that deliver large chunks of functionality in a serial

manner risk delivery failure should the project over-run. Instead, you should strive to release an MVP that will form the basis for all subsequent releases. Each further release should add significant value rather than incremental detail. An agile project is about more than just an incremental delivery: it is about having a continuous improvement mind-set that repeatedly inspects and adapts the product and its delivery process. User feedback may result in features being enhanced or dropped if they don't provide enough value.

It is equally important to iterate the DDC itself. Inevitably, there will be issues which impact on delivery in a new DDC, when there are a large number of new people using new processes and technology. So you should seek to continually improve its delivery processes and ways of working by establishing a central continuous improvement team. They will be responsible for identifying issues that are impacting team effectiveness and for implementing continuous improvement solutions to mitigate these issues. The continuous improvement team should contain functional leads who are responsible for setting design and development standards and ensuring the consistency and quality of services (e.g. design, architecture and development).



### Share ownership and trust the teams

As well as having a dedicated continuous improvement team, all team members in the DDC should take responsibility for continuously improving the DDC, so it takes a shape that best fits its team members. Directed by a 'Scrum master', the team should continue to ask themselves "*How can we deliver faster?*" Fostering a start-up mentality, with team members assuming a sense of responsibility and accountability for the progress of the DDC, will help to embed this commitment to continuous improvement.

The mobilisation and operation of the DDC is not just an activity for management. People need to feel part of the centre and assume a sense of ownership for the DDC 'product' that is being developed. You should include individual teams and cross-DDC communities (e.g. developers, designers) in the design and implementation of mobilisation and continuous improvement activities. Working cultures grow organically: the working practices and collaborative behaviours that best fit your DDC will gradually evolve if the wide range of skills and perspectives of the DDC team members are included in its development.



### An unremitting focus on value

Everything within the DDC should be planned to deliver value. You should design your operating processes so they identify and deliver value quickly and on an ongoing basis (e.g. through user research). Technology should be streamlined so it can be released quickly and adapted rapidly in response to changing user needs. Your metrics should align to business value drivers (e.g. reduction in calls), and should be used by the product owner to prioritise development activities.

However, processes, technology and metrics are only important if they are supported by a mind-set that is focused on the final outcome. You should assign passionate digital practitioners who are committed to the development and ongoing delivery of high-quality digital services. The question for DDC Teams should not be "*What can we deliver by point X, many months in the future?*", but rather "*How long will it take us to release an MVP that delivers some value?*" Releasing the service is not an outcome in itself. The product owner needs to consider the operability of the service and must strive to deliver a sustainable and continuously improving service that provides value on an ongoing basis. As services become 'live' earlier in the delivery lifecycle, delivery teams need to develop the mind-set that they own the live service and must lead all operations, monitoring and continuous improvement activities. This repositions the delivery approach, with 'value' being the primary objective.



### User-centric services

Digital delivery practices put the end user at the heart of the service being developed. IT services used to be developed based solely on requirements defined by the business, but digital services are developed based on requirements defined by end users (i.e. the public). This leads to services that are led by customer need and fit-for-purpose, delivering greater benefits for both government departments and the public they serve.

Digital delivery involves user research and user-centred design at each stage of the delivery process, incorporating user feedback and ensuring their requirements are given extensive attention throughout. Responding to evolving user needs means that the service design and scope – as well as the techniques and technologies needed to deliver it – are often in a state of flux. To successfully deliver these projects, you need to adopt a responsive test-and-learn agile approach.

The introduction of DDCs and user-centred design has opened up a new channel of communication between citizens and government bodies. For the first time, public sector organisations are receiving feedback on their policy and technology, and revising both based on this feedback. Establishing such a fluid feedback loop helps to deliver better-quality, user-focused services. However, it does raise certain challenges with the need to balance and align the needs of the end users and business. Organisations need to expand this user-focused approach to their legacy services: what do customers need from these services, and how can business policy align with user needs? The increase in UX specialists and user research facilities that DDCs bring can be used to increase policy innovation and responsiveness in line with user expectations.



### **Becoming a digital enterprise and aligning change**

Mobilising a DDC in isolation will limit the benefits of digital delivery. Consider what other changes are needed throughout the rest of your organisation to embed digital operations, support the new digital services and become a truly “digital enterprise”. Digital customers expect a consistent and joined-up experience, and your organisation should be structured to meet this expectation. For example, HMRC is moving many existing services to the cloud, modernising and consolidating its IT estate, developing digital contact centre services (e.g. web chat) and moving to agile delivery on many of its legacy systems. DDCs are a key part of any digital enterprise, but they must be supported by change throughout the rest of the organisation - and effective digital leadership is essential to support these high levels of change.

At a digital delivery level, you should invest time in developing an effective approach to managing the organisation’s backlog of potential projects. Establish a process for conducting a light touch evaluation of potential new projects to ensure that a) there is a user need, and b) that it is best served by a digital service. You should consider appointing a lead product owner for each area of the business, who has overall ownership for the digital product vision, and the ability to prioritise different digital services and functionality within their business area.

Appointing effective product owners to lead service delivery is a key factor for success in any new DDCs. They must work closely with the delivery team, as well as own and make all key decisions on the service direction. The product owner is responsible for the ongoing operation of a high-quality service, striving to deliver as much value as possible. This is a new role for many organisations, but is critically important, as the product owner replaces groups of business stakeholders as the sole owner and decision-maker for the service.



### **The most important part of a DDC is its people**

If you want your DDC to be a successful, exciting place to work, you need to staff it with passionate, digital specialists who possess a continuous improvement philosophy and development approach. But selecting and training your team members is one of the greatest challenges for a new DDC.

Digital Delivery Teams are self-organising and cross-functional, and may require skills that don’t currently exist in your organisation. So, to inform your recruitment, you should identify a target team composition, while remembering that the exact size and composition of each Scrum team, including the number of people in each role, should vary according to the size and scope of the project. Consider the optimum mix of internal, external and supplier resources you’ll need. Bringing people together from a range of different backgrounds is a challenge, because individuals need to be effectively integrated into your organisation’s operating model. However, by bringing in people from different sources, you can also create important opportunities - for example, your department staff will be able to develop their own capabilities through exposure to experienced digital practitioners. When you are recruiting externally, remember that the DDC is competing with digital private sector organisations (i.e. Facebook, Google, Amazon) for the best talent, so your recruitment approach and employee value proposition must be tailored accordingly.

Developing an approach to recruit, upskill and monitor the development of your staff is a key dependency for a successful DDC. You will need a structured approach to capability development in order to create high-performing teams in a short space of time, build the skills you need to deliver high-quality digital services and deliver on the commitments made during the recruitment process. These skill sets, technologies and learning approaches are likely to be new to your organisation, so this should be a priority area to address when you are setting up your DDC.



### **Automate and streamline as much as possible**

Digital delivery is at its most effective when software development and deployment is highly automated. Continuous integration, automated testing and automated deployment tools and infrastructure will enable agile teams to rapidly develop and release incremental functionality. This technology is important if you want to maximise the benefits of agile.

DevOps should be a top priority activity for any new DDC, because a strong DevOps platform can help you to automate and streamline the development process. It involves increasing the automation of your operations, security and testing functions, so you can reduce the number of interactions required and the number of dependencies between operations, development, security and testing. DevOps functions provide the technical building blocks to allow agile Scrum teams to create and scale environments, develop and maintain automated testing and deploy to development and production environments. This will help you to streamline the deployment pipeline and increase the number of releases to end users, delivering value earlier.



### **Autonomous teams, common standards**

Digital delivery teams work effectively when they have high levels of autonomy. Teams may follow certain mandatory governance and delivery processes, but they should be self-organising, with the power to define exactly how they will work together most effectively (e.g. by using agile methodologies such as Scrum and Kanban). Autonomous teams have limited dependencies, which reduces the amount of disruption caused by coordinating with external teams. You should form teams with limited external dependencies, so they are self-sufficient and can define their most effective working practices.

When you are establishing a DDC, you should consider the degree of standardisation and reuse that you will need in order to develop your applications. If you intend to increase the proportion of applications that you develop in the DDC, and there is a strong desire or need for a common look and feel, along with architectural compliance, then it may be sensible to have a platform. In such cases, you should establish cross-DDC functional leads, as well as authorities in design, user research, development and architecture, who can drive consistency and quality across all new digital services. The consistency created by these functional leads can reduce the time you take to develop new applications - for example, developers will get more familiar with using the same tools and environment, and your library of reusable components and services will increase. So, while teams have autonomy and flexibility on how services are developed, the functional leads can ensure that each individual service conforms to certain organisational, or platform-level, digital design and development standards.



### **Every DDC is different**

Every organisation is different, so we do not advocate a single, prescriptive process to deliver a DDC. Rather, this paper provides a proven framework, based on HMRC's experience, which your organisation can adapt and develop to meet your vision, objectives and user requirements.

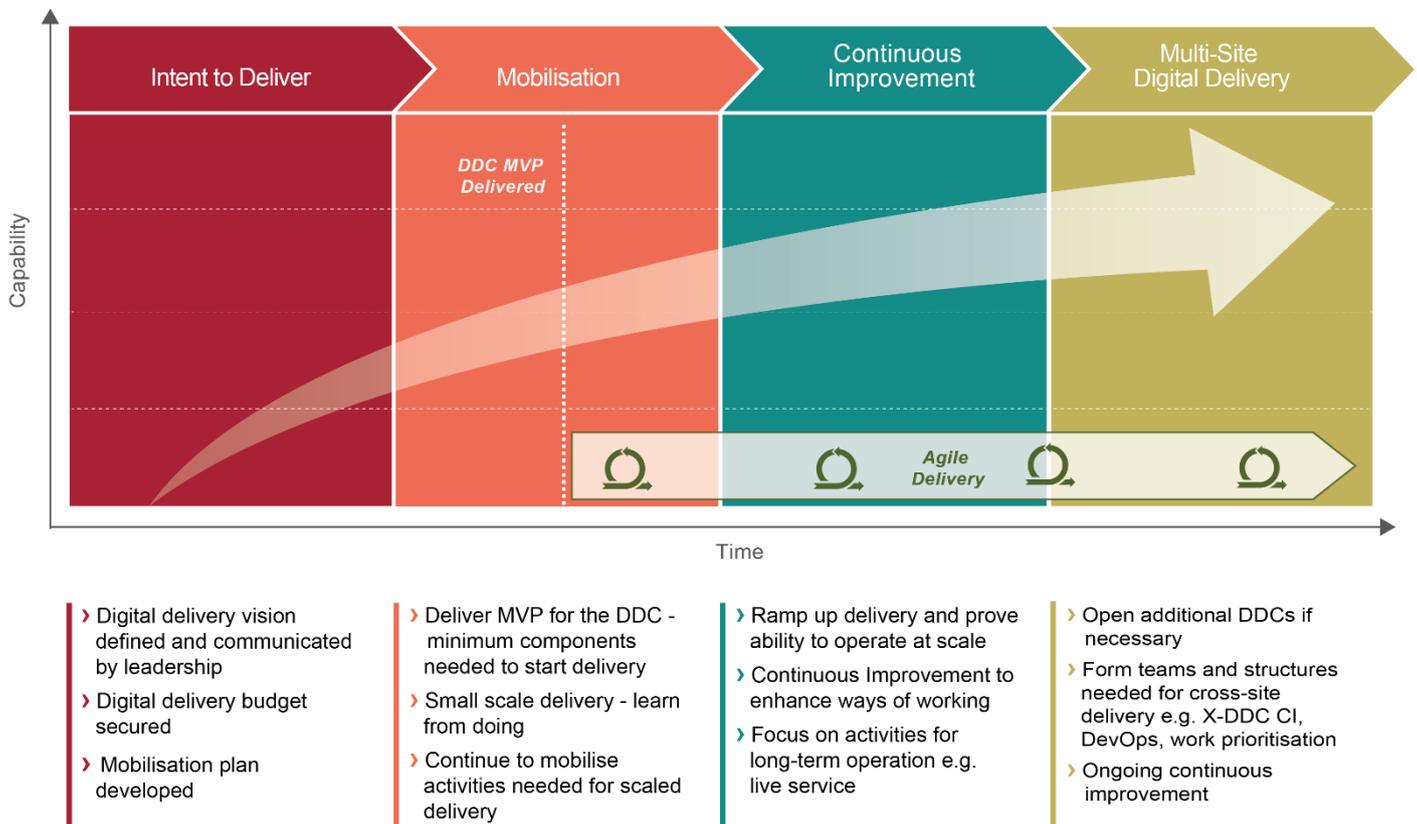
We recommend carrying out some priority mobilisation and continuous improvement activities, but recognise that how you prioritise these activities, and their underlying detail, will vary widely between DDCs. Even within the same organisation, the mobilisation priorities and challenges will vary between DDCs.

Digital service delivery is highly flexible, involving a series of behaviours, practices and approaches that can flex and vary to meet each organisation’s requirements and capabilities. As such, some organisations may approach the activities in this paper in a different order. The topics raised in this document should help you reach tailored decisions on digital delivery for your business. However, developing a culture of continuous improvement is a key factor in successful digital delivery - and this should be considered throughout the life of any DDC.

### 1.3 The evolution of digital delivery

We believe that the evolution of digital delivery involves the following stages:

1. **Intent to deliver** – defining a digital delivery vision, with support and buy-in from leadership, and beginning to make the necessary cultural changes.
2. **Mobilisation** - delivering the DDC’s minimum viable product, starting small-scale delivery to test and learn while simultaneously continuing to mobilise and ramp up.
3. **Continuous improvement and optimisation** – developing a culture of continuous improvement at service, team and DDC levels, establishing a dedicated continuous improvement team supported by all DDC team members, and scaling and iterating the delivery approach.
4. **Multi-site digital delivery** - scaling digital delivery across multiple sites, implementing central functions, and agreeing how work will be prioritised and managed across multiple locations.



**Figure 2: The evolution of digital delivery**

We believe in this framework, but recognise that the strands of work within each stage and their relative priority order will vary between DDCs. Each of these stages of evolution are explored below.

### 1.3.1 Intent to deliver

**Leadership:** Developing a new DDC requires sponsorship and buy-in from senior leaders within your organisation from the outset. They must secure the necessary funding and visibly champion a clear vision for the DDC, externally and internally. Everyone involved needs to understand their role in achieving the digital objectives. You should employ digital leaders with experience in leading digital business transformations, and/or large-scale digital delivery, to supplement the existing subject matter expertise of your organisation. This will also help to build enthusiasm and increase the pace of change. Not only should they be good leaders, but they should also be passionate and knowledgeable about digital, and empowered to implement the changes to the organisation's operating model, technology estate, people and services that digital delivery will offer. They need to secure buy-in and cooperation from other impacted leaders across the organisation, so strong leadership is required throughout the development of your digital delivery capability.

**Mobilisation planning:** Achieving real and demonstrable progress will underline your commitment and build momentum. We recommend iteratively rolling-out components of the DDC – starting with an initial dedicated DDC mobilisation team responsible for leading the set-up of the DDC. The mobilisation team provides the initial operational-level leadership you need to enable the DDC to start delivering, and will need to support the initial teams as the DDC ramps up. The Mobilisation Team should scope and plan the DDC's minimum viable product.

**Culture:** Digital delivery is as much about cultural change as it is about changing technology or delivery methodology. For many in the DDC itself, as well as the wider organisation, it is a new way of working. At this stage, you should focus on building familiarity with the agile way of working for anyone who will be involved in digital delivery. Cultural change is an ongoing process that will take time. Key cultural changes include:

- The approach to developing digital services, following an agile methodology
- Greater levels of collaboration within the DDC, and between the business and technology;
- Alignment of end user and business requirements
- The increase in number and frequency of live releases, many of which will still be going through the development lifecycle
- Embedding a continuous improvement approach to digital services and DDC ways of working

The first steps towards successfully making these changes should take place in the *Intent to Deliver* stage, though they will take considerable time to fully implement.

### 1.3.2 Mobilisation

Your dedicated mobilisation team should first focus on delivering the DDC MVP. Once this is in place, you should set up one or two experienced exemplar Scrum teams. After this, continue to conduct the mobilisation activities you need for large-scale digital delivery, and continuously improve your ways of working to increase delivery effectiveness. These mobilisation activities can run in parallel to delivery from the exemplar teams. You should gradually roll out and enhance the DDC operating model, increase the number of teams, build the technology stack and target infrastructure, whilst continually improving the DDC ways of working through lessons learned and by refining longer-term plans. As delivery teams join the DDC, you should involve them in these mobilisation and continuous improvement activities.

While mobilisation activities may vary between DDCs, we recommend you consider the following as part of your mobilisation:

- **Operating model** - define how the DDC will be organised to meet its business goals. This involves the following components:
  - o *Functions* – how work is organised to deliver services
  - o *Processes* – the activities required to complete the work
  - o *Governance* – how to make, sponsor and enforce the right decisions
  - o *Roles / Organisation Structure* – who is accountable for doing the work
  - o *Metrics* – how progress and business value are tracked and measured
  - o *Interfaces* - how to interact within the DDC and externally to deliver services

- **Recruitment** - identify, attract and recruit the right people to fill the open roles at the DDC. Use a mix of internal staff, new recruits and suppliers
- **Learning and capability development** - implement a structured approach to on-boarding, capability development and monitoring key skills and behaviours
- **DevOps** - define and implement your development approach, architecture and infrastructure to support streamlined delivery of digital services
- **Tools** - select and implement the tools required at all stages of delivery and DDC operation
- **Physical location** - design and build an engaging work space that encourages high levels of team interaction and joint-working
- **Communications** - deliver a communications strategy that engages the internal and external stakeholders that the DDC needs to involve in order to succeed;
- **Work prioritisation** – establish a streamlined, value-driven approach to selecting, approving, allocating and managing digital projects
- **Leadership and culture** – as mentioned in the *Intent to Deliver* section, strong leadership and ongoing cultural change are required to maximise the benefits of digital delivery.

### 1.3.3 Continuous improvement and scale

The role of mobilisation is to focus solely on the delivery of the DDC components needed to operate at scale - but continuous improvement has two distinct roles. Firstly, it's about the ongoing development of the areas covered during mobilisation, and secondly, it's about expanding your focus to new areas which you need in order to effectively operate the DDC. The mobilisation work strands remain important, but after addressing those priority activities, some focus should turn to the next tier of DDC delivery activities.

While an aggressive ramp up of the initial teams can be beneficial, it is likely that multiple teams will experience similar issues which impact on their delivery (e.g. process, governance, tools, infrastructure, team forming, learning and more). Rather than continuing to ramp up at a constant pace, we recommend pausing or reducing the speed of ramp up and deploying a dedicated continuous improvement team. This team will be responsible for identifying delivery issues and risks, then designing and implementing mitigating solutions that benefit all your delivery teams.

When the DDC reaches capacity and your initial delivery issues have been resolved, the continuous improvement team should continue to operate. It can lead initiatives to improve ways of working and delivery processes. The team should also act as an overarching design and development authority for all digital development activity, setting standards and ensuring consistency and quality of service design, user research, architecture, development and other activities. As with mobilisation, team members throughout the DDC should be involved in continuous improvement. This will foster the collaborative, solution-oriented and flexible culture needed for successful agile delivery.

We recommend focusing on the following additional areas as part of continuous improvement<sup>2</sup>:

- **Continuous improvement team** – establish a team to deliver solutions that mitigate common delivery risks across teams, drive continuous improvement in the DDC, and set common standards and principles for service design and development
- **User experience** – improve the DDC approach to UX and user-centred design
- **Integration between agile and legacy delivery teams** – embed methods for managing dependencies and integration between DDC agile teams and legacy teams
- **Supplier management** – develop a collaborative working approach with supplier delivery partners
- **Live support** – implement an approach to effectively manage and support digital services that are live to the public
- **Security** – Implement central, platform-wide, and service-level security measures, based on outcome-focused risk assessments
- **Platform vs. service model** – establish common, consistent services, on a reusable and scalable environment which supports further development of services

---

<sup>2</sup> The relative prioritisation of these focus areas may vary between DDCs. For example, some DDCs may choose to focus on some or all of these 'continuous improvement' areas as a priority during mobilisation.

- **Open source** - consider publishing your code in order to contribute to the open source community, and improve the modularity and quality of the code the DDC develops
- **Learning** – ongoing development of staff capabilities as maturity increases and more DDCs are established.

### 1.3.4 Multi-site digital delivery

Where greater digital delivery capacity is required, it may make sense to set up new DDCs in different geographic locations - but maintaining consistency and quality across a multi-centred digital delivery organisation is challenging. We recommend following the approach above for all new DDCs, reusing any existing learning and materials to increase the pace of delivery. For example, you could reuse operating models, learning and capability development materials, documented technical guidance, tools, infrastructure and lessons learned from any existing DDCs in your organisation. You should also consider moving people from the new DDC into teams in more experienced DDCs to support the development of people’s skills.

Where managing multiple DDCs, we recommend considering the following cross-DDC functions:<sup>3</sup>

- **Communications** – establish effective communication channels between DDCs, building a clear understanding of the communications approach
- **Continuous improvement** – set up a centralised continuous improvement team to build the continuous improvement mind-set across locations, drive up the quality of products and processes, and ensure consistent design, development and architecture standards
- **DevOps** – implement an appropriate DevOps model for multi-site distributed digital delivery
- **Work prioritisation** – define how to approve, prioritise and schedule work across multiple centres in order to drive economies of scale without creating silos
- **Governance** – implement a single governance structure across all DDCs that supports rapid development and release of functionality.

## 1.4 Conclusion

Building and operating DDCs is not easy. It requires change throughout the business and takes time to achieve. However, the benefits are clear. Specific activities may vary between organisations, but structuring activities based on the principles and framework highlighted in this paper will help you to quickly mobilise and scale your digital delivery capability.

Use this content to plan your DDC, then learn by doing.

*Accenture and HMRC co-authored this document with extensive input from Equal Experts and Capgemini, HMRC’s other digital delivery partner organisations.*

---

<sup>3</sup> As above, the underlying detail within each of these areas may vary between DDCs.

## 2.0 INTRODUCTION

The UK public sector is transforming the way it serves the British public. Increasingly, public-facing digital services are replacing more manual, labour-intensive legacy approaches to public service. These high-quality digital services are developed in collaboration with citizens. They meet real end-user needs and build stronger relationships between public sector departments and the people they serve.

As government departments invest more in digital services, they need to build their own internal digital delivery capabilities. This means a step change in the way that technology is developed, with agile, iterative and collaborative delivery techniques replacing traditional approaches to IT development.

The purpose of this document is to provide clear, practical guidance on the activities and considerations necessary to set up and operate a digital delivery capability. The content is primarily based on HMRC's experience since 2013 in establishing and operating large-scale digital delivery across its five DDCs in London, Newcastle, Telford, Yorkshire and Worthing.

This document is divided into the following chapters:

- 1) **The value of digital delivery** – we consider how digital delivery is changing the way that public services are developed and delivered, with a focus on HMRC's recent experiences.
- 2) **Setting up a Digital Delivery Centre** – the key activities and considerations required to mobilise and operate a large-scale Digital Delivery Centre (DDC).
- 3) **Operating and continually improving a Digital Delivery Centre** – after mobilisation, there are additional considerations needed to support and continually improve digital delivery.
- 4) **Managing multiple Digital Delivery Centres** – the key activities and considerations required to operate and scale multiple distributed DDCs.

Throughout this guide we make recommendations alongside specific examples of activities that support digital delivery. We also provide relevant case studies and lessons learned. We recognise that no two organisations are the same and that there is no single definitive answer to many of the questions raised below. Nonetheless, the topics raised will help you to reach decisions on digital delivery for your business.

Both public and private sector organisations can use this document to explore their own approach to successfully developing high-class digital delivery capabilities and public facing services.

*Accenture and HMRC co-authored this document with extensive input from Equal Experts and Capgemini, HMRC's other digital delivery partner organisations.*

Please note that information in this document is accurate as of its publication date in January 2016.

## 3.0 THE VALUE OF DIGITAL DELIVERY

This chapter outlines the reasons for the increase in digital delivery in the public sector, highlights the key principles for effective digital delivery and summarises agile delivery – the predominant approach used to develop digital services. We compare agile working with waterfall delivery, highlighting the benefits as well as the increased focus on the needs of the end-user.

Finally, we look back on the progress that HMRC has made in establishing five large-scale DDCs.

### 3.1 The UK Government digital agenda

In 2012, the UK Government published its first cross-departmental Digital Strategy<sup>4</sup> in which it committed to becoming “digital by default”. This was explained as providing “*digital services that are so straightforward and convenient that all those who can use them will choose to do so whilst those who can’t are not excluded*”.

The government digital agenda is driven by a number of factors:

- digital services are more efficient, convenient, secure and accessible than the more manual services they replace and provide a better customer experience
- by making transactions quicker and more reliable, digital services save time and money, both for the public and government departments
- increased transparency and accuracy help the public sector work more effectively, getting things right first time
- the increase in digital services has stimulated the UK’s technology industry and helped create jobs.

The Government Digital Service (GDS) has worked with individual departments to help them develop new digital services and achieve their digital objectives. GOV.UK was developed by GDS as the cross-departmental home of public sector digital services. It has improved the quality and increased the volume of available government digital services. GDS encourages using agile delivery methods for all digital development projects, because agile focuses on building the highest priority digital service functionality quickly and iterating and releasing frequently. End user requirements are now central to service delivery.

### 3.2 The digital delivery vision and culture

Digital delivery radically changes the way that public services are provided. The primary focus is now on meeting the needs of the end user, rather than the business. In practice, this means greater alignment between the two and providing customer-led services that benefit both the public and the business. Every aspect of a DDC needs to support that objective. The operating model, culture, ways of working, physical location and technology should all be designed to deliver high-quality user-focused services. Defining and communicating a clear vision for a new DDC is also key to success. This is a challenging, yet exciting transformation of public sector IT delivery.

#### 3.2.1 High quality, customer-focused services

*A commitment to user centred design has resulted in over 90% user satisfaction for many of the services delivered by HMRC DDCs.*

Digital delivery practices put the end user at their heart. Previously, IT services were developed solely on business requirements and the public were rarely involved in developing the services they would use. Now, digital delivery involves user research and user-centred design at each stage of the delivery process. It incorporates user feedback and makes sure their requirements are given extensive attention throughout. Responding to evolving user needs means that the service design and scope – as well as

---

<sup>4</sup><https://www.gov.uk/government/publications/government-digital-strategy>

the techniques and technologies needed to deliver it – remain fully flexible. To achieve success, a responsive test-and-learn agile approach is needed.

The introduction of DDCs and user-centred design has opened up a new channel of communication between citizens and government bodies. For the first time, public sector organisations are receiving feedback on their policies and technology and are making revisions based on this. This fluid feedback loop helps to deliver better quality, fit-for-purpose, user-focused services that benefit both customers and public sector organisations. But, it does raise certain cultural challenges with the need to balance and align the needs of customers and the business a top priority. These cultural challenges are explored in section 4.2.

### **3.2.2 The philosophy and collaborative culture of digital delivery**

A positive philosophy and collaborative culture underpin successful digital delivery. An agile project is more than just an incremental delivery: it's the continuous improvement mind-set behind repeated inspection and adaptation of a product and its delivery process. User feedback may result in features being enhanced – or even removed – if they don't provide enough value.

Team members at every level should assume responsibility for the development of the DDC. Its mobilisation is not just an activity for management. Employing passionate, digital specialists, who share the continuous improvement philosophy and development approach, will help to make the DDC a successful, exciting place to work.

*HMRC followed certain principles in the design of its DDCs:*

- *To build transformational DDCs that support the development of outstanding digital services, faster and cheaper*
- *To employ talented digital specialists while simultaneously developing its internal workforce*
- *To instil an exciting digital culture*
- *To build vibrant, modern digital working environments across multiple sites that embrace true agile, collaborative working*
- *To create digital 'centres of excellence'*
- *To make the DDCs great places to work!*

For the full benefits of digital delivery to be realised, the team needs to actively collaborate: that means live debate and developing real-time solutions **as a group**. This doesn't come easy to some teams, but with talented team members, encouraged by experienced agile coaches, it can be achieved. Successful DDCs will invest in the necessary agile ecosystem to support user-focused delivery. This includes a modern, vibrant working environment that supports high levels of interaction and group working. It also includes cutting-edge development tools, an infrastructure that automates testing and deployment and supports frequent release of functionality, as well as collaboration tools to share information quickly between any distributed teams.

Figure 3 demonstrates an agile team "swarming". This means working outside of, or beyond, job roles and working together as a team on all the activities needed to develop the product. This increases team productivity and reduces the time taken to deliver the project, due to the reduction in handovers and working collaboratively to solve the highest priority problems.



**Figure 3: A “swarming” cross-functional agile delivery team**

Working in this way also makes a difference to people’s experience at work. HMRC has seen an 11% increase in engagement within its digital teams compared with the rest of the delivery organisation.

### 3.3 Agile delivery

Agile is the preferred approach for digital service delivery in the public sector and<sup>5</sup> has a strong affinity with digital delivery. Digital transformations are interactive, evolving, and customer facing – this makes agile the natural choice of delivery approach. It helps accelerate the delivery of value, and it helps mitigate the risks associated with quick emergence of new technologies and rapidly evolving markets. This section describes the key differences between agile and waterfall delivery methods, and outlines the agile roles, processes and delivery artefacts.

#### 3.3.1 Agile versus waterfall delivery

For many organisations, agile delivery involves a departure from the waterfall approach previously used to deliver IT solutions.

Waterfall delivery involves specific sequences of development, with formal design, build and test teams and phases. It involves significant planning up-front as well as an agreed set of requirements and outputs at an earlier stage. Typically, there’s also more paperwork involved.

*Focusing on the MVP, HMRC DDCs have been able to start realising the business value of some digital services within 2-3 months from project inception.*

Agile delivery involves the iterative development and release of small chunks of functionality. It follows short development cycles called Sprints that encompass all design, build and test activities. Agile delivery

<sup>5</sup> <https://www.gov.uk/service-manual/agile>

encourages the team to focus on delivering the **minimum viable product (MVP)**. This means identifying and delivering the minimum functionality required for a product to start providing some value to the end users or business, then iterating it further to add more features later, as needed.

Agile acknowledges that requirements will change throughout delivery. The delivery team learn more about the service from user feedback and have the flexibility to adjust requirements and reprioritise deliveries quickly in line with user need. Agile approaches are often used for the delivery of digital services. Typically, they involve greater levels of collaboration within the team, as well as with the customer, than waterfall delivery methods.

The most common agile management framework is 'Scrum'. Agile teams in HMRC use Scrum techniques as well as other disciplines, such as Kanban and XP. This paper uses some of the more common Scrum terminology, such as 'scrum master' and 'scrum of scrums', though we recognise that all agile practices have their own benefits.

### **3.3.2 Agile roles**

Agile methodology recognises three roles:

1. **Product owner:** the product owner is a critical role. They have day-to-day responsibility for the product being developed by the scrum team. They manage and prioritise the product backlog and turn user stories into product features. They should be empowered to own and make all key decisions. This includes not just the development of the service, but its operability. The product owner is responsible for the ongoing operation of a high quality service, aiming to deliver as much value as possible.
  - o Note: in the UK public sector, the role of product owner is often split between a **digital service manager (DSM)**, with ownership of multiple digital deliveries, and **product managers** with operational responsibility for one digital delivery team. In these cases, the Product Manager performs the day-to-day responsibilities of a product owner, such as backlog management and prioritisation, with the DSM retaining ownership of the overall service vision, budget and resources.<sup>6</sup>
2. **Scrum master:** the scrum master helps the scrum team work in an agile manner and removes any impediments to progress. They act as the facilitator of the team, rather than a team lead, as the team is self-organising. The scrum master coaches the product owner, the team and any other business stakeholder groups on agile.
3. **Team:** the team are cross-functional and self-organising. They have all the skills and expertise required to build a "potentially shippable product"<sup>7</sup> as directed by the product owner. Key skills include user research, business analysis, design, build and test, though the size and composition of the team will vary depending on the project requirements. Although there shouldn't be any outward distinction between the different team roles and responsibilities, it's only natural that some team members will be specialists in one area more than others. The team should be set up to provide the right mixture of skills.

See section 4.8 for more information on the team composition.

### **3.3.3 Agile working practices**

Agile projects are delivered using a series of repeatable working practices that give a predictable structure for delivery. These are outlined below:

---

<sup>6</sup> Throughout this paper we refer to the product owner rather than DSM/product manager, though on occasion key issues and considerations for DSMs are highlighted where applicable.

<sup>7</sup> The team try to build 'potentially shippable products' in each sprint, such as outcomes (typically software) that are completed to a high degree of confidence and represent good quality work that is potentially shippable to end customers at the end of a sprint.

Practice	Duration	Description
Sprint	1-4 weeks duration	A time-constrained cycle of development. The scrum team iteratively develops potentially shippable slices of the overall product, according to the top priority user stories in the product backlog.
Release planning	Once per release	The release plan identifies the goal of a release, the overall release features, functionality and any potential obstacles. It also establishes an estimated release date and cost. The release plan can be updated on a sprint-by-sprint basis.
Sprint planning	Once per Sprint	Plan <b>what</b> functionality is going to be developed during the sprint, based on the highest priority user stories in the product backlog. A sprint goal is created, which is an objective based on user stories. The team then plans <b>how</b> they'll convert user stories into working software by identifying tasks in a sprint backlog.
Daily stand-up	Daily	A daily team checkpoint during which each team member briefly explains: <ol style="list-style-type: none"> <li>1. What they've done since the last stand-up</li> <li>2. What they're going to do before the next stand-up</li> <li>3. What obstacles (if any) are in their way?</li> </ol>
Show and tell / Sprint review	Once per Sprint	The team present the functionality that they've produced in the sprint to the product owner and other stakeholders and invite questions. The product owner verbally agrees which user stories have been completed and revises the product backlog to re-prioritise the remainder.
Sprint retrospective	Once per Sprint	An internal team meeting where the previous sprint is evaluated in terms of people, relationships, processes and tools. It identifies areas that went well and areas for improvement, turning these into actions for the next sprint.

You can find out more information on agile at [GOV.UK](https://www.gov.uk).

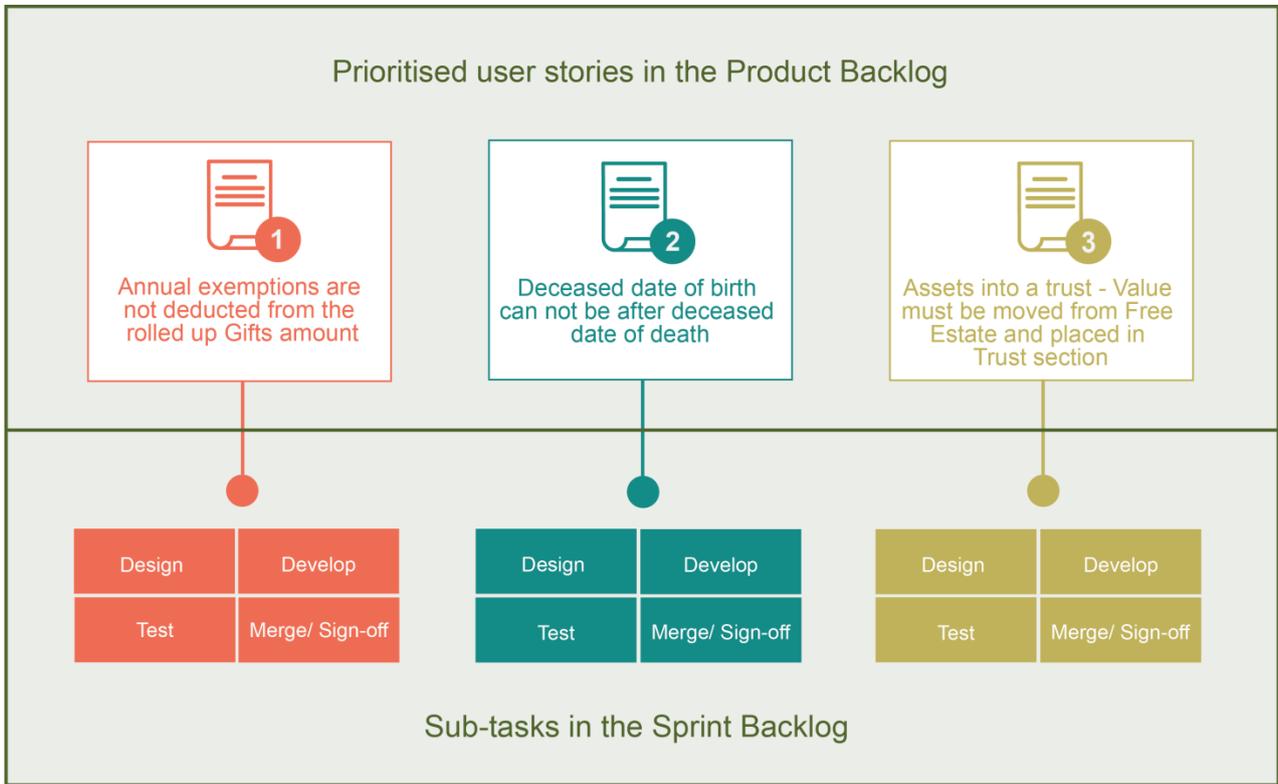
### 3.3.4 Agile artefacts

Agile recognises four key delivery artefacts that support development. These are:

- 1) **Product backlog:** The product backlog contains all outstanding requirements for the product, such as user stories and high-level user epics<sup>8</sup>. The product backlog is dynamic. As key stakeholders' requirements change, new user needs are discovered, or unforeseen technical challenges arise, user stories can be added to, removed from or re-prioritised in the product backlog. The product backlog is created during Discovery. It is managed throughout the agile lifecycle by the product owner, with support from business analysts.
- 2) **Sprint backlog:** The sprint backlog contains all committed user stories for the current sprint, broken down into smaller, more manageable tasks (see Figure 4). The sprint backlog is developed by the whole team in the sprint planning meeting. It contains all tasks which are essential for turning the user requirements into an increment of a potentially shippable product.

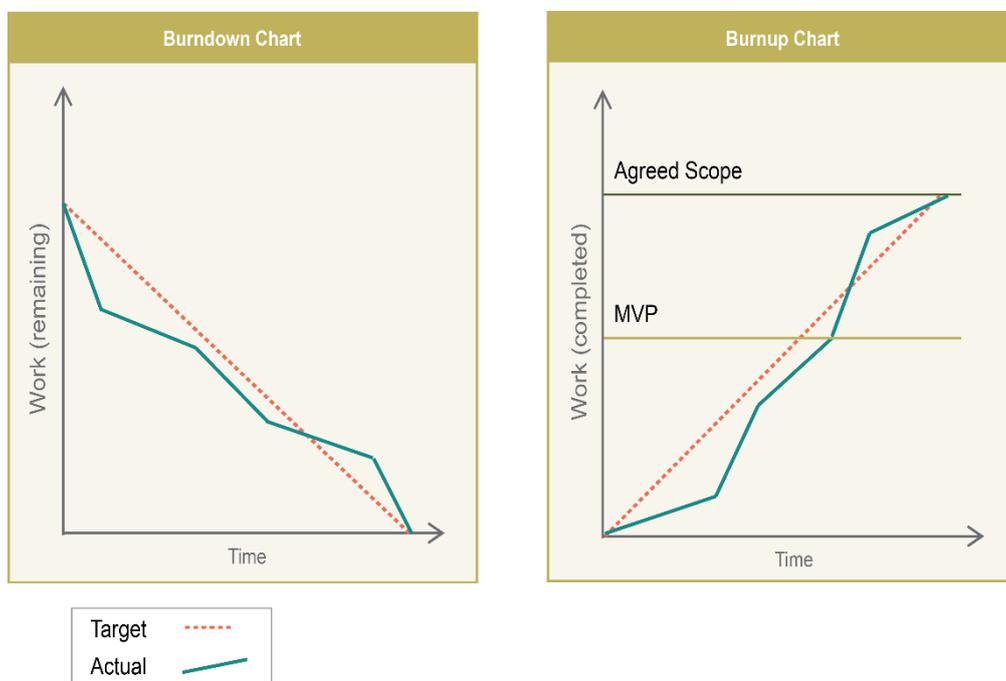
---

<sup>8</sup> Epics are essentially large user stories that capture a significant amount of work and can be broken down into a series of smaller user stories.



**Figure 4: Translation of product backlog user stories into sprint backlog sub-tasks**

- 3) **Burn down chart:** The Burn down chart is used to track team progress. It shows the amount of work **remaining** in a sprint, release or project. There are two lines – one projected and one actual progress line (see Figure 5). The burn down chart helps to measure the progress of the team against timelines planned in the sprint or release planning meetings, and is useful in helping teams visualise the end goal.



**Figure 5: Burn down and burn up charts**

- 4) **Burn up chart:** The burn up chart is also used to track team progress. It shows the amount of work **completed** in a sprint, release or project. There are a number of lines – an agreed scope line (and possibly a MVP line, lower than agreed scope), a target progress line, and an actual progress line. The burn up chart helps teams understand their progress towards an agreed scope, and is particularly useful in visualising changes in agreed scope.

Further information on the practical use of burn up and burn down charts can be found in section 4.6.

### 3.3.5 Agile at scale

At first glance, agile development can appear to be suitable only for small projects, but this isn't the case. Scrum was designed to be intrinsically scalable and can be used to successfully support and structure the project activities of several hundred developers or multiple inter-dependent teams.

Scaling agile for large projects can be achieved by grouping agile team members from similar backgrounds into appropriate teams, performing similar functions. These extend across the organisation, such as (but not limited to):

- The product owner community
- The scrum master community
- The UX community (service designers, interaction designers, visual designers)
- The scrum-of-scrums (the best-informed agile team members)
- The continuous improvement team (cross-functional specialists seeking improvements to delivery)

All of these local or virtual communities communicate regularly and meet as often as required – in some cases more than once a week – to perform their function effectively.

The product owner community allows an organisation to adopt a scale-by-value delivery model to the product or service. A lead product owner collaboratively sets the vision for the overall value stream. The individual product owners take responsibility for separating and refining the backlog items. Their respective scrum teams will then deliver these items into the overall solution.

The scrum-of-scrums is accountable for bringing together cross-team solutions. It meets to align team activities for successful integration at the end of each sprint.

A continuous improvement team provides a forum for cross-functional specialists, or leads, to share their experiences and ideas about issues they've identified, effective mitigations and performance improvements.

If you only have one scrum team to support, strong executive sponsorship is often a key driver for success. The team needs support and protection given they're likely to deviate significantly from accepted and established delivery processes and governance (for example, waterfall delivery).

When an organisation decides to scale agile, then these new delivery processes and ways of working need to become the norm. Senior leadership support and dedicated operational management, such as digital delivery management and continuous improvement teams, are crucial for making the necessary changes to how the organisation will operate in future.

## 3.4 Digital delivery progress in HMRC

Much of the content in this document is based on HMRC's experience of setting up and operating five DDCs in London, Newcastle, Worthing, Telford and Yorkshire.

HMRC's digital strategy, first published in December 2012<sup>9</sup>, underpins its digital vision: *"Our ambition is to deliver a transparent tax system that encourages voluntary compliance, enabled by customer-focused digital services which are so straightforward and convenient that all who can use them will choose to do so"*. HMRC is embedding digital delivery at the heart of its business to achieve its three strategic objectives: improving customer experience, reducing cost and maximising yield. Given HMRC is responsible for more than two-thirds of all government transactions<sup>10</sup>, its digital transformation is a key component of the wider government digital agenda.

---

<sup>9</sup> <https://www.gov.uk/government/publications/hmrc-digital-strategy-2014>

<sup>10</sup> <https://www.gov.uk/performance/services>

In spring 2013, it was announced in the UK Spending Review that HMRC would invest £200m in a three-year digital investment programme with further investment expected beyond this three year period. This investment has led to the development of the Multi-Channel Digital Tax Platform (MDTP) where all HMRC digital services are implemented. It also led to the mobilisation and operation of HMRC's five DDCs, responsible for delivering the department's digital strategy.

From 2013 to November 2015, HMRC released 26 new digital services on the MDTP through its DDCs. HMRC currently has over 35 Digital Delivery Teams in operation across its five DDCs.

As HMRC has scaled its digital delivery capability, the proportion of HMRC staff compared to suppliers has increased. This has been achieved by a structured capability development programme, knowledge transfer from suppliers to HMRC, and the recruitment of digital practitioners (see section 4.7 for more information). Simultaneously, HMRC also worked with its suppliers to reduce the average day rate of supplier team members, in line with longer-term staffing commitments and increased internal capabilities.

Figure 6 demonstrates an **indicative** view of the ramp up of an HMRC DDC, gradual increase in the proportion of HMRC staff, and reduction in supplier day rates.

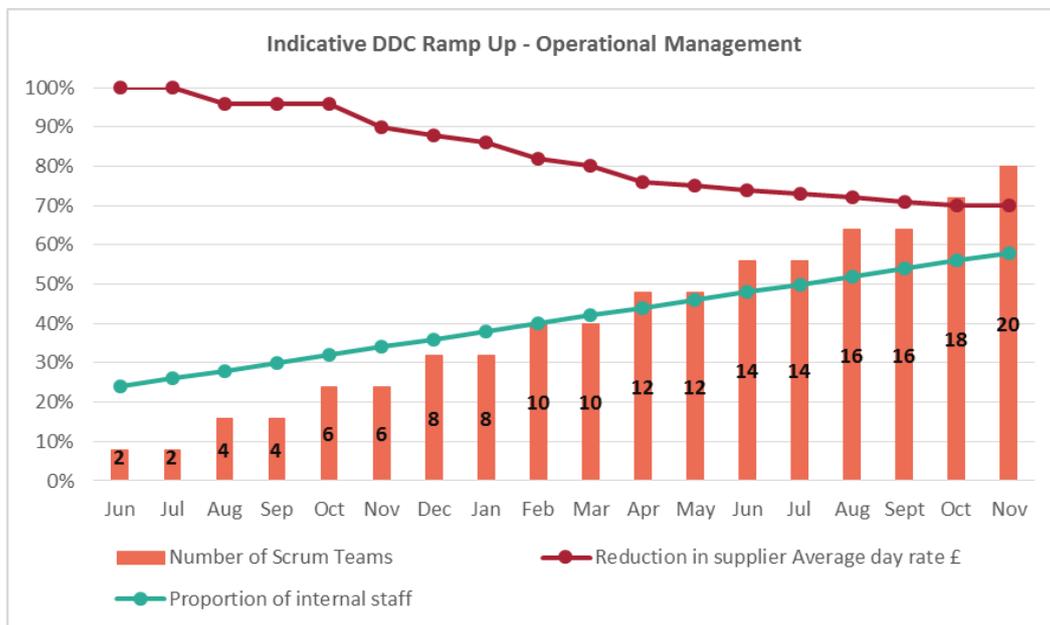


Figure 6: Indicative DDC ramp up and stabilisation

### 3.4.1 Digital Delivery Centre South Bank (DDCS)

The Digital Delivery Centre South Bank (DDCS) in London was the first DDC to be established by HMRC in 2013. Throughout 2013, HMRC developed four digital exemplar services using an agile delivery approach, established a digital WebOps capability and then launched the Multi-Channel Digital Tax Platform (MDTP) in February 2014.<sup>11</sup> This platform now hosts all digital services developed by HMRC digital delivery centres. DDCS is responsible for:

- Developing the MDTP
- Operating HMRC's WebOps team
- API delivery
- Transaction monitoring services
- Live Support for HMRC digital services.

It also has five scrum teams responsible for the delivery of individual digital services.

<sup>11</sup> <http://assets-production.govstore.service.gov.uk/G5/0348/5.G2.0348.001/QD5/Equal%20Experts%20Case%20Studies.pdf>

HMRC has partnered with Equal Experts to set up and operate the DDCS. Following a refit the centre was inaugurated in May 2015 and has grown significantly, with a headcount of over 250 at November 2015.

### **3.4.2 Digital Delivery Centre Newcastle (DDCN)**

Following the success of the DDCS, HMRC wanted to establish a second DDC in the North-East, where it already had a presence. HMRC wanted to take advantage of a growing, local, digital community while supporting significant job creation.

In November 2013, the department engaged Accenture to help establish the Digital Delivery Centre Newcastle (DDCN). Accenture worked with HMRC to put in place the operating model, people, facilities and plans needed for large-scale digital delivery in just five months. Simultaneously, joint HMRC-Accenture teams began digital service delivery during the mobilisation period, delivering new services while continuously improving the ways of working in the centre.

The DDCN officially opened in April 2014. It currently has 20 Scrum development teams, each responsible for delivery of an individual digital service. It shares responsibility with DDCS for providing dedicated live service support - on a rota basis - required for HMRC's digital services (see section 5.6 for more detail on HMRC's live service model).

As the Newcastle centre has grown, the proportion of HMRC staff to suppliers has increased. This is due to:

- the delivery of a range of capability development initiatives
- training HMRC apprentices and trainee team members
- knowledge transfer between the supplier and HMRC staff.

By November 2015, over 240 people were working in the DDCN.



**Figure 7: HMRC DDCN collaboration areas**

The teams at the DDCN have delivered a number of pioneering services. These include a new online tax credits renewal service, which allowed more than 750,000 people to renew their tax credits online in 2015 -

an increase of 94% on the previous year<sup>12</sup>. HMRC continues to invest heavily in the centre. A new floor has been designed and built ready for further expansion.

### **3.4.3 Digital Delivery Centre Worthing (DDCW)**

In May 2015, HMRC set up a new DDC in Worthing (DDCW). The DDCW sits within the wider HMRC digital delivery organisation, but also provides the opportunity to collaborate on digital delivery with the Valuation Office Agency (VOA). The VOA is now also using the MDTP to release its own digital services. Worthing was set up reusing the operating model, infrastructure and ways of working from the other HMRC DDCs. As HMRC's digital delivery capability matures, it opens the possibility of developing and operating services for other public sector organisations.

### **3.4.4 Digital Delivery Centre Telford (DDCT)**

HMRC further accelerated its digital transformation plans by partnering with Capgemini to open its third DDC in Telford (DDCT). Building on lessons learned from the first two DDCs, the team were able to mobilise the centre within three months. The official opening was in September 2015. The DDCT also provides new job opportunities in Telford.

The foundation for the DDCT's initial success has been down to the mix of knowledge - about HMRC estates and agile delivery - held on the team. This meant teams could quickly build upon the lessons learnt from the earlier DDCs and adapt their methods of working. The DDCT demonstrates how quickly centres can mobilise and become operational as HMRC's experience of digital delivery grows.

The Telford Centre established a structured approach to the mobilisation and induction of its initial teams. It has:

- Communicated a clear vision for the DDCT, made sure team members understand this vision and their role in HMRC's wider digital strategy
- Reused the operating model and delivery processes developed by the DDCN and DDCS to familiarise themselves with HMRC's approach to agile
- Reused the existing technical guidance to identify tools and understand the deployment pipeline process
- Provided role-specific technical training as recommended by existing teams
- Introduced mock sprint development using a tutorial application developed by the DDCN. This helped the teams learn how to use each of the key development tools and processes necessary to deliver new services on the MDTP. It also supported team forming
- Seconded team members from DDCS and DDCN teams for a sprint to better understand technologies and ways of working, and to establish key relationships.

By December 2015, five scrum teams were in place in Telford. With a number of potential projects in the pipeline, the DDCT will grow in 2016.

### **3.4.5 Digital Delivery Centre Yorkshire (DDCY)**

HMRC has also set up a DDC in Yorkshire (DDCY) to support the digital services developed at the other centres.

The DDCY provides a number of key functions:

- **APIs**<sup>13</sup> - modernisation of existing APIs and development of new ones in line with HMRC's API roadmap and strategy<sup>14</sup>. HMRC and Equal Experts partner within the API teams

---

<sup>12</sup> <https://www.gov.uk/government/news/94-increase-in-online-tax-credits-renewals>

<sup>13</sup> Application Programme Interfaces (APIs) are a set of routines, protocols, and tools for building software applications. APIs allow applications to access the features or data of an operating system, application, or other service. See [https://en.wikipedia.org/wiki/Application\\_programming\\_interface](https://en.wikipedia.org/wiki/Application_programming_interface)

- **Innovation, collaboration and engagement** - responsibility for conducting evaluations of proposed digital services prior to Discovery to establish a) whether there is a user need, and b) whether this can best be met by a **digital** service. Collaboration facilities have been set up to support business groups, DDC teams and third party software providers and to analyse customer insights
- **Analytics** - provide customer insight to the business through performance and social media analytics of digital services
- **First line support** - for HMRC digital services that are live to the public but still undergoing significant development (i.e. In the beta phase).

The centre has not been formally inaugurated.

### 3.4.6 Managing a multi-site delivery team

Each HMRC DDC is different. They have their own identities and subtly different ways of working. As HMRC’s digital capabilities grow, the centres must be able to work effectively together. Figure 8 shows HMRC’s digital delivery operating model. Each centre is loosely aligned to a different area of the business, though projects can be allocated to any centre based on business priority and capacity. HMRC has implemented cross-centre platform level functional leads in design, user research and architecture, to define and ensure consistent standards, tools, software, and methods.

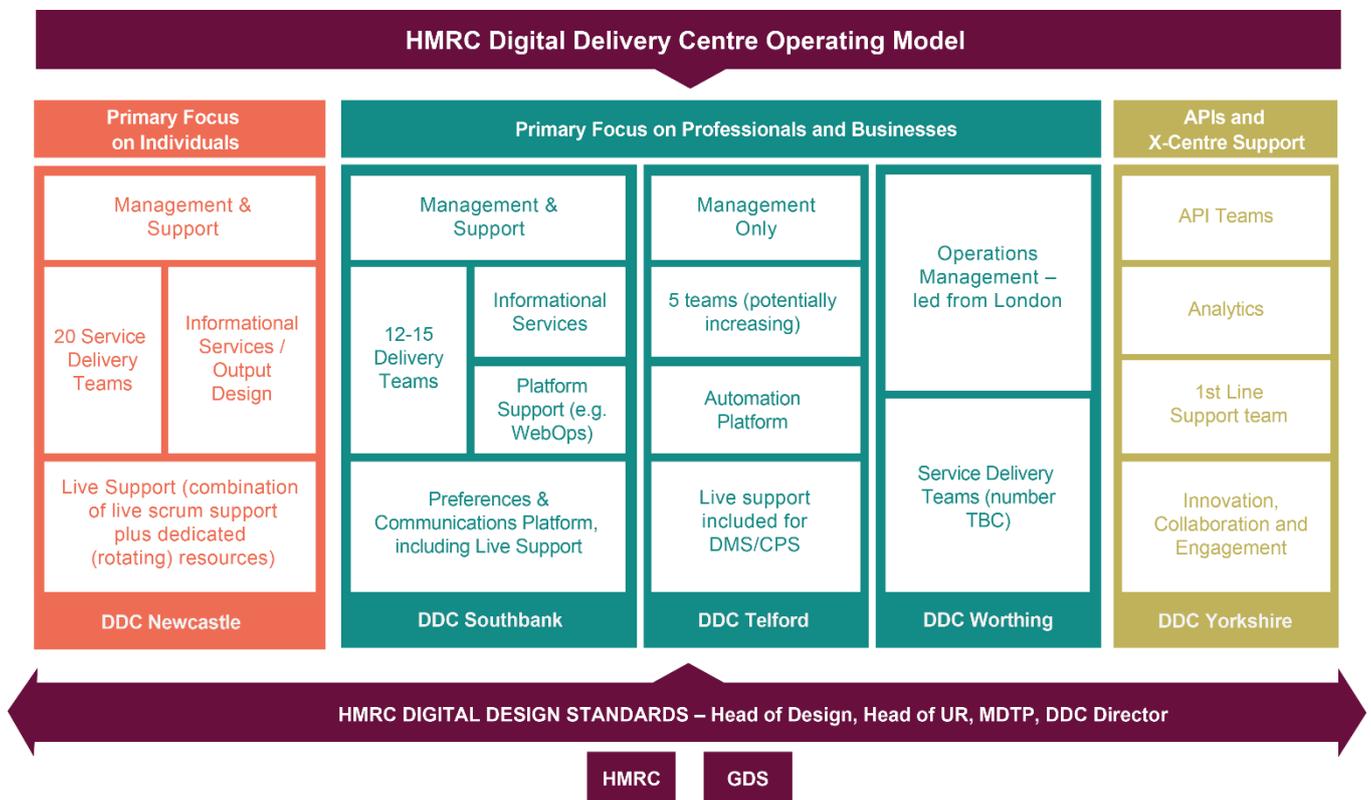


Figure 8: HMRC Digital Delivery Operating Model 2016

<sup>14</sup> <https://www.gov.uk/government/publications/hmrc-third-party-tax-software-and-API-strategy/hmrc-third-party-tax-software-and-application-programming-interface-API-strategy>

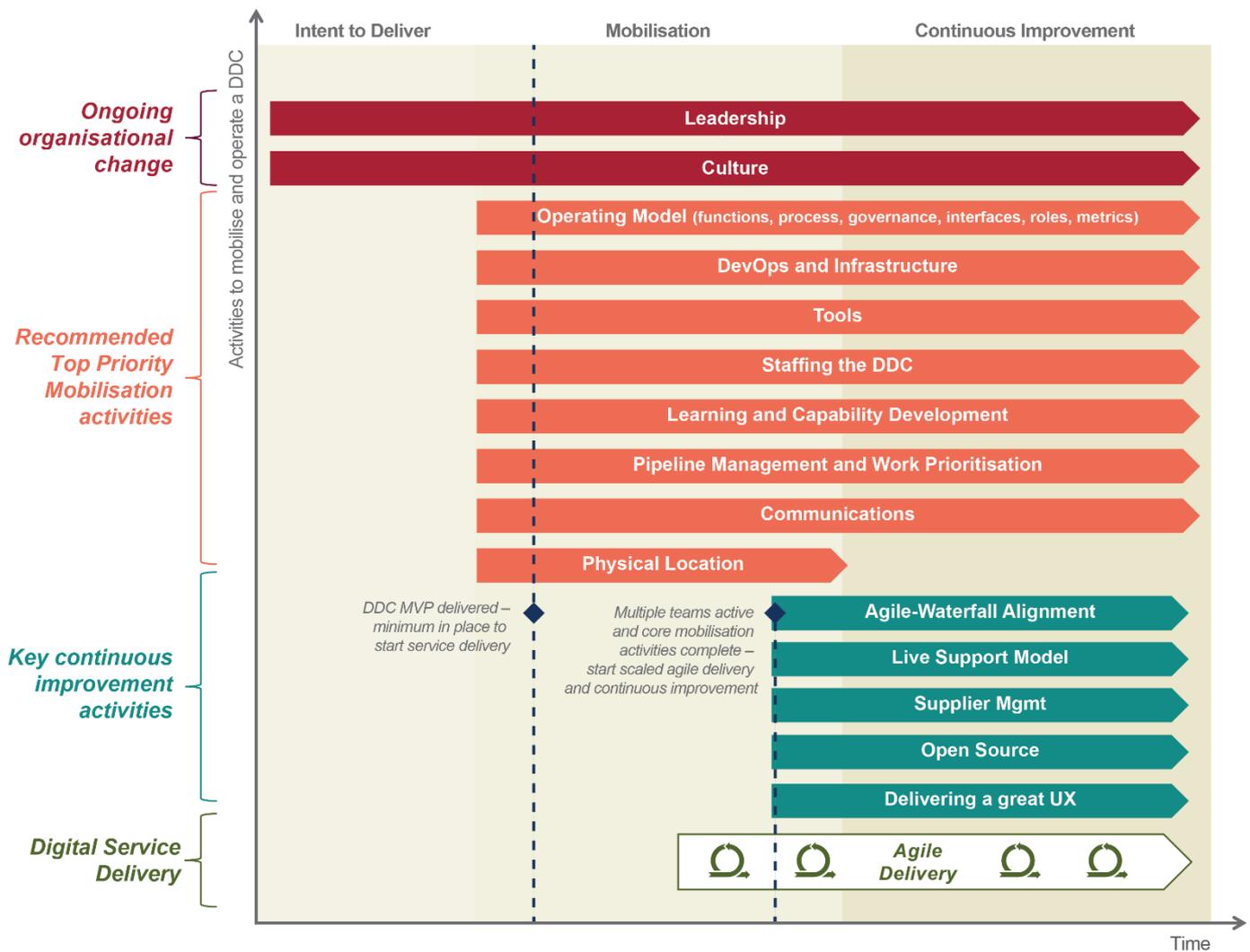
## 4.0 MOBILISING A DIGITAL DELIVERY CENTRE

Setting up a new DDC is a significant challenge for any organisation. Digital delivery changes the way that businesses operate internally and externally in their interaction with customers. To manage this change requires radically different processes, people, structures, technology, culture and more. Increasingly, public sector leaders want to set up their organisations to support digital delivery but knowing how to plan and deliver this change can be a challenge.

It is vital to instil a culture of continuous improvement as early as possible. This will help nurture a successful DDC. Start with a focused period of mobilisation that aims to deliver the centre's MVP, which is the minimum DDC components required to commence digital service delivery. Once these are in place, install the first scrum teams and learn from doing. Simultaneously continue with the priority mobilisation activities to prepare the DDC for scaled agile delivery, whilst ramping up the teams. Once the core DDC capabilities are in place, continue working iteratively to improve them. At the same time address the next tier of activities needed to effectively operate the DDC.

This approach, requires strong enterprise and operational level leadership and considerable cultural change to the organisation at every level. These leadership and cultural changes are big, ongoing activities that will continue to be important as the DDC matures. Our experience demonstrates that you have to involve all team members in the development of your centre. All levels of the team are responsible for shaping the centre's culture and the way that it operates. Management cannot foster a successful digital delivery culture on their own. This paper highlights some recommended priority mobilisation and continuous improvement activities to focus on. However, every organisation is different and the underlying detail within these areas, and the order in which they are addressed, will vary depending upon the organisation and its people.

Figure 9 indicates an example approach for mobilising and operating a DDC.



There are multiple components in each horizontal work strand in the diagram. Some are needed as part of the DDC MVP to start delivery, whereas others are lower priority and can be addressed in parallel to team ramp up and delivery.

**Figure 9: Example DDC Mobilisation Methodology**

This chapter outlines the key activities to consider in order to mobilise and operate a large-scale DDC. We intend to provide practical guidance that can direct and support you to transform your business to deliver digital services. We also highlight how to manage the DDC in its early stages of operation.

## 4.1 Leadership

Strong leadership is necessary to successfully implement the DDC. It is also important to support the wider changes that digital delivery brings to organisations, such as how employees interact both with each other, and their customers. A clear vision of the DDC and its impact should be communicated to all levels of the organisation. Securing buy-in from senior and middle management is a priority. Leadership in a digital age requires new capabilities, such as increased responsibility for handling data, and new behaviours - for example, Digital Leaders have to be active and visible on digital channels such as social media. This section outlines the leadership characteristics and activities required to effectively establish a DDC. In addition, it also details the operational leadership needed to deliver an effective DDC mobilisation and then effectively operate the DDC.

### 4.1.1 Senior digital leaders

Developing a DDC requires sponsorship and buy-in from senior leaders within the organisation from the outset. They must secure the necessary funding and visibly champion a clear vision for the DDC externally

and internally. Everyone involved needs to understand their role in achieving the digital objectives. Employing digital leaders who have experience leading (digital) business transformations, and/or large scale digital delivery will help build enthusiasm. They will help supplement the subject matter expertise of the existing organisation and increase the pace of change. Not only should they be good leaders, but also passionate and knowledgeable about digital, and empowered to make the changes to the organisation's operating model, technology estate, people and services that digital delivery will require. They need to secure buy-in and cooperation from other leaders across the organisation who will be affected.

Digital leaders need to both understand the effect of digital, and act as role models to drive new behaviours in the organisation. They must lead by example. They must engage digitally with colleagues, customers, agents and suppliers and have an inspirational online presence. They should help the digital workplace culture evolve, with collaboration, mobile and virtual working and evolution of working practices, for example through gamification and social media crowdsourcing.

Mobilising a DDC in isolation will limit the benefits of digital delivery. Leaders need to consider what other changes are needed throughout the remainder of your organisation to embed digital operations, support the new digital services and become a truly "digital enterprise". Digital customers expect a consistent and joined-up experience, and your organisation should be organised to meet this expectation. For example, HMRC is moving many existing services to the cloud, modernising and consolidating its IT estate, developing digital contact centre services, for example web chat, and transitioning to agile delivery on many of its legacy systems. DDCs are a key component of any digital enterprise, but they must be supported with change throughout the rest of the organisation. Effective digital leadership is essential to support these high levels of change.

#### **4.1.2 Developing digital leaders**

Aside from a lack of knowledge surrounding digital or agile working practices, culture is often cited as a prime reason for the failure of agile projects, and governance of such projects is often seen as challenging. Our experience has demonstrated that training the leadership is an important priority.

Leaders need to consider the new operating models required to serve the digital customer. Effective leadership and focused communication will help colleagues, customers, agents and suppliers achieve the right levels of awareness, understanding and buy-in to change and adopt desired digital behaviours. Industry and government feedback shows that new digital workplace leaders and employees want more training to do their job effectively and keep up with new customer demands. Leadership in a digital age requires new capabilities. They have increased responsibility for handling data. They make analytics-based decisions. They work with flatter control structures and provide content leadership across the organisation. There needs to be a focus on what this means for each organisation. A Digital Leadership Programme can help with this, focusing on:

- **Inspire** (Why should we?)
- **Learn** (What should we?)
- **Apply** (How should we?).

Leaders need to consider the level of change they want and need. They should ask "*How agile do we want to be?*" and then "*How can we take our existing situation and make it agile?*" If the cultural environment remains incompatible with digital or agile, or the delivery process itself is not adaptive, then delivery will still be at risk.

#### **4.1.3 Starting digital delivery**

Achieving real and demonstrable progress will underline commitment and build momentum. You can iteratively roll-out components of the DDC – starting with an initial dedicated DDC mobilisation team responsible for leading the setup of the DDC. The mobilisation team should be swiftly followed by one or two experienced exemplar scrum teams. You can run mobilisation activities in parallel to delivery from the exemplar teams, gradually rolling out and enhancing the DDC operating model, increasing the number of teams, building the technology stack and target infrastructure, whilst continually improving through lessons learned and refining longer-term plans.

The mobilisation team should be responsible for providing the initial operational level leadership necessary to enable the DDC to start delivering. They should also support the initial teams as the DDC ramps up. You should involve a mixture of experienced internal employees, new recruits (potentially in leadership positions) and suppliers or contractors with digital experience in the mobilisation team. These 'externals' can supplement the necessary internal subject matter expertise and bring the freshness needed to deliver the necessary level of change. The mobilisation team should collectively possess:

- experience of the organisation's processes, organisation design, governance and personnel
- knowledge of the legacy technology estate
- experience of the digital development technology stack
- experience of digital operations, including DevOps / WebOps
- experience in (digital) delivery and change
- delivery management experience
- agile delivery experience
- business transformation experience including operating model design, process optimisation, capability development and change management

Focus on delivering the DDC MVP, i.e. the core mobilisation activities needed to enable teams to start delivering. Thereafter, you should continue to run the mobilisation activities needed to effectively operate large scale digital delivery. Continuously improve ways of working to streamline and increase effectiveness of delivery. Once the delivery teams join the DDC, involve them in the mobilisation and continuous improvement of the DDC. People need to feel part of the centre. They need to assume a sense of ownership for the DDC 'product' that is being developed.

You have to demonstrate that digital delivery is different to existing regimes. This will help to increase acceptance and adoption of the change. It is critical to build an office environment that supports collaboration. You should also invest in new tools and facilities and establish governance and delivery processes that support agility. This need not be dependent on the availability of a new delivery location. Practical changes can be made quickly by seating teams together, rearranging existing desks to encourage collaboration, printing out and using agile metrics, and deploying key agile ceremonies e.g. sprint planning, and daily stand-ups.

Key strands of Mobilisation include:

Work Strand	Description
Operating Model: Functions, Process, Governance, Organisation Structure, Metrics, Interfaces	Define how the DDC will be organised to meet its business goals. This involves the following components: <ul style="list-style-type: none"> <li>● Functions</li> <li>● Process</li> <li>● Governance</li> <li>● Organisation Structure / Roles</li> <li>● Metrics</li> <li>● Interfaces.</li> </ul>
Recruitment	Identify, attract and recruit the right people to fill the open roles at the DDC, using a mix of internal staff, new recruits and suppliers.
Learning and Capability Development	Develop and implement a structured approach to on-boarding, monitoring and building key skills and behaviours.
DevOps	Define and implement the development approach, architecture, infrastructure and tooling to support streamlined delivery of digital services.
Tooling	Select and implement tooling required at all stages of the delivery lifecycle.
Physical Location	Design and build an engaging work space that encourages high levels of team interaction and joint-working.
Communications	Deliver a communications strategy that engages internal and external stakeholders that the DDC needs to involve to succeed.

As the DDC increases in size, team members from the exemplar teams should be moved into new teams. They can share their experience of ways of working and best practice. Agile recommends maintaining stable teams, but this principle is more applicable at steady state; placing people with the right experience in new teams helps the new teams to form and upskill quickly, while also effectively scaling delivery capabilities in a consistent manner.

Following the mobilisation phase we recommend establishing a DDC-level continuous improvement team. Embedding a large-scale new digital delivery capability in an organisation involves significant levels of change. A wide range of common issues impacting delivery will arise across the teams. The continuous improvement team can work across the teams to embed the new ways of working, identify issues impacting delivery and design, and implement mitigating solutions in order to stabilise and improve the centre. However, as with mobilisation, each individual in the DDC should take responsibility for continuously improving the centre in order to develop a collaborative culture and DDC that best suits the people within its teams. See Section 5.1 for more information on the scope and composition of the continuous improvement team.

#### **4.1.4 Operational delivery leadership requirements**

The mobilisation team are responsible for leading the DDC mobilisation. The product owners are responsible for leading the service delivery. In addition, you will need to have in place operational managers responsible for the delivery and day-to-day operation of the DDC. We recommend having the following leaders to manage deliveries across multiple teams:

- **DDC lead**
- **Operations managers**
- **End-to-end test management**
- **Continuous improvement leads** - works across the teams to embed the new ways of working, identify issues impacting delivery and design and implement mitigating solutions.
- **Functional leads** - for example scrum master, design, user research and architecture, to lead communities and define and ensure consistent standards, tools, and methods in their specialist area; these leaders operate as part of the continuous improvement team on either a DDC-wide or platform/cross-DDC level.

These people sit outside of the individual teams, and support their specialist area across all the individual teams.

## **4.2 Culture**

Digital delivery is as much a cultural change as it is a change in technology or delivery methodology. For many in the DDC itself, as well as the wider organisation, it is a new way of working. This section outlines some of the key activities required to embed digital delivery within the organisation, and highlights some specific challenges to address.

### **4.2.1 Alignment of business and end user requirements**

Chapter 3 stated that delivering digital services to meet user needs will benefit both customers and public sector agencies, strengthening the relationship between the two. Departments need to balance the needs of their customers with the needs of the business, and as soon as possible align the two. Alignment is particularly challenging where agile deliveries (requirements defined by users) and waterfall deliveries (requirements defined by the business) co-exist. The backlogs between these teams will be radically different: one with requirements previously not considered (agile, customer-focused), and the other with requirements which may no longer be needed (waterfall, business-focused). Organisations need to expand the customer focus of digital deliveries to their legacy services. What do customers need from this service? How can business policy align with that user need? The increase in UX specialists and user research facilities that DDCs bring can be used to increase policy innovation and responsiveness in line with user expectations.

## 4.2.2 Fail fast and cheap – the iterative release of public services

To achieve the agile benefits of risk reduction, time-to-market, and fit-for-purpose delivery it is important to iterate the digital service, rather than merely adding new functionality upon each release. GDS specifically ask for “evidence of iteration” that shows the product or service has changed direction by applying learnings from previous releases. Projects that deliver monolithic epics of functionality in a serial manner risk delivery failure should the project over-run. Instead, you should strive to release the MVP to a subset of users. The MVP will form the basis for all further releases. Each subsequent release, should strive to add significant value, rather than incremental detail. In this manner, when the project’s delivery date arrives, the only variable is how “refined” the product is – not whether it has successfully delivered a cohesive product release. Involving stakeholders from across the business and technology in the development process will help build confidence in the value of releasing products before all requirements have been met (e.g. within the team, and attendance at ‘show and tells’).

HMRC has moved from a model of 2 Enterprise Releases each year, to over 50 releases on the MDTP every week

Key to this approach is the product owner. They must be empowered to define the MVP and display a commitment to releasing the service early to start delivering value. The question for product owners, and over time the whole organisation, should not be “*What can we deliver by point X, many months in the future?*”, but rather “*How long will it take us to release an MVP that delivers some value?*” Similarly, releasing the service is not an outcome in itself, rather the product owner needs to consider the operability of the service and must strive to deliver a sustainable and continuously improving service that provides value on an ongoing basis. This repositions the delivery approach, with value the sole objective.

Within the team, more regular releases mean that services become ‘live’ earlier in the delivery lifecycle. Live service support arrangements need to be planned accordingly. Teams need to develop the mind-set that they own the live service in Beta and must lead all monitoring and continuous improvement activities. A central DevOps team may be in place to manage and oversee the live platform, but individual services should be monitored by their development teams, with team activity and resources planned accordingly.

Figure 10 shows how teams may divide work across releases to deliver priority functionality (story maps in this case) across a number of releases. Note that release 1 does not involve the full application scope, but that it is iteratively added in releases 2, 3 and 4 to provide full coverage.

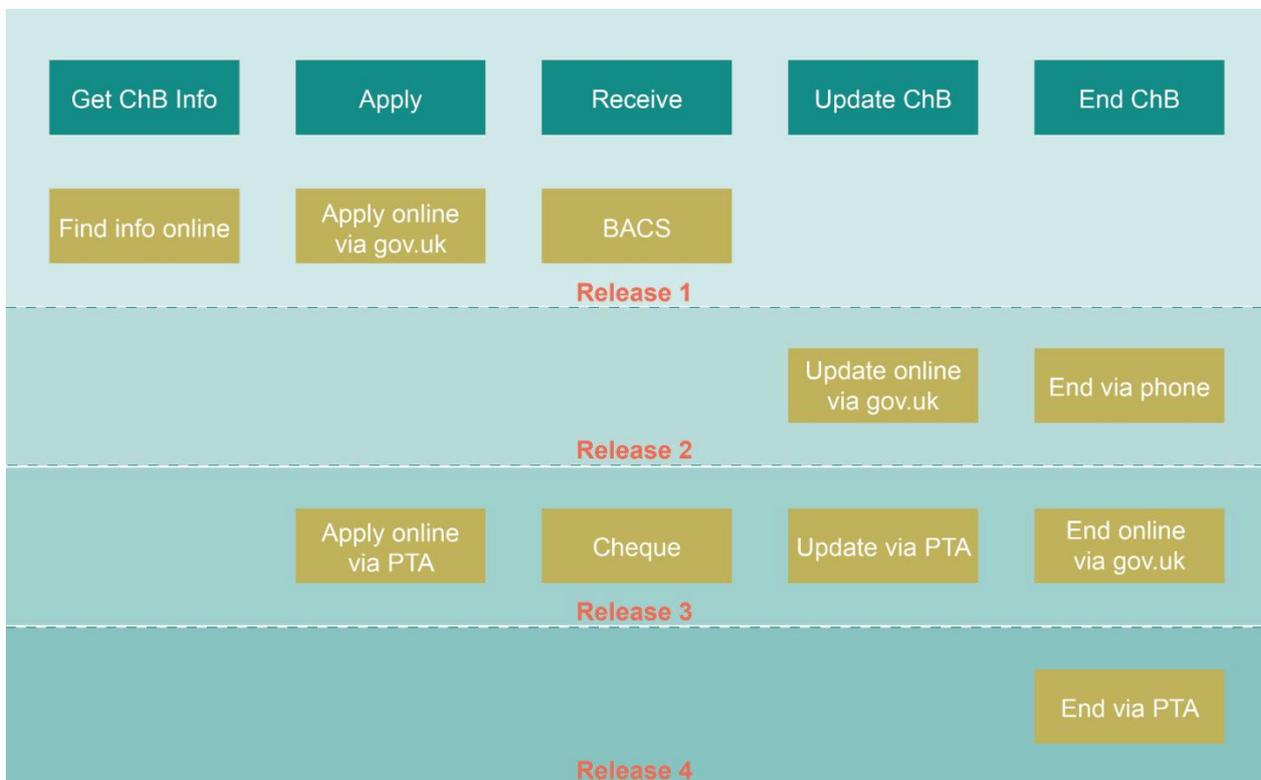


Figure 10: Iterative release of MVP

### 4.2.3 A collaborative and continuously improving working culture

DDC teams operate differently to traditional IT delivery teams. They should transcend job-title boundaries and work **as a team** on all of the activities needed during development. This means that software engineers perform user research and UX specialists assist with coding activities. Start-ups work this way, they have to, and it is crucial that DDCs learn to work this way to foster the cross-functional and collaborative working necessary for agile delivery. The DDC should provide the tools the team need to instil this culture, including technology, facilities, processes and training. Within the DDC you should give teams autonomy to let them establish which working practices best suit their team members, for example Scrum, Kanban etc.

We outlined the importance of iterating a digital service above. It is equally important to iterate the DDC delivery processes and ways of working. Fostering a start-up mentality, with team members assuming a sense of ownership for the progress of the DDC will help embed this commitment to continuous improvement. On a practical level, you can develop this mind-set by allocating mobilisation activities to the teams and communities within the DDC, encouraging suggestions and solutions to common problems, establishing information sharing events and scrum of scrums, and including DDC mobilisation within performance objectives of team members. This will help to organically grow the desired collaborative and continuously improving working culture.

Scrum masters should be held accountable for delivering an increasing velocity. This means the average amount of work completed per sprint or iteration. The productivity benefits a scrum master produces should more than cover the cost of resourcing the scrum master role. This does not mean that the scrum master should overburden the team, rather that the scrum master should constantly ask themselves the question: *“How can this team deliver faster?”* They should seek guidance from the team on how this can be achieved in the form of improved working practices, automation, and the re-use of solution components from both within the team and other sources.

Finally, digital delivery enables greater alignment and interaction between the business and technology than in legacy organisations. Rather than upfront business requirements development, service requirements are developed iteratively in collaboration between the development teams, business experts, internal operations, and external customers. Operational business experts need to be involved throughout service development, working with development teams to continually prioritise and agree service requirements. Digital delivery is intended to enable the business vision to be realised. The active participation of business leaders during delivery is a prerequisite. You should include representatives from both the business and technology within the scrum team. They need a shared understanding that they are both responsible for delivering a high-quality user-focused service.

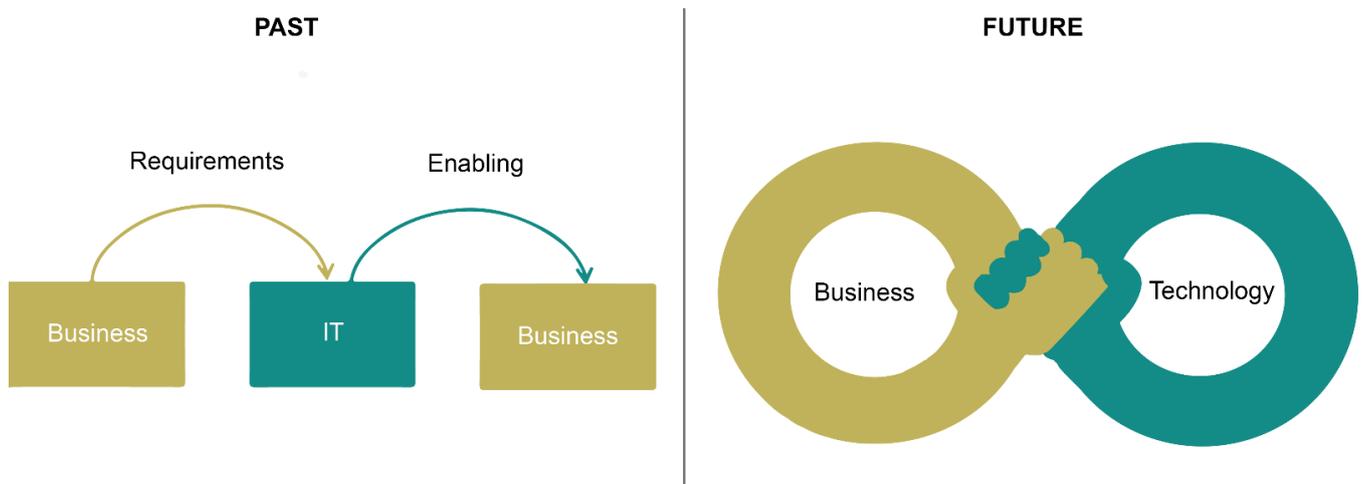


Figure 11: Business and Technology Alignment

### 4.2.4 Building knowledge of digital working practices

DDCs need to collaborate and communicate effectively with other teams in the organisation. You should involve leaders and back office staff throughout mobilisation. For leadership, this could mean visits to the

centre and attendance at show and tells for new services, though it could also extend to digital leadership courses. For back-office staff, advertising prototypes of new services on the department's intranet can allow these individuals to comment on emerging services, provide their knowledge and insight, and become engaged in the development process.

At an operational level, this can be achieved by communicating to key stakeholders the DDC operating model components as they are iteratively developed during mobilisation. Business leaders should participate in the early stages of project delivery. We recommend holding inception events to detail the approach to delivery and the respective roles and responsibilities of all stakeholders. You can also involve leaders in the development of the digital delivery operating model and run set-piece sessions to share the operating model and answer questions.

### 4.3 Operating model functions

One of the key features of a successful DDC is a well-defined operating model that provides a blueprint for the delivery of services. An effective operating model provides clarity on roles and responsibilities and enables full integration of people, process, technology and governance.

Digital delivery is driving important changes to traditional IT operating models. First, the IT architecture must evolve to support the new delivery model and processes. Second, both the DDC and extended workforce must adopt new skills - ranging from technical expertise to relationship building - to support the new model. Finally, new governance policies are required to establish priorities for innovation and to manage both internal and external stakeholders. The digital delivery operating model should contain the following components:

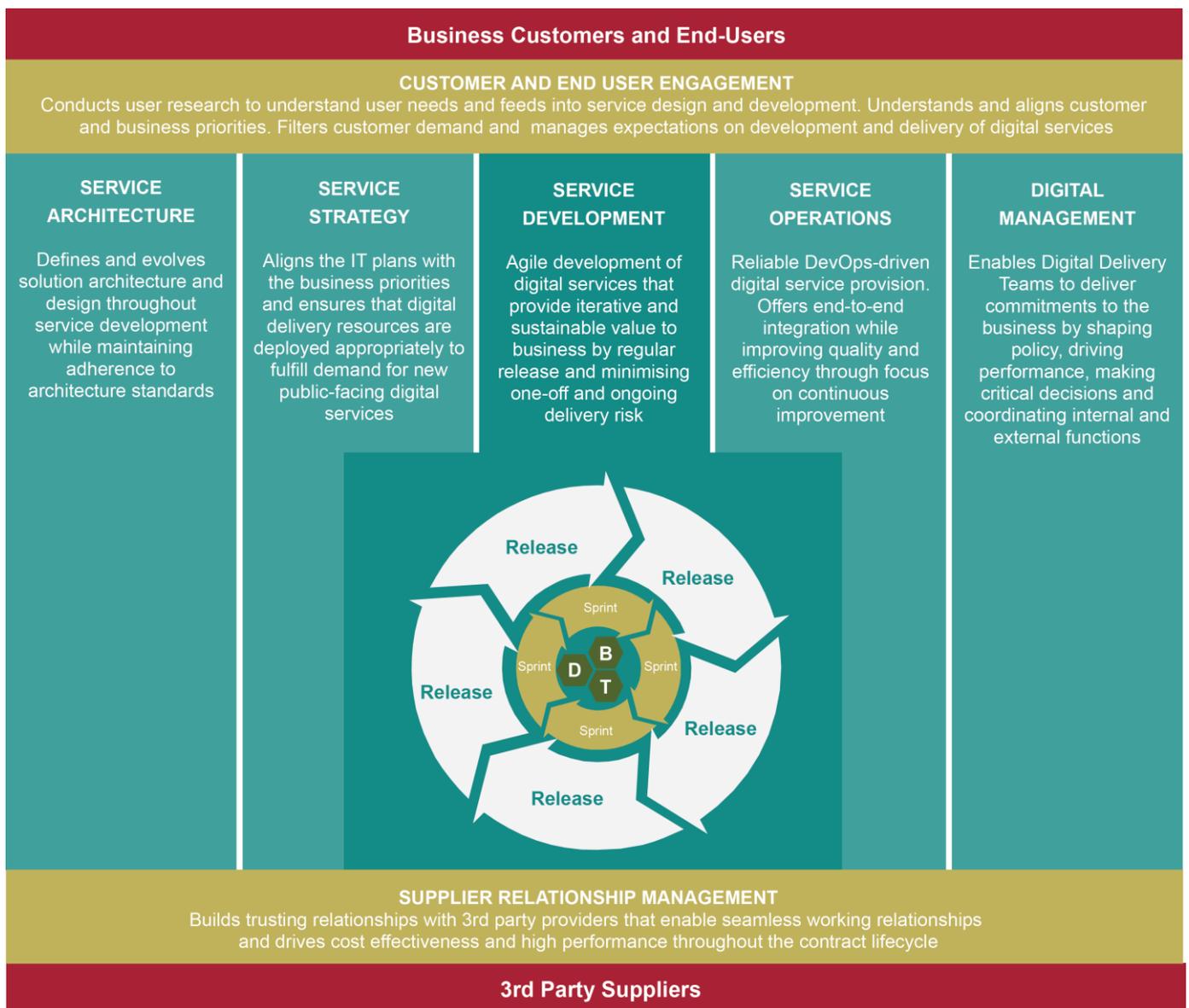
1. **Functions** – building blocks of the operating model, defining how work is organised to deliver services.
2. **Processes** – coordinated sets of activities that enable delivery of particular business outcomes required by a function.
3. **Organisation structure and roles** – define who is accountable for doing the work.
4. **Governance** – define how to make, sponsor and enforce the right decisions.
5. **Interfaces** – describe how to interact to deliver consistent services.
6. **Performance metrics** – provide the insights needed to deliver excellent digital services.

This section outlines the key functions of a typical DDC, with the other operating model components explored in the following sections.

#### 4.3.1 Digital Delivery Centre operating model functions

The first step in planning how a new DDC will operate is to define its functions: *What does it need to do to achieve its goals?* Both the processes used within the DDC, and the roles needed to perform the processes will be driven by the DDC functions. Digital delivery operating models involve seven functional areas:

- Customer and End User Engagement
- Service Strategy
- Service Architecture
- Service Development
- Service Operations
- Digital Management and Administration
- Supplier Relationship Management



**Figure 12: Digital Delivery Centre functions**

Within each of these functional areas there are a series of lower level functions, encompassing the full scope of digital delivery. These functions are driven by a set of processes (see section 4.4).

When designing the DDC operating model, you should consider which functions are most needed for successful digital delivery in each of the functional areas above. There will be some variance between organisations. For example, some organisations will have dedicated functions for innovation or supplier management, whereas others may not, depending on the priorities and capabilities already existing within the organisation. During mobilisation, definition of the operating model functions is an important activity.

Functions can be self-contained within the DDC or provided elsewhere in the organisation. This will vary between different organisations. Including a function within the DDC will depend upon:

1. Which functions currently exist within your organisation
2. Their current suitability for digital delivery (alignment to agile processes and timelines).

Many of the functions required for digital delivery are likely to already exist to some extent within other parts of the organisation. For example, solution architecture (architecture functional area), project management (management functional area), and commercial (supplier management functional area) will already exist in most organisations. You should assess whether these teams can perform the function adequately for the DDC. If not, we recommend either changing the existing organisation, or establishing a new dedicated function for the DDC.

### 4.3.2 Initial functions vs. advanced functions

Consider whether your existing organisation can provide each required function, or whether a new capability is needed. We recommend prioritising functions that are a) critical to delivery, and b) significantly different to the legacy functions. Priority functions that often meet this criteria include:

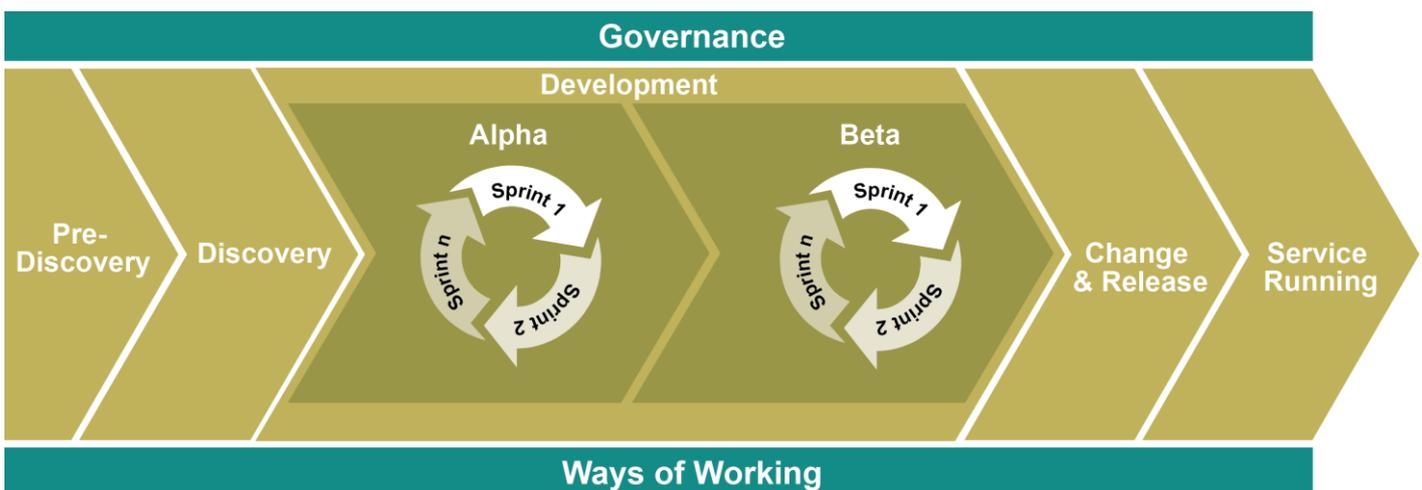
- **User research and stakeholder management** – conducting ongoing user research is an integral part of agile delivery; establishing the people that can do this effectively is a priority
- **Digital service management and implementation** – establishing product owners to set the strategic direction of the service
- **Service development** – the agile delivery capability to conduct all key development activities
- **Operations management** – project delivery management capability to support, de-risk and structure DDC deliveries
- **DevOps or WebOps** – establishing the infrastructure and tooling processes and platforms necessary to support frequent iterative releases
- **Recruitment and talent management** – resourcing teams, and supporting on-boarding and capability development; this can be a big challenge when establishing a large-scale DDC
- **Continuous improvement** – responsible for development of common design and development standards; lead business change initiatives to stabilise and develop the DDC; embed the digital delivery operating model within the wider organisation.

## 4.4 Process

As the primary objective of DDCs is to develop and maintain new digital services, building a clear understanding of the delivery processes and ways of working is essential for all team members. This is particularly important for organisations that are transitioning to agile delivery. You should consider how agile will operate within the organisation and what information the teams need to deliver effectively. This section outlines the end-to-end lifecycle used in public sector agile delivery, as well as key process definition and implementation activities required.

### 4.4.1 An effective agile delivery lifecycle

Chapter 3 summarised the key attributes of agile delivery. New DDCs must consider how to embed agile ways of working and processes within the organisation. Figure 13 outlines an effective end-to-end approach to public sector agile delivery that has been used successfully in the HMRC DDCs.



**Figure 13: HMRC digital delivery lifecycle**

This model involves the following delivery stages:<sup>15</sup>

<sup>15</sup> <https://www.gov.uk/service-manual>

**Pre-discovery** – the objective of pre-discovery is to understand whether there is potential value in taking forward a new digital service idea by conducting a light touch evaluation and securing approval and funding to proceed. This evaluation should seek to answer at a high level:

- Whether sufficient user need for the service exists to proceed to Discovery?
- Whether this user need is best met by a digital service?

To support an effective pre-discovery process, you should consider how projects are identified and prioritised by the business for delivery by the DDC. Consider also how funding is approved at this stage.

**Discovery**<sup>16</sup> – A short phase focused on establishing a user need for the service and defining at a high level what functionality will meet this user need. This will inform a high-level plan for subsequent development phases, with a prioritised high-level product backlog of functionality. Core members of the scrum team, including user researchers and the product owner, research the needs of the service users, understand how current services are performing, find out what KPIs should be measured, and explore technological or policy-related constraints. Outputs include:

- An initial prioritised list of user needs
- The team and capability required to complete the project
- Rough prototypes and user personas
- A decision to progress to the next phase – alpha

**Alpha**<sup>17</sup> – The main objective of alpha is to build a working prototype or proof of concept and receive feedback from key stakeholders or a closed group of customers on whether the solution is appropriate, viable and able to meet their needs. Core members of the scrum team, including designers, developers, user researchers and the product owner rapidly iterate prototype solutions for users' needs. Outputs include:

- A plan for beta and running of the live service
- Options for assisted digital support
- An understanding around legacy systems to replace or wrap or integrate with
- Final analysis on the research you have commissioned on user needs
- A decision to progress to the next phase – beta.

**Beta**<sup>18</sup> – The objective of beta is to build the service. Test it with users and continuously iterate it until it is ready to go live. “Betas” can be released as private (regional or invite-only, providing the team with more control) or public (open to everyone, and can exist alongside an existing service). The phase is completed by the full scrum team. Its duration depends on the scope of the service being developed, although it should not be longer than a few months.

The iterative nature of agile means that the service can continue to be improved over time in beta and live. The scrum team takes account of ongoing user feedback and develops functionality that was not a top priority for the initial beta release. Key to this is being able to release code quickly into the live environment supported by a continuous integration approach. The key output is that the end-to-end service works for customers. Other outputs include tested assisted digital support for the service, and accurate metrics and measurements to monitor KPIs.

**Live**<sup>19</sup> – The service is ready to go live once it has met the user needs identified in the previous phases. It should also meet all security and performance standards and pass the digital by default service standard<sup>20</sup> assessment. Analytics should be configured to accurately monitor the key performance indicators identified, and the team should plan the transition or integration of any existing services. Following launch, the service should be improved continuously, based on user feedback, analytics and further research. Technical and customer-focused operational support should be put in place as necessary, and monitoring methods should

---

<sup>16</sup> <https://www.gov.uk/service-manual/phases/discovery.html>

<sup>17</sup> <https://www.gov.uk/service-manual/phases/alpha.html>

<sup>18</sup> <https://www.gov.uk/service-manual/phases/Beta>

<sup>19</sup> <https://www.gov.uk/service-manual/phases/live>

<sup>20</sup> <https://www.gov.uk/service-manual/digital-by-default>

be implemented. The development process (discovery, alpha, beta and live) will be repeated for smaller pieces of improvement work that are identified during this phase – the team will find something that needs improvement, research solutions, iterate and release rapidly.

#### **4.4.2 Embedding agile delivery processes within a DDC**

Give teams autonomy to decide how best to conduct delivery within their team (e.g. agile methodology, reporting metrics, ceremonies etc.) There are certain delivery processes and approvals that teams will need to follow to effectively release a service, albeit the way that they perform such processes may vary. Process is a critical component of the DDC operating model and a priority work-strand to deliver during DDC mobilisation. We recommend developing a centralised and visible collection of documented working practices and processes in a digital delivery operational reference guide (also including other operating model components.) This is particularly relevant as traditional delivery and governance processes are replaced with new approaches that align with the agile way of working and associated timelines. Once the MVP for this operational reference guide is in place, it can act as a point of reference for both new joiners to the DDC and wider organisation, developing iteratively over time. Balance the level of detail and specificity in the guide with the need to retain some autonomy in the teams to define how they work.

The process design team should be comprised of people with combined experience of agile delivery, process design, and the existing processes, governance and stakeholders in the organisation. An integrated in-house / supplier team is often well-placed to effectively deliver the process work strand.

An effective approach to process design is to define a high-level ‘To-Be’ agile model as a reference and then work to tailor the approach to suit the nuances and legacy structures of the organisation. The GDS Service Design Manual is a great starting point<sup>21</sup>. Developing tailored delivery processes is a time-consuming activity involving workshops with a range of stakeholders in the organisation that will be impacted by the new processes. The process design team will propose changes to well established ways of working with a greater focus on user need. It may take time for stakeholders to understand the need to change and to agree upon practical yet effective new ways of working. Process design workshops should focus on understanding ‘As-Is’ processes, and agreeing as a group the ‘To-Be’ processes. We recommend designing all the processes within the digital delivery operating model, prioritising those processes that are new or involve significant levels of change. At a minimum, you should document the end-to-end functional delivery processes (pre-discovery, discovery, alpha, beta, live) and key technical processes, for example the “deployment pipeline” process that outlines how code is developed, deployed and released. You should also consider any processes that appear complex in an agile environment. Examples include:

- The testing approach
- Deployments
- Live service model
- Integration with waterfall
- Release management.

It can be a time-consuming and challenging activity to embed agile in an organisation that has traditionally used waterfall. Amongst other things, the organisation may not have the understanding of agile, the technology, or the governance processes and structures to fully adopt agile immediately. Accordingly, we recommend avoiding an “agile purist” approach to delivery as this can sometimes hinder rather than help the organisation. The process design team should account for this, defining processes that support agile ways of working within the existing business operating model. You can develop processes and ways of working iteratively, alongside wider changes to the organisation’s operating model, culture and technology.

Embedding new agile delivery processes in the DDC can be achieved by:

- Documenting end-to-end delivery processes iteratively within an operational reference guide in a central repository / collaboration tool
- Involving any existing DDC teams in the development of the operational reference guide
- Involving affected stakeholders from the wider organisation in process design workshops
- Leading process walk-throughs during mobilisation,

---

<sup>21</sup> <https://www.gov.uk/service-manual/agile>

- Process walk-throughs during on-boarding of new joiners
- Identifying 'process champions' from each scrum team, responsible for owning, maintaining and updating the operational reference guide.

### **4.4.3 Evolution of delivery processes**

Delivery processes and ways of working will continue to evolve over time, particularly when an organisation is new to agile. To ensure the consistency and relevance of the processes, you could designate a member(s) of the continuous improvement team to work with the Scrum teams and communities to continue to iterate and embed the processes as the DDC evolves.

It can also be useful to transfer greater ownership of the processes to the delivery teams. Communities of process champions, comprised of representatives from each Scrum team, can help to embed agile ways of working across the DDC. This group should meet regularly, for example once a month, to discuss any proposed changes to process. The process champions can also take ongoing responsibility for embedding and championing the processes and ways of working within their teams. To ensure that these process champions have capacity to inform and support process embedding we recommend that processes are documented in a manageable level of detail. The designated process design lead could support this community by leading any significant changes to process documentation and process embedding.

## **4.5 Governance**

Digital delivery requires governance models that allow people to make effective decisions and which align to agile delivery approaches and timelines. DDC governance needs to align business with technology, and support operational efficiency and architectural integrity, as well as accountability and performance. This section will define how to make sure the right decisions are taken on DDC delivery.

### **4.5.1 Governance in agile deliveries**

Governance in agile digital deliveries should be lightweight, collaboration-focussed and structured to support rather than delay delivery. Delivery sprints focus on developing working software rather than extensive documentation or sign-off processes. Governance processes and structures need to align to these agile delivery rhythms and timelines.

At a programme level, many of the traditional governance structures can still exist, such as programme boards or architecture boards. However, their governance processes and metrics will often need to change to limit disruption to development. It is possible to maintain the processes and structures for these groups prior to sprint development, for example during pre-discovery and discovery. During the sprints these groups should empower individuals to work with the delivery teams to make strategic decisions quickly on their behalf.

The most obvious example is the role of product owner<sup>22</sup>. They are empowered as the sole decision maker on the direction of a digital service. They own the product vision, prioritise service requirements and advise the team on what to build for an upcoming sprint. At the end of the sprint in the show and tell ceremony it is the product owner that signs off the functionality that was developed (often verbally). They can seek input from any number of stakeholders. However, ultimately it is the product owner who agrees to the product and signs off the requirements. When establishing a DDC, it is essential that product owners are empowered to make decisions on service requirements. They should be knowledgeable enough to make such decisions and have capacity to work closely with the development team and attend all show and tells.

Digital development teams are collaborative, cross-functional and self-organising. As a unit, they are trusted to develop the best possible service, informed by user need and the direction of the product owner. Teams are not hierarchical and have a flat structure.

---

<sup>22</sup> Or combination of digital service manager/product manager in some UK public sector deliveries.

## 4.5.2 Implementing a digital delivery governance framework

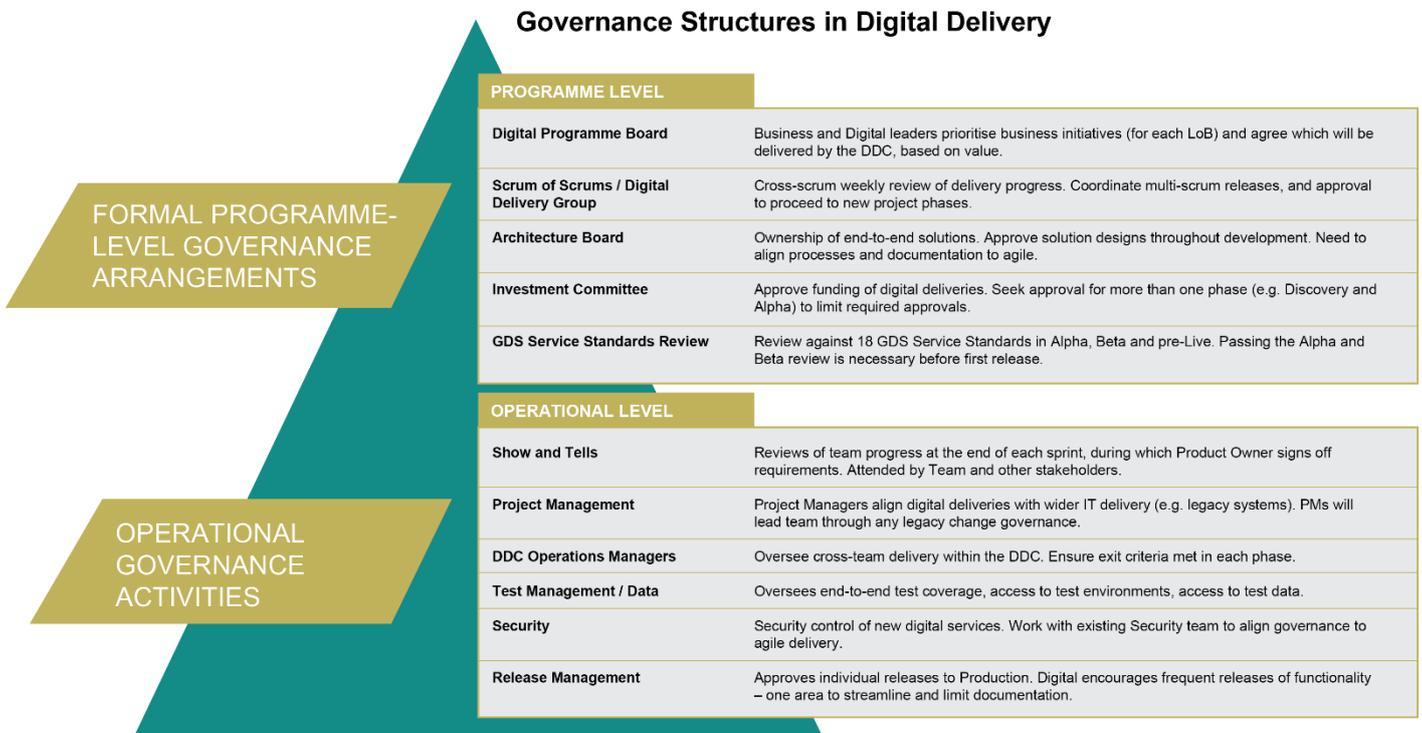
You should establish a digital delivery governance regime during mobilisation. As with the process design work-strand, this should involve a series of workshops with a wide stakeholder group to build up a picture of the existing IT governance framework and to assess whether it is fit for purpose for agile digital delivery. Assuming some change is needed, options include either:

- Establishing new groups and processes to perform the necessary light-touch, collaborative governance, or
- Changing the processes and metrics of existing structures to support agile delivery.

The governance framework may include many of the same capabilities as a traditional IT governance framework, but will operate in a different way and to a different pace. The governance framework will evolve over time as your organisation becomes more agile. One big challenge can be aligning digital delivery with the legacy approach to change governance and budget approval processes. Collaborate with these teams during mobilisation to understand what is required for the DDC teams to successfully access funding and release their services. Assess whether the DDC teams need to adhere to this governance, and if so, how they can satisfy these requirements without producing significant additional documentation. We recommend designing reports that support business outcomes, and satisfy reporting requirements of the digital programme as well as any legacy change governance.

The DDC needs to have the support (commercial and delegated authority) to quickly form specialised operational functions where required. For example, a DDC where multiple services use the same fixed number of third party test environments may find themselves with delays in delivery awaiting use of a test environment. In this example, you could work with the test environment provider to look at ways of providing more capacity for testing. An interim measure might be to form a test data management function that could help coordinate the setup and maximise the effective use of the test environments.

The governance framework adopted will depend partly on what already exists in the organisation. Figure 14 demonstrates some useful governance arrangements that would support digital deliveries at both a programme and operational level.



**Figure 14: Example governance arrangements for digital deliveries**

You can document the governance framework in a central operational reference guide to build understanding.

## 4.6 Metrics

As highlighted in Chapter 3, agile delivery uses different metrics to traditional IT project management. Metrics should be used to support decision making and are a key component of the governance model. When selecting the metrics to be used within the DDC, you should consider the target business outcomes and collect data needed to help achieve these. Agile metrics provide insights both on delivery progress and also the value that the service provides to the organisation, informing ideas for improvement of digital services and development approaches. Metrics are particularly useful *within* teams, rather than just across the DDC. They help the self-organising teams track and adapt their work. The table below outlines some of the key agile delivery metrics:

Delivery Progress	Business Value Metrics
<ul style="list-style-type: none"><li>● Team velocity</li><li>● Sprint burn-down</li><li>● Release burn-up (or down)</li><li>● Staff satisfaction in role (1-5)</li><li>● Staff satisfaction with organisation (1-5)</li><li>● Flow or throughput (CFD)</li><li>● Test coverage</li><li>● Code quality (Scala style)</li><li>● Code complexity</li><li>● Total effort to deliver the service</li><li>● Average productivity-per-Developer.</li></ul>	<ul style="list-style-type: none"><li>● Number of reduced calls / post</li><li>● Benefit delivered to the UK electorate</li><li>● Financial savings</li><li>● Contribution to overall HMRC strategic goals</li><li>● End user satisfaction</li><li>● Lead time.</li></ul>

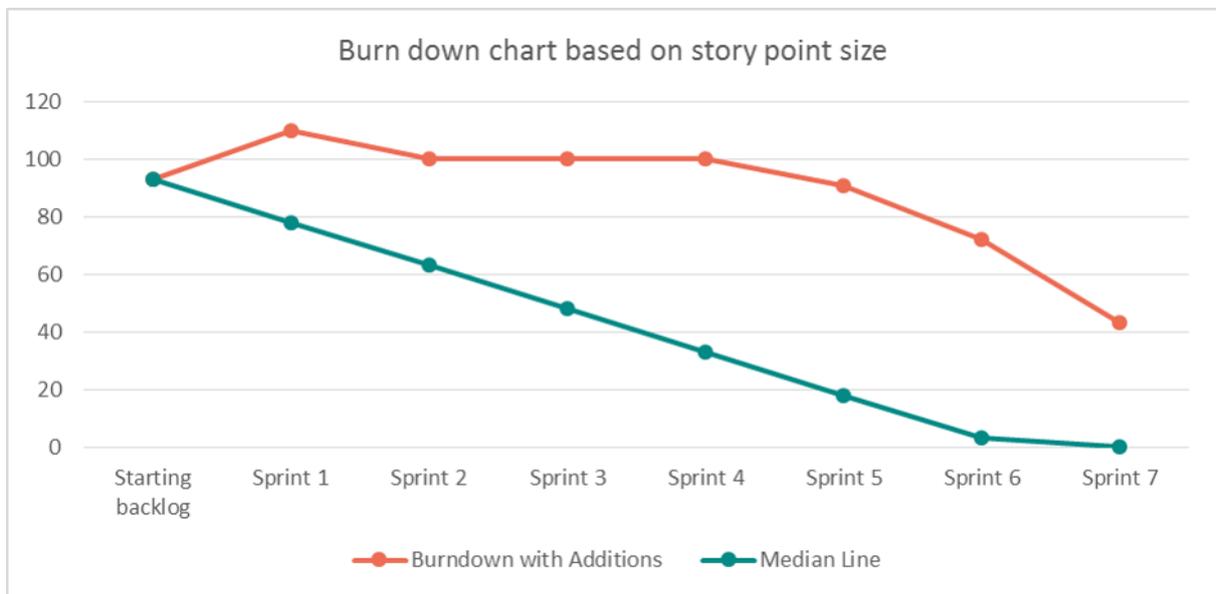
You should collect metrics that most easily help identify opportunities for improvement. It is more valuable to collect metrics on how particular delivery activities affect overall delivery progress, rather than simply collecting metrics on progress itself. Agile delivery metrics are specific to a team and should not be used to compare across teams.

The two most important metrics at sprint level are *velocity* and *value*, both of which are considered below.

### 4.6.1 Metrics demonstrating progress

Within the sprint the key metric demonstrating progress is velocity. Velocity is the per-sprint total of the story points of the user stories fully-completed during that sprint. Partially completed stories should not count towards this total. The scrum master is responsible for increasing team velocity by helping the team identify performance improvements at the retrospective meeting held at the end of each sprint. You should consider a wide scope for improvements, including possibilities such as code reuse from other projects, training and wider organisational improvement. The scrum master should be given executive support to facilitate organisational change beyond the boundaries of the scrum team, if this is limiting velocity.

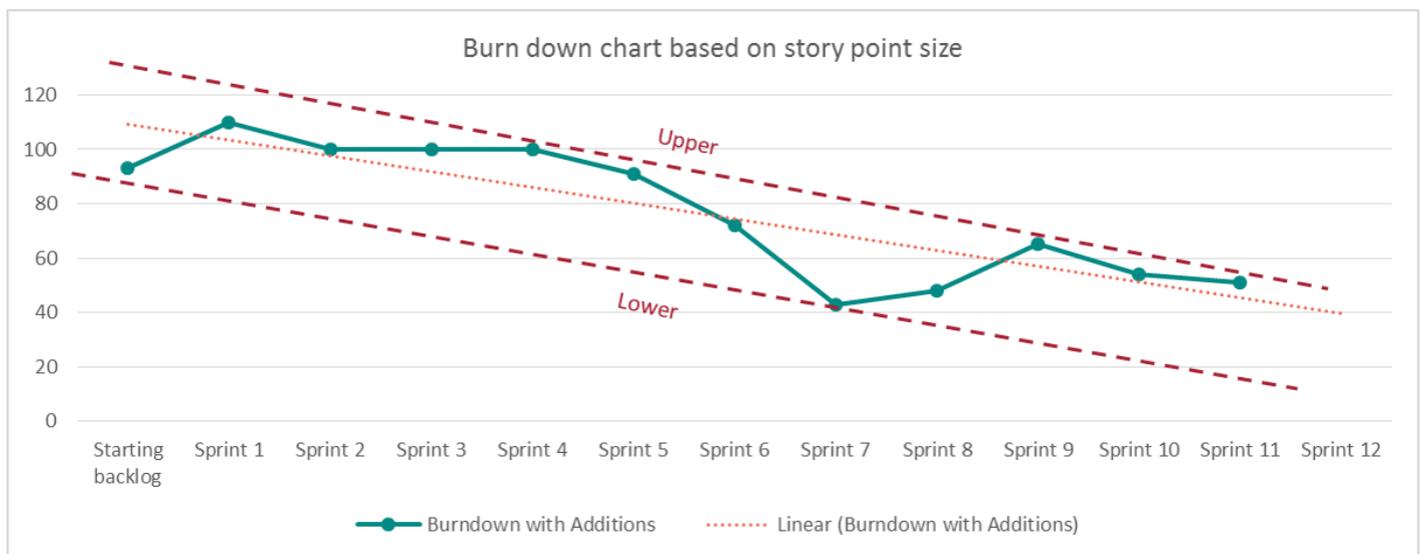
Focus on mastering more basic agile velocity metrics initially, to ensure the organisation understands the value they provide, before increasing the scope of reporting as the DDC becomes more mature. An example of this would be starting with burn down and burn up charts then moving to cumulative flow diagrams.



**Figure 15: Product burn down chart**

A burn down chart summarises the work to be completed and tracks the completion rate. It is used by the delivery team to communicate blockers and understand their relative importance to inform the actions they take next. If used as a prediction of remaining delivery time of a known scope, users need to understand the context - *How mature is team velocity? How stable are the team?*

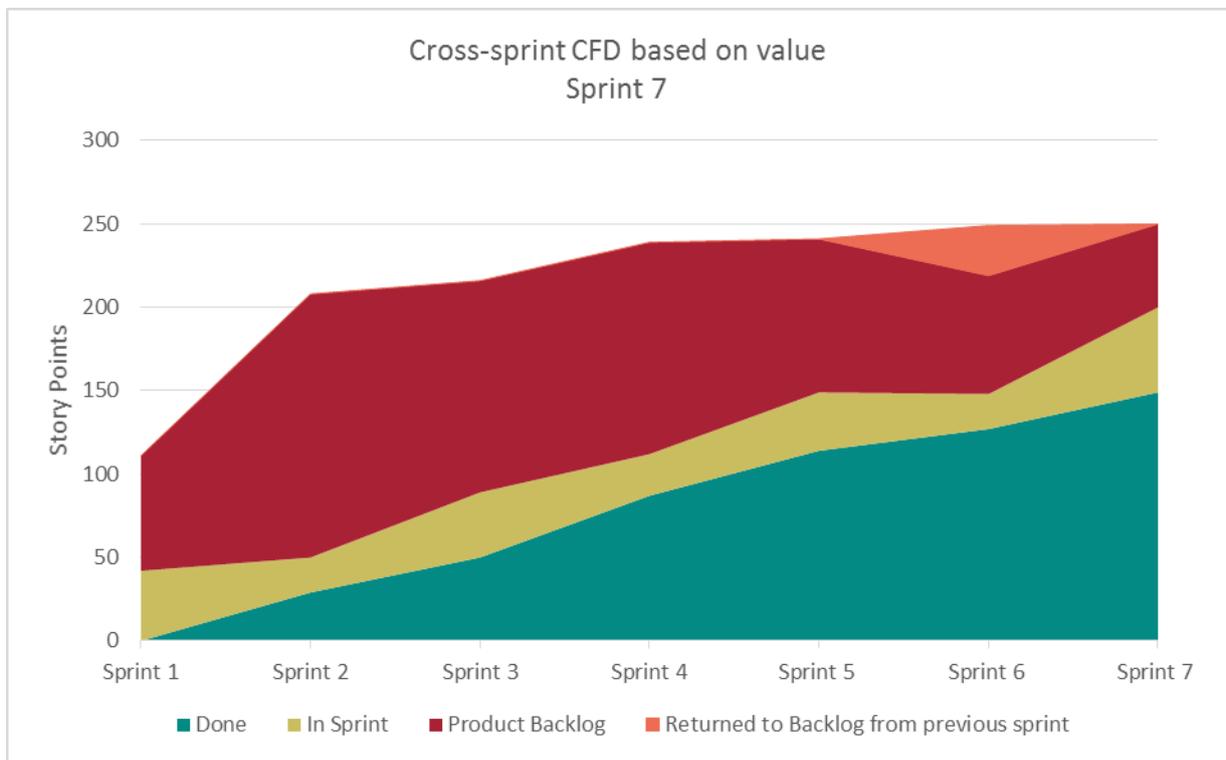
Burn down charts can also be used to indicate uncertainty. In Figure 16 the team has estimated potential delivery risks and predicted an upper and lower boundary of scope by drawing on historical data, as well as the central trend line.



**Figure 16: Burn down chart including scope boundaries**

Both burn down and burn up charts have limitations. They only indicate either the total effort left that is expected to be delivered or the total value that has been delivered so far.

You can use cumulative flow diagrams (CFD) to understand how the flow of work progresses through the team. CFD demonstrate both the effort or value delivered, and also the full process the teams go through in delivery, identifying possible bottlenecks and how long it takes for a lead in or delivery cycle time.



**Figure 17: Cross-sprint CFD based on delivered value**

Figure 17 indicates that at sprint 2, the Development Team accepted less to work on in the sprint than at the majority of other points. At the same time, a greater number of items were added to the product backlog. This type of information is very helpful for the team in retrospectives. It allows them to analyse whether a lower volume of work was accepted into the following sprint due to the time taken to review and understand the high number of backlog items.

#### **4.6.2 Metrics Used to Measure Business Value and Improvement of the Service**

Teams can measure **value** as the per-sprint total of the value of the user stories fully-completed. Partially-completed stories should not count towards this total. The product owner has a responsibility to drive up team value by breaking-down stories into smaller ones, and prioritising them to deliver the highest value aspects of a feature early.

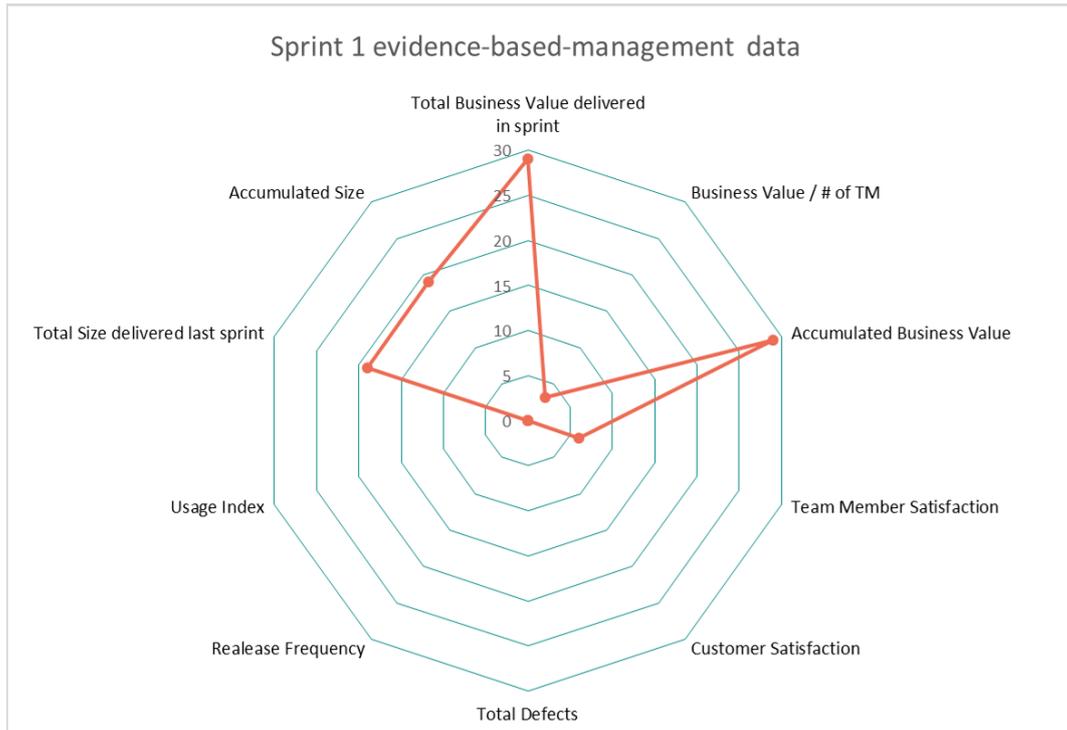
As the DDC matures, increasingly teams should align their goals to end user/business value, and capture metrics accordingly. Live services should be monitored in terms of business KPIs. Continuous improvement of services should be informed accordingly. For example, everyone in the scrum team should understand service objectives, for example to reduce call-handling time. This should be a focus of the user-stories, service design and should be the metric monitored once the service is released.

Consider how value is measured in the organisation, and how this can be captured in the DDC metrics. For example, value returned to the organisation may be yield earned from tax, reduced cost per transaction or outcome based such as being able to process x amount of tax submissions within an identified period.

An example of value measures selected on a digital project include the following:

- total business value delivered in the sprint
- accumulated business value across all sprints
- team member satisfaction
- customer satisfaction
- total defects
- release frequency measured in weeks (this will be zero until two releases have been made)
- usage index (to be captured from analytics data when the service has reached private beta)
- total size in story points delivered in the last sprint
- accumulated size in story points.

Some of the above metrics can only be generated after the service has been released. Figure 18 demonstrates how the team may capture metrics of a live service.



**Figure 18: Agility Index Snapshot**

Using Figure 18 the organisation can start looking at more complex correlations between metrics. They can also answer the following types of value and improvement questions:

- Is the effort being spent delivering value?
- Where team member satisfaction levels are high, does this lead to more value or effort being delivered?
- Do the innovations created lead to value delivered and therefore customer satisfaction?

A single snapshot will not provide insight into either a product or an agile implementation alone. Instead, these should be built up over time to understand trends. Refinement of any metrics should be discussed at team level and DDC-wide retrospectives.

On any project, the new value delivered by any given team can be expected to drop over time as a natural consequence of working through the backlog. This might give reasonable cause to trigger the redeployment of a team to areas where there is larger, outstanding value to be realised.

### 4.6.3 Continuous improvement of metrics

As the DDC mobilises, you should gradually deploy agile reporting metrics. Your initial focus should be on the more straightforward delivery progress metrics such as burn downs and burn ups. Understandably, the emphasis in the early days of the DDC will be on understanding whether teams are delivering. Using agile coaches to build awareness of agile concepts, such as the use of story points for estimating and planning velocity, will help to embed delivery progress metrics.

As the centre matures and the practices are embedded, you should consider how best to implement the business value metrics. *Does the DDC deliver more value to the organisation than the previous approach to service delivery?* All team members, from the product owner onwards, should be aware of the business value drivers. It will help the product owner and business analyst prioritise the backlog if they understand the value drivers for the service.

As with other mobilisation work strands, metrics should continuously improve over time. Until the full goals of an organisation's digital strategy have been met, improvement can be made and metrics should be

designed and adjusted accordingly. This applies not only to the DDC, but in the wider organisation, aligning the business, technology and end user to deliver more effective public services.

## **4.7 Staffing a Digital Delivery Centre**

The skills and experience needed to deliver high quality, user-focused digital services are different to traditional public sector IT teams. There are three main sources of resource supply:

- internally from other areas of the department
- externally from the labour market (including other departments)
- suppliers and contractors.

Bringing people together from a range of different backgrounds can be a significant challenge. Individuals need to be effectively integrated into the organisation and overall operating model. However, bringing in staff from different sources can also bring about important opportunities, such as the capability development of department staff from exposure to experienced agile professionals. This section will consider how to effectively resource a DDC.

### **4.7.1 Planning and managing recruitment**

You should plan recruitment as early as possible during mobilisation. The DDC management team should build a clear view of the digital delivery pipeline and understand how many employees are needed to fulfil the demand. The pipeline of work, target team composition and competency model should combine to inform the recruitment requirements and answer the questions of “*how many people do we need?*”, “*what skills do we need?*” and “*when do we need them by?*” See section 4.8 for more information on the competency model and team composition.

Consider the best mix between internal, external and supplier resources. We believe that having teams with a mixture of these groups can be beneficial. Knowledge of both the business and technology estate are essential for each Scrum team. Accordingly, internal resources are often well placed to fill product owner or business analyst roles. On the other hand, recruiting permanent staff with the digital skills that may not currently exist in the organisation is also important. These external people may come with a different mind-set and skill-set to existing staff. In some cases, they will become the digital leaders of the future. Finally, suppliers and contractors can help to scale capability quickly, providing access to the digital skill pool required to succeed. Using experienced digital practitioners also helps build the capability of existing staff.

### **4.7.2 Internal Recruitment**

Define which roles you can fill internally. As highlighted above, knowledge of the business and technology estate should be present in all teams. We also recommend filling some management roles with existing staff who understand the IT delivery management approach and key contacts within your organisation.

Where there are capable individuals with some, but not all, of the skills needed for DDC team roles, one option is to place them in the teams as trainees. Suppliers can coach and upskill these individuals. These resources need not necessarily replace any of the full members of the team, but could be tagged as trainees with a view to backfilling supplier resources once they are upskilled. Nonetheless, specialist digital delivery roles are challenging and time-consuming to learn, and not suitable for everyone. Capability development techniques exist to upskill less experienced team members (see below), but this should not mask the investment in training and learning required.

### **4.7.3 External recruitment**

External recruitment can provide the specialist digital skills needed for high quality digital deliveries. Recruiting digital leaders will help you drive change, build momentum and offer fresh ideas and experience of digital delivery, innovation, and new technologies. Hiring externally has the additional benefit of giving existing employees access to highly-skilled and experienced agile professionals.

Recruitment of digital practitioners requires a different approach to the standard recruitment methods. The DDC needs to attract digital delivery specialists, many of whom may have different backgrounds and skill-sets to those who traditionally have worked in the Civil Service. Public sector organisations are competing with digital private sector organisations (i.e. Facebook, Google, Amazon) for the best talent and your recruitment approach and employee value proposition must be tailored accordingly. When selecting the location for the DDC, consider the local talent pool: *are there universities and a thriving digital industry in the area?* If the answer is “no”, then you may need to reconsider whether this is the best location for the DDC.

Develop a compelling employee value proposition (EVP) to share with prospective applicants. The EVP is the combination of factors that are used to sell the role, DDC and organisation to a prospective joiner. This includes the role description, benefits and pay, public sector digital agenda, location (*why is it a great place to work?*), capability development commitment (*what training can you offer?*), projects (*how does the work benefit the public?*), and technology (*which cutting edge open-source tools are used?*). Communicate the EVP to applicants via a variety of media, including website, print, job-fairs and recruiters. Potential recruits need to be motivated that the DDC is a great place to work for digital specialists.

Participation of DDC team members in the local digital community can also help publicise the DDC as an exciting place to work and help attract applicants. Encourage team members to attend digital events, contribute to digital online communities and forums, and participate in digital innovation, for example hackathons. If team members feel responsibility for the development of the DDC this can help them attract like-minded people to work there.

The digital workforce is likely to change jobs more frequently than previous employees. To reduce churn, take action to encourage the best talent to commit to the DDC. At a minimum, you must deliver on the commitments made as part of the EVP. Working in a fast-paced, collaborative digital environment with the opportunity for personal development should be achievable for new joining DDC recruits. The mind-set of public sector HR and people managers needs to change. There should be less focus on HR mechanics, for example benefits and pay, and probation periods, and more on collaboration, digital progress, innovation, new technologies and social media.

#### **4.7.4 Third parties**

Employing suppliers and contractors in the DDC can be a quick and effective method to scale. It can also provide the deep delivery skills needed to effectively build digital services and to upskill existing employees. Working with a variety of suppliers can bring a range of perspectives, experiences and working styles.

Many capability development techniques manifest themselves as part of day-to-day working within the scrum team. The mix of internal and external employees should be tuned to allow for a broad spread of digital delivery expertise across scrum teams with a particular focus on placing supplier resources with those less-skilled in order to facilitate maximum skill transfer. Employ experienced scrum masters from suppliers to act as in-team coaches to your internal employees, while building momentum and increasing the understanding and adoption of agile.

*In its DDC Newcastle, HMRC provided 25% of staff in the first 10 teams. Following the development by Accenture of a structured approach to capability development, HMRC now provides 50% of staff across 20 teams. This % continues to rise.*

## **4.8 People and capability development**

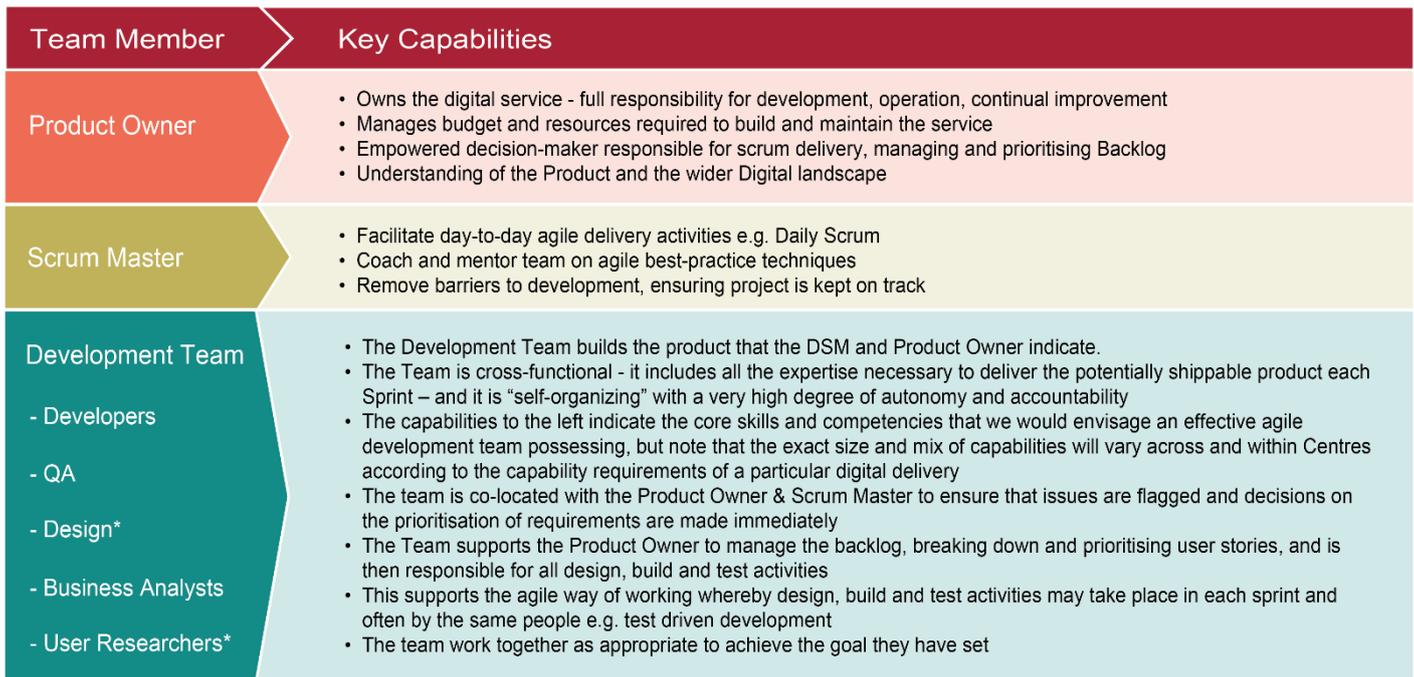
A clearly defined organisation structure, roles and responsibilities are essential to the DDC operating model and staffing strategy. During Mobilisation, a structured approach to capability development is a priority in order to form high-performing teams in a short timeframe. This will also help build the skills needed to deliver high quality digital services, and to deliver on the commitments made as part of the EVP. This section will outline key steps needed to address these challenges.

### **4.8.1 Digital delivery team and organisation structure**

We recommend defining delivery team compositions and the wider digital delivery organisation structure. This should include delivery components within the DDC as well as any other connected teams. It is

important to develop well-defined role descriptions within the DDC to ensure clarity of roles and responsibilities for all team members, irrespective of background or experience.

Chapter 3 outlined that the three key roles in agile delivery are product owner, Scrum master and Scrum team. Figure 19 below provides further information on the key capabilities of each of these roles.



\* Could be shared across multiple scrum teams. These team members are supplemented by a number of cross-cutting resources.

**Figure 19: Key responsibilities of Scrum team resources**

The figure above outlines the key capabilities we would typically recommend in an agile delivery team.

- **developers** - write and test software to develop digital solutions incrementally; work with business analysts and the product owner to ensure user stories are translated into service features; also work with QAs to conduct test- or behaviour-driven development
- **QAs** - ensure that software has sufficient test coverage, requirements are testable, teams have access to the right data and environments, and automated testing is deployed; assist developers in writing and executing test scripts in each sprint through test-driven development
- **UX design** - lead design of all user elements of a digital solution (visuals, interaction points etc.); lead design activities with support from other team members
- **business analyst** - translate client requirements into a prioritised list of user stories in the product backlog; support the product owner in prioritising; business analysts also work with user researchers to translate user feedback into user stories for the backlog
- **user researcher** - conduct interviews, test prototypes and collect feedback from user groups; work with business analysts to ensure feedback is integrated into user stories.

*HMRC has found that projects involving the development of a low number of web pages often require a proportionately higher number of User researchers and BAs than Developers*

The size and composition of each Scrum team, including number of each role, should vary according to the size and scope of delivery. **Scrum teams are staffed according to a skills template, not according to a role template.** Initially, as the DDC ramps up it may be useful to have a standard team size, adjusting for each project as maturity grows and understanding of delivery requirements increases. Including apprentices and trainees in the teams is a good way to upskill staff for the future, learning alongside experienced digital practitioners; however these should be in addition to, rather than instead of, core Scrum team members.

Aligning the flat control structure and self-organising nature of Scrum teams with organisational grade structures can be challenging. It is unlikely that all team members will be at the same career level. However, teams must be encouraged to ignore any hierarchies during agile development so as to debate, challenge

and collaborate as a self-organising team. For example, we cannot have team members deferring to the product owner solely because s/he is a higher grade. Accordingly, it may make sense for your Scrum team members to have line managers outside their Scrum team, yet in the same community. Having a product owner or Scrum master as line manager for some of their Scrum team members may stifle challenge and input.

Adherence to agile principles, a well thought-out team composition and good tooling are all required to maximise team productivity. An important lesson from HMRC DDCs is that the delivery teams should clarify exactly what actions and input are required to get a user story "Sprint-ready". These actions should be planned and suitably resourced. Teams can increase productivity by limiting the number of business stakeholders that input into delivery decisions. Teams should prepare what is known as a "Definition of Ready" (to complement the "Definition of Done") in order to get the story "Sprint-ready". The "Definition of Ready" sets out all conditions and criteria a user story must meet before it can be entered into a sprint backlog. The aim is to feed the Scrum team a steady stream of work to maximise their productivity.

To complete end-to-end delivery of high-quality digital services, the Scrum teams need support from local cross-cutting resources that sit outside the Scrum team, but provide input at key points. These include:

- **operations management** - overall responsibility for digital delivery activities within the centre; provide updates to senior management
- **technical architects** - provide technical leadership across multiple Scrum teams; ensure the effective analysis, design and implementation of high-quality technical architecture
- **solution architects** - responsible for designing the solution; ensure that it meets user requirements and aligns to wider IT strategy
- **content designers** - responsible for writing the content that will help users understand how to use the new digital services
- **end-to-end test management** - work with development teams to ensure that teams have everything they need to effectively conduct testing (e.g. data, environments); help teams conduct full end-to-end testing prior to new releases
- **HR/people managers** - overall responsibility for all people-related issues in the DDC, including HR and performance management
- **continuous improvement team** - responsible for identifying issues that are affecting team effectiveness and efficiency; design and implement continuous improvement solutions to mitigate these issues; this team can include Functional Leads within the DDC (e.g. Lead Scrum master, design, architect), as well as people with experience in delivery management, business change, learning and development and digital tools
- **digital performance analyst** - responsible for specifying, collecting and presenting key performance data and analysis for digital services.

*HMRC has found that co-locating architects with the DDC teams minimises handovers and provides more effective technical guidance to Delivery Teams than when a central pool of Technical Architects exists at one location.*

Where multiple DDCs exist, there may also be cross-DDC shared service capabilities. These support delivery of multiple digital projects at a number of different locations. Examples of these functions include: DevOps; dedicated live services; commercial; IT support; and cross-DDC functional leads of specialist areas including design, user research and architecture (likely to sit in a cross-DDC continuous improvement team). While teams have autonomy and flexibility on how services are developed, the functional leads help define and ensure that each individual service conforms to the organisational, or platform-level, digital design principles and standards.

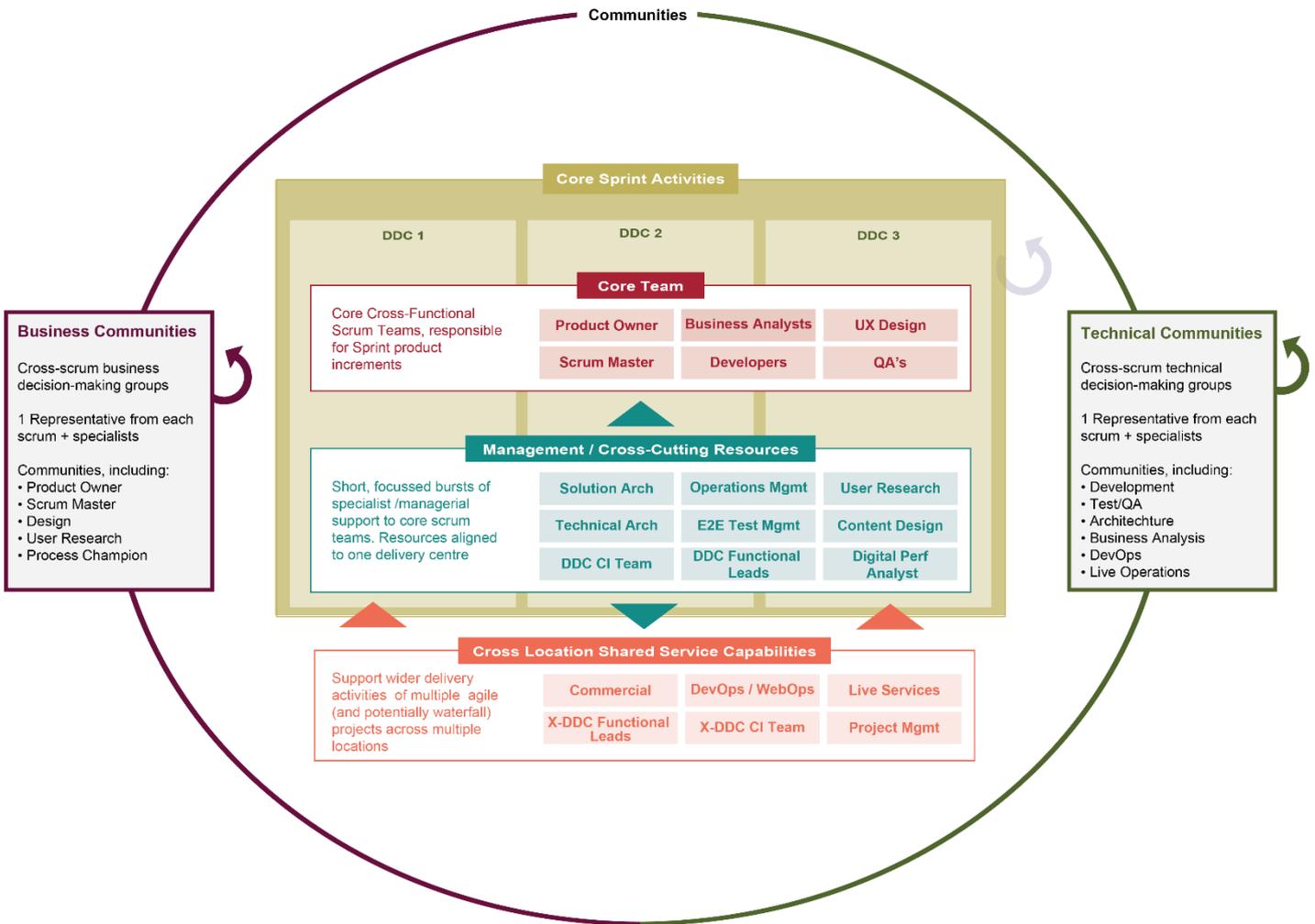
#### **4.8.2 DDC communities**

We recommend establishing cross-team communities of specialists e.g. product owners, Scrum masters and developers. These communities should meet regularly to share information and best practice, upskill their members, and set the strategic direction of the centre in their specialist areas. This is a collaborative and effective way to build capability and reach centre-wide specialist decisions quickly. You should communicate the purpose of the communities across the centre and demonstrate their value. This will help support their operation. Ownership of the communities lies with the teams themselves, as opposed to

management. Over time these can expand to Cross-DDC communities. The communities provide economies of scale and generate many ideas for continuous improvement of the DDC.

This is more than just a matrix organisation structure. It is heavily weighted towards delivery through the stable, co-located scrum team (vertical dimension), while the communities meet periodically to share information, learning and drive continuous improvement (horizontal dimension).

Figure 20 demonstrates how communities interact with the different digital delivery groups.



**Figure 20: Digital delivery roles and organisation structure including cross-team communities**

### 4.8.3 A structured approach to capability development and learning

As part of mobilisation you should define a structured approach to the on-boarding and capability development of staff in the DDC. This involves the following components:

1. Competency assessment and monitoring
2. On-boarding and initial learning
3. Longer term capability development.

#### 4.8.3.1 Competency assessment and monitoring

Develop an approach to assess and monitor the competencies of staff. The competency model should clearly set out the role-specific knowledge, skills and behaviours people need in relation to their job in the DDC. Each of the competencies should have a description that highlights what it means within the DDC, with specific mention of any tools necessary. There should be a way of assessing proficiency in each competency. You should provide definitions of essential knowledge, skills and behaviours at each proficiency level.

Categorise competencies as **Technical** (including knowledge of particular tools), **Collaboration and Communication** (focused on interactions with colleagues), and **Method** (functional and softer competencies). Candidates can complete the assessment with support from their team lead or line manager. They should assess their proficiency level against each key competency using a grading system.

Developing a competency model will help you define the skills needed for each of the roles in the DDC. It should be completed during induction, and thereafter on an ongoing basis at fixed points (e.g. twice yearly) to track development at both an individual level, as well as role, team and DDC levels. This will help you identify priority areas for development across the centre. Conducting competency assessments at fixed points every few months will demonstrate the impact of any DDC learning initiatives.

A competency model is a key building block in the DDC's Capability Development Framework. Over time this could become part of a cloud-based collaborative learning space for the DDC and wider organisation, alongside a digital skills academy for example. Initially it can be developed and completed manually (e.g. Excel).

#### 4.8.3.2 On-boarding and initial learning

The initial period after joining the DDC is critical to the performance of a new joiner. You should structure induction activities to build understanding of the objectives and work ahead for the DDC, engage people and provide the technical and functional training that each person needs to do their job effectively.

New colleagues (both internal, external and suppliers) should join in "Start Groups" of 5 or more people if possible. This will help them build a network and begin to form relationships with colleagues immediately. We recommend the following structured approach to induction:

**1. General induction:** All new joiners conduct the same initial on-boarding activities. No delivery work is conducted during the first few days. Key areas include:

- the vision for the DDC and how each role contributes to the DDC and wider organisational objectives
- introduction to digital leaders (this can help to build motivation among new joiners)
- the Digital agenda in government (if applicable)
- DDC organisation structures, including teams and communities
- DDC processes and ways of working
- DDC technology stack
- introductory agile courses if relevant (certified Scrum master courses provide a useful introduction)
- any standard HR, security and administrative activities.

**2. Role-specific on-boarding:** Tailored on-boarding conducted within the team setting. You can plan the length and content of this training based on the competency model insights. Define on-boarding programmes for each role, and where necessary competency level.

Where a full new team joins, set 2-3 weeks aside for the team to upskill before starting delivery work. Where one person joins an existing team, allocate at least 50% of their time to upskilling in their first month. Deliver on-boarding initiatives across multiple channels if possible (e.g. classroom, interactive e-learning, guidance material, on-the-job). We recommend structuring much of the training within the team setting as this helps team building. The collaborative nature of agile aligns well with on-the-job learning. Suggested activities include:

*The DDCT successfully on-boarded new joiners in one month through:*

- *agreeing a shared vision for the DDCT and team ways of working*
- *reusing previously documented process guidance and learning apps*
- *mock sprint app development to learn tools and form as a team*
- *e-learning and delivery guidance recommended by other DDCs*
- *shadowing other DDC teams.*

- mock sprints where teams can form and learn the processes and technology in a safe environment; developing a tutorial or sandpit application can help with this
- documenting the operating processes; teams can use this to inform the mock sprint
- documenting the deployment pipeline and encourage new joiners to learn about the tools and practice some of the development techniques

- e-learning on development tools and languages
- seconding joiners to established teams for one sprint to shadow experienced colleagues
- employing experienced agile coaches to upskill new joiners.

Once new joiners have completed this intense on-boarding, learning and capability development, activities should continue as set out in the section(s) below.

#### 4.8.3.3 Longer-term learning and capability development

We recommend developing a framework in Mobilisation which sets out the approach, format and key activities for the long-term capability development of staff. We also recommend developing tailored learning journeys for all employees, outlining core and recommended learning options, aligned to their competency assessment results. It is useful to involve any existing teams in the development of this framework. The specialists in the teams will likely be aware of useful learning materials for their area of speciality (e.g. design, testing etc.)

*DDCs create jobs and stimulate the local digital delivery market. Accenture and HMRC staffed school leaver apprentices and trainees with no previous digital delivery experience in the DDCN. Following a 12-month specialised on-boarding and capability development programme, including learning alongside experienced digital practitioners, the apprentices and trainees developed the skills needed to become full team members.*

The first step is to map learning options against each of the skills included in the competency model. Ideally, content should be provided through a variety of channels, dividing learning into short bite-sized chunks that can be delivered on the job and within agile delivery timelines. Some areas may require set-piece training, either delivered by an experienced team member, or through professional trainers (e.g. on using new development tools). Once the catalogue of learning options has been established, line managers, coaches and the continuous improvement team can develop tailored learning

journeys for each individual. These journeys prioritise development options based on each individual's competency assessment results. As with the competency model, over time this could be developed into a cloud-based digital skills academy, but can be delivered manually initially.

Finding time to learn and upskill is a challenge in many DDCs. Consider allocating a set time periodically across the DDC for learning. For example every second Wednesday afternoon could be used for capability development exercises. Including team learning as user stories within sprint backlogs will ensure the team are allocated time to build the skills they need to increase efficiency. Collaborative, on-the-job learning is a great way for people to learn from more experienced colleagues.

We have used a number of effective learning techniques:

- **Formal classroom-based training courses** - utilised for complex skills or technologies that require dedicated time away from the team to learn and practice
- **E-learning** - material and courses for people to work through in their own time, on their own or in small groups; these should be as interactive and practical as possible
- **Delivery guidance** - detailed functional and technical guidance in a centralised location for team consultation on delivery processes, governance, development approaches and more
- **Coach relationships** - appointing more experienced team members as coaches for less experienced colleagues; coaches should provide ongoing guidance on upskilling
- **Buddy relationships** - unlike coaches, buddies should be at a similar career level to individuals, providing more informal guidance or help to solve any work issues or concerns
- **Brown bags** - informal knowledge-sharing sessions where an experienced team member presents on a key delivery area e.g. test-driven development, backlog management
- **Process champions** - within each team a process champion should be assigned responsibility for sharing processes, governance or ways of working with their teammates
- **pair programming** - team members work in pairs to develop and test code at the same time, on the same computer, learning from one another as they develop
- **Mock sprints** - the whole team works on developing a mock application following an agile methodology; helps teams form and helps individuals learn the tools, infrastructure, development standards, processes, governance and ways of working (we have developed tutorial applications to inform mock sprint activities)

- **Tutorial application** - develop sandpit tutorial applications; teams follow guidance and a mock backlog using all the key tools and processes needed for agile digital deliveries at the organisation; also helps team forming; the application could also be designed for individual working if a team member wanted to practice a specific skill e.g. Selenium testing
- **Recruitment of experienced digital/agile practitioners** - staff some experienced, capable agile practitioners to upskill others; experience of complex agile deliveries will prove invaluable; their remit can be fully/partly focused on upskilling DDC resources, for example, experienced agile coaches can conduct agile ceremonies, embed practices and plan and oversee learning activities in sprints
- **Communities** - role-specific cross-Scrum communities that meet regularly to share ideas and best practice, address cross-Scrum issues, increase collaboration and conduct learning.

You should also consider those working in other areas of the department that will be affected by the DDC. Consider whether they need any upskilling in the new ways of working (see Section 4.2).

## 4.9 Interfaces

Teams need to understand how to interact; both with each other, the wider organisation, and real users in order to deliver high-quality consistent services. This section outlines the way that teams can interact internally within the DDC to share information, and also the way that they should interact with third parties e.g. project management, policy, suppliers, end users etc. Developing effective communication channels helps the DDC learn and mature, informing much of the activity of the continuous improvement team (see Section 5.1).

### 4.9.1 Interacting with agile delivery teams

Only the product owner should set the direction of the Scrum delivery team, prioritising the requirements and user stories for the team to develop. As such, influence and direction to the team on the product being developed should be limited during sprints. Interested stakeholders will have the opportunity to attend show and tells in order to review progress during the last sprint, and share their opinion.

*HMRC teams have on occasion experienced pressure from external stakeholders to prioritise certain requirements. This disrupts the team and wastes time as the team take on user stories outside the product owner's prioritisation.*

In complex public sector digital deliveries, the team will necessarily have to interact with a range of external stakeholders at different points in the process. This will be particularly true as the DDC is mobilising and its functions, processes, governance and people are evolving. The product owner and Scrum master should, as much as possible, lead the team's interactions with external groups (e.g. governance committees), although it may make more sense for the team to interact directly with external teams on occasion e.g. a dependent legacy development team.

You should take steps to limit disruption from external stakeholders to the team. One effective approach is to hold an inception event at the start of any project discovery. This should include all project stakeholders. The purpose of the session is for all attendees to gain a clear understanding of:

- the purpose and objective of the service
- the scope of the project
- a very high-level solution design
- the agile development approach and ways of working
- method of interacting with the team (e.g. through product owner).

This informs the development of a high-level view of delivery timelines, estimated effort and agreement on delivery roles and responsibilities.

### 4.9.2 Identifying the key interfaces

During Mobilisation, identify the individuals and teams that the DDC teams have to interact with in order to deliver successfully. You **must** engage these stakeholders during Mobilisation to inform the operating model design, and to ensure they understand how the teams will interact with them.

The table below summarises some key teams or individuals that public sector Digital Delivery Teams have to interface with (though we recognise this may vary between organisations).

Team	Scrum team Interaction guidance	Frequency
Customers / Citizens	Conduct user needs analysis throughout delivery to inform service requirements.	Ongoing
Back office operations	Conduct user research of operations teams to understand their user needs and to better understand existing services.	Ad-hoc
Contact centres	Conduct user research of contact centre teams to better understand existing services and user needs. Regular interaction with first line support required when service is in production to respond to incidents.	Ad-hoc pre-release. More frequent interaction once service is in production
Business SMEs	Product owner identifies suitable SMEs to engage with, to understand business requirements and inform user story development. Attend show and tells at end of each sprint.	Show and tells
Operations management	Weekly delivery report to DDC operations managers. Seek sign off for major release plans, and MVP. Greater levels of interaction may be required as projects approach live releases or when there are delivery issues that cannot be resolved within the team.	Weekly (or more frequently)
Architecture office	Enterprise architect should be aligned to the project, working closely with DDC architects to integrate DDC solution into wider end-to-end solution involving other Delivery Teams. The Architecture Office may sign off the solution at set points in delivery (try to minimise the number of sign-offs).	Ongoing work with Enterprise Architect
Legacy system delivery teams	If the service integrates with a legacy system the Scrum team needs to align with legacy delivery teams to confirm delivery dates, and integrate software. Secure test data/environments from legacy teams as needed.	Ongoing basis as needed
Project management	Share access to delivery plans and progress to align with any wider deliveries. This could include attendance at show and tells and sharing delivery metrics.	Weekly or more frequently as required
Continuous improvement team	Continuous improvement team(s) work with teams and communities to identify areas for development and issues/risks impacting team delivery.	Ongoing
Functional leads	Set standards and ensure consistency and quality of services (e.g. design, architecture and development). These resources sit within centralised cross-DDC continuous improvement team.	Ongoing
Security	Ensure service meets cyber-crime and security standards.	Regular delivery checkpoints
Internal governance groups	Minimise the number of governance sign-offs and groups during sprint delivery. See Governance Section 4.5 for more information.	Variable
DevOps	Interaction for tool requests, infrastructure access and co-working to release services. Greater levels of interaction initially. As DevOps matures greater automation will reduce interfaces.	Initially very regularly, will reduce as DevOps matures

Live service	If a dedicated live service team exists, then a structured handover of the service is required. During initial period after handover, daily interaction is likely. Consider seconding a team member into the live service team.	Daily after service handover
Government Digital Service	Develop services in line with the Government Service Design Manual. Complete GDS Service Standard reviews in alpha, beta and live.	GDS reviews in alpha, beta and live

### 4.9.3 *Sharing information between DDC teams*

As the DDC scales you should focus on establishing effective mechanisms to share information between teams. Teams learn the same development process, tools, system architecture, deployment pipeline, governance and delivery estate. Each will progress at different timelines, so reusing learnings and materials can only be helpful. Effective solutions include:

- deploying internal collaboration tools such as Skype, Lync, Google Hangout
- storing key functional and technical guidance (e.g. the DDC Operating Model, deployment pipeline etc.) in a central document management system or wiki e.g. Confluence
- a dedicated continuous improvement team to design and implement activities to share information and to upskill the teams e.g. through Brown Bags
- holding regular Scrum-of-Scrums to share delivery progress and address cross-Scrum issues
- seeding experienced team members from the initial exemplar Scrum teams into new teams
- establishing communities for each specialist area in the DDC
- set piece events such as ‘DDC Speed Dating’ to provide short overviews of deliveries to other teams.

## 4.10 *Tools*

To successfully deliver agile projects, you should define and deliver an appropriate tool-set. Effective use of tools also improves staff engagement, automating easier tasks, increasing time spent on higher value tasks.

### 4.10.1 *Tool selection*

Many digital delivery tools can be used for the same purpose. Few of these tools will be perfect and satisfy every user (e.g. the age old Linux vs. Windows debate). You could engage with an organisation already doing similar work or with a significant breadth of IT delivery experience to better understand the industry leading tools and determine which to select.

We recommend teams within the DDC to use the same toolset, dividing trials of new tools between them. A consistent toolset across teams improves the speed with which teams identify the optimal way to use the tools. This also reduces the overhead in learning and supporting the tools and means any movement between teams is more straightforward.

### 4.10.2 *Maintaining a relevant toolset*

Encourage the specialist communities in the DDC to continually review their toolset and look for additions or improvements. In a large DDC, a lot of improvement ideas will come from the breadth of backgrounds of the experienced staff recruited into the DDC. Encouraging staff participation in the many publications and online communities that review new tools will help identify improvement opportunities. Selecting open source tools are typically free to use, reducing your investment in tool deployment.

The DDC needs a DevOps function to set up the tools and support their use. In addition, encourage the more experienced team members to publish user guides and tips for others on how best to use the tool. The focus should be on quickly getting started on the tool, and hints and tips on how to use it to best effect. Many users may not read comprehensive documentation; for the more complex tools it may be more efficient to shadow an existing user.

### 4.10.3 Tooling functions

Tooling can be employed to good effect in the following areas:

#### 4.10.3.1 Work management

This refers to work at both a DDC level (i.e. business outcomes) and also at a team level (i.e. features/needs). At both levels the tooling should capture historic, current and future potential work. This needs to be accessible by all stakeholders (consider your target scale-up of concurrent users). It should support customisation of the workflow for a work item. It should support the provision of the relevant metrics to track progress or measure value.

*HMRC DDCs have implemented JIRA for work management at the team level. Each team has its own product backlog and scrum boards.*

Delivery teams can use work management tools to provide visibility of what they are working on, what is approaching, and where the challenges, risks and issues might arise. Whiteboards and TVs can help with this quick access to information.

#### 4.10.3.2 Collaboration

Face-to-face interaction is preferable in agile delivery, but is rarely possible all of the time. Collaboration tools help to share information constantly throughout development, while minimising travel costs (where you are operating multiple DDCs). The collaboration tool(s) you select must be available to all stakeholders that might be involved in the deliveries. Useful features include:

*HMRC DDCs use Confluence, Google Apps (Drive, Calendar, Groups and Hangouts) and Skype to collaborate*

- support for searchable wiki style articles
- video calls, with screen-sharing and whiteboarding
- instant messaging, ideally with an indication of availability as well
- ability to share calendars and documents amongst staff and schedule use of any limited resources such as meeting spaces, test environments (both virtual and physical)
- chatrooms, searchable and with ability to store history
- question and answer tools.

#### 4.10.3.3 Development and testing

The development toolset will be influenced by the service being developed. Discuss with any experienced staff in the centre which tools they are experienced in and prefer, whilst still encouraging exploration. Balance the effect on delivery of using the latest toolset with the skills and experience available to you e.g. quality and efficiency vs. ability to adopt tools rapidly.

Development tool requirements include:

- tools that support scripting of all the testing types relevant to delivery to minimise the effort required to repeatedly re-run tests and support iterative development:
  - Unit testing (that facilitates test driven development)
  - Functional testing (that facilitates behavioural driven development)
  - Compatibility testing (emulation of multiple physical end device types and browsers)
  - Security testing
  - Performance testing
  - Accessibility testing
- tools that support continuous integration (CI), maximise test automation and perform checks every time the code-set is changed
- tools that support software configuration management, particularly those that support coding standards and reviews
- an Integrated Development Environment

*HMRC DDCs use the Scala programming language and testing tools including ScalaTest, Cucumber, Browserstack, OWSA ZAP, Gatling, and Wave.*

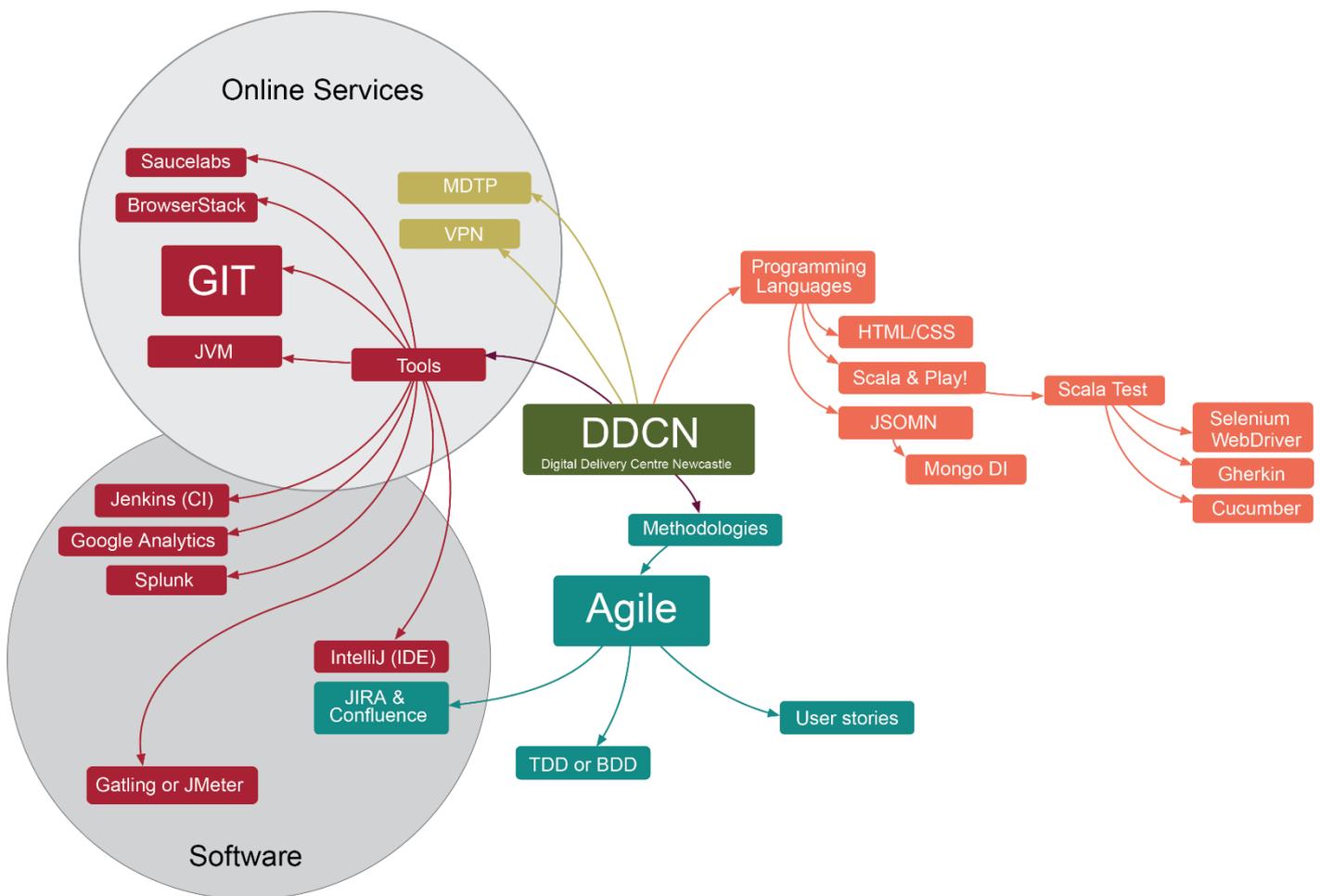
*HMRC DDCs develop and manage the deployment pipeline using: Jenkins, Github, Scala Stylechecker, FindBugs, Wart Remover, IntelliJ, Scoverage, and Play.*

- the use of appropriate frameworks, chosen based on the coding language being used and the application being developed, will improve the speed with which applications can be delivered
- tools that help monitor development, providing useful metrics for people to act upon e.g. code coverage
- tools that help monitor the applications in production e.g. hardware utilisation, application log analysis, user data analysis and defect management and investigation; tools such as Google Analytics also support multi-variant testing
- tools that allow the automated creation, configuration, re-configuration, response and recovery of environments
- test environments, including those of any third parties (e.g. Heads of Duty Systems) - the ideal setup is where teams can self-serve, i.e. create their own test environments to the performance specification they need, deploy to them as they need, setup the test data they need, and reset them to any given point in time, as and when they need (this setup requires the use of virtual machines, probably with cloud hosting to provide the flexibility needed to scale)
- tools for automated build and deployment of applications to environments with the appropriate configuration for that environment - this provides more control and flexibility to the delivery Teams and speeds up some of the more mundane activities involved in development.

*HMRC DDCs utilise Google Analytics, Splunk, Grafana, Kibana, Deskpro to monitor applications in production.*

*HMRC DDCs use effective Python scripts alongside Jenkins, Puppet and Chef to achieve automated builds and deployments.*

Figure 21 shows the HMRC DDC Newcastle technology stack.



**Figure 21: HMRC DDCN technology stack**

#### 4.10.3.4 User-centred design (UCD)

*HMRC DDCs have used Camtasia, Survey Monkey, Silverback and a custom rapid prototyping tool.*

If the DDC is developing front-end user interfaces you should invest significant effort into UCD practices. There are a number of tools that can help support UCD. These include survey tools, data analysis tools and tools that allow video recording of user research sessions (including providing data into where the user is looking at any given time). Tools that allow rapid construction of interactive prototypes will support the delivery teams in testing ideas with end users to identify the best design, before commencing full end-to-end delivery.

### 4.11 DevOps

Creating a DevOps (Development Operations) platform for delivery is an essential element to creating a successful DDC. **Without a strong DevOps platform the DDC will not be able to meet its stated goals.** Accordingly, development of a DevOps capability has to be among the very first activities in Mobilisation.

A DevOps function transforms delivery from:

- long lead times
- late testing
- difficult deployments
- manual processes
- developers writing non-functional code.

Into:

- new environments in minutes
- fully automated testing
- automated deployments
- automated process and tooling
- developers only writing functional code that adds value to the business.

A strong DevOps function provides the technical building blocks to allow agile Scrum teams to create and scale environments, develop and maintain automated testing and deploy to Development and Production environments.

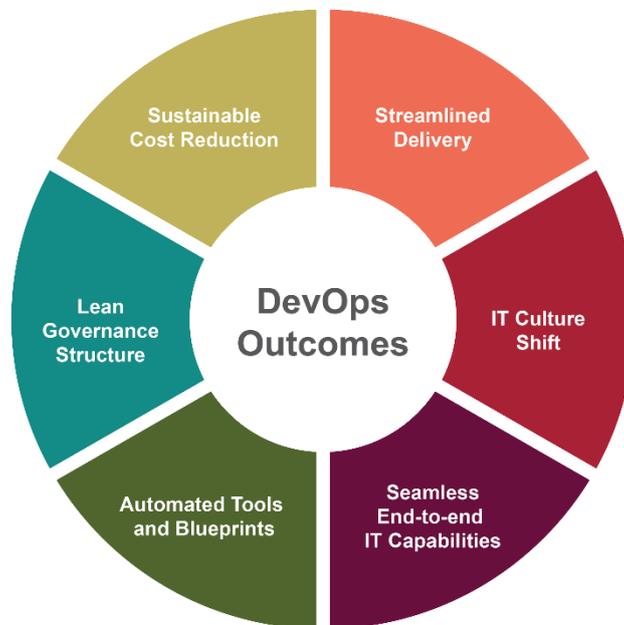
The DevOps capability must be among the first you create in Mobilisation. Starting digital delivery without a DevOps platform will result in a non-cohesive technology approach. This leads to rework and a lack of consistency.

When setting up the initial DevOps function you should create a capability maturity model. This will keep the continuous improvement of your DevOps capability aligned to the strategic intent of the service, i.e. automation, consistency and ensuring that the functions created continue to service the core capabilities of the DDC.

DevOps covers the following facets:

- streamlined delivery
- IT culture shifts
- end-to-end IT capabilities
- automated tools and blueprints (continuous integration, continuous delivery, continuous testing)
- lean governance structures
- sustainable cost reductions.

This section will cover each of these elements in turn.



**Figure 22: DevOps outcomes**

### **4.11.1 DevOps vs. WebOps**

Over the last 15 years most technical organisations have transitioned from System Administration to Web Operations (WebOps). Typically, WebOps teams are talented generalists. Their subject matter expertise spans system and network configuration, security, application deployment, database management and troubleshooting across these areas. The more sophisticated WebOps teams focus on automating environment configuration and deployment. However, system and developer support is a manual affair, with a single WebOps engineer working on a single problem or developer interaction.

Since the advent of the cloud, many technical organisations have transitioned from WebOps to DevOps. DevOps involves automating more operations, security and testing functions. This further reduces the number of one-to-one interactions required between operations, development, and security and testing. This is achieved by adopting powerful scripting languages like Ruby and Python and using the same tools as development such as Software Configuration Management, automated-build and workflow. The wide variety of new tools facilitates the DevOps movement. A key differentiator between WebOps and DevOps is the DevOps platform. This is an implementation of a range of operations tools made accessible by APIs and portals that allow Scrum teams to build their own environments from a service catalogue provided by the DevOps platform. This automatically builds secure environments on secure networks. It also provides appropriate tools and capabilities to allow Scrum teams to start coding, building and testing without any need for personal interaction with operations engineers.

In this section, and throughout the document as a whole, we refer to DevOps. We do acknowledge that you can also use a WebOps model to support digital delivery with some, but not all, the DevOps capabilities. HMRC has used a WebOps model for its existing DDCs.

### **4.11.2 Streamlined delivery**

The goal of any DevOps capability is to streamline IT delivery. The crux of this is to create a **DevOps platform** that can support all digital delivery. The size of your DevOps team should be dependent on the level of sophistication of the platform required. The goal of this centralised team is to create a set of automated tools and processes that ultimately result in fast, test-driven deployments into live environments. This starts with the automated creation of an environment and ends with deployment into the live estate.

### **4.11.3 IT culture shifts**

The change from Waterfall-driven development, with IT departments (in-house or supplier) providing fully-scaled environments, to a largely self-service function that can commission environments in minutes, is a significant cultural change for any organisation. Training and communications will help you successfully make this transition. Training should concentrate on three main areas:

1. **What is DevOps?** An overview on environments, tools and test-driven development.
2. **How to use DevOps:** A summary of offerings by process, e.g. how to obtain a new environment.
3. **DevOps in action:** Demonstration of the functions in action.

This cultural shift requires buy-in from senior management. Building blended teams that have the combination of agile, existing business knowledge and digital skills will help, but a structured training programme focusing on the three areas above is the most effective factor.

Encourage continuous improvement of the DevOps function to help improve the quality of services developed. Learning feedback loops should be promoted and championed. Once the training has been completed and your DevOps function is in place, establish periodic and regular DevOps retrospectives. These should mirror the structure of delivery retrospectives, with a focus on continuous improvement. Retrospectives will help drive cultural change as they concentrate on all elements of DevOps (i.e. not just technical improvements, but the DevOps Scrum team's less tangible needs as well).

#### 4.11.4 End to end IT capabilities

During Mobilisation, agree upon the minimum DevOps function required to support your agile delivery. We recommend conducting a **short discovery to understand the DevOps scope and mandatory start points**.

Use the following approach to structure this activity:

- a) DevOps kick off – establish the need for a centralised function
- b) staff a small team – comprised of DevOps experts and end users (the Scrum teams)
- c) agree the technology – use cloud based services. Settle on a single vs. multi-hosted approach
- d) agree the initial scope – define what is needed to start development e.g. environments with code repositories, continuous integration, test automation
- e) understand the roadmap – detail the target objectives for the DevOps function (self-service, links with other tools and functions such as governance and commercials)
- f) get started – begin with the first environment and test and learn through using.

#### 4.11.5 Automated tools and blueprints

A key component of the approach above is the selection of automated tooling and the environment blueprints that your DevOps team will support. This choice should be made during DevOps discovery, informed by the environment cloud provider selection. Tooling should be built in a way that minimises the need for re-writes (as has been seen in many early DevOps adopters). Three key elements for you to consider are:

1. **open source** – fast, flexible and cheap
2. **scalability** – how large a development capability is required (users / environments)? Do not choose tools with limits if this will constrain future scale
3. **adaptability** – how it will function with evolving and improving cloud providers.

#### 4.11.6 Lean governance structures

*Amazon can release code into Live every 11 seconds across up to 10,000 servers.*

DevOps enables production of code into the live environments with increasing speed and agility. Continuous integration tooling (e.g. Chef, Puppet) allows the orchestration of rapid releases, from auto deployments to automated testing functions (e.g. Gherkin, Selenium, Cucumber). You need to design a clear governance model to support this technology. Having the ability to deploy frequently is only useful if you are not hampered by a heavy governance structure. As your digital delivery capability is maturing, you need governance arrangements with the appropriate flexibility to allow for fast upgrades for certain changes, while also recognising that some changes may still require a longer and more stringent life cycle of reviews and approvals.

### 4.11.7 Sustainable cost reductions

A centralised, fully automated, open source driven DevOps function based on scalable and expansive cloud providers will provide substantial cost reductions compared to traditional development architectures. Start early in order to test and learn. Choose the right tools and create a roadmap for the DevOps service. This will help ensure that your up-front investment is used wisely and the overall benefits to the service are sustainable. Investment will reduce as delivery automation increases.

## 4.12 Physical location

To maximise the benefits of agile delivery you need a physical delivery environment that stimulates and engages the DDC team members, and supports the development of modern, customer-led digital services. Addressing the other mobilisation activities in an existing office environment is a start. Designing a new delivery space to encourage collaboration, creativity and a fresh perspective on public service delivery will help build high performing digital delivery teams. A new location not only builds engagement - it is key to effective cross-functional and self-organising agile teams.

### 4.12.1 Setting up the new environment

Setting up a suitable digital delivery location is important for two key reasons:

- agile requires working facilities that facilitate greater collaboration, encourage participation, and foster a culture of community, knowledge-sharing and creativity
- the right working environment demonstrates that digital delivery is 'different' to the other work going on in the organisation. This helps to build momentum in the new delivery approach and embed the required culture.

You should design and develop the new location during mobilisation. A large facility with room for expansion is better as further digital delivery is required/desired. The DDC design should include space for regular team breakout sessions and project management ceremonies, visible and interactive delivery management tools (e.g. whiteboards), and real-time collaboration and data analysis tools. As the DDC is sourced and secured, it can be useful to bring in experienced digital delivery colleagues to create your first development teams. You need not wait until a new delivery location is available. Seat teams together, rearrange existing desks to encourage collaboration, print out agile metrics, and deploy key agile ceremonies (e.g. sprint planning, daily stand-ups) from the outset.

### 4.12.2 Workplace and facilities

Important workplace facilities for the DDC are outlined in the table below.

Facility	Uses / Benefits
Collaboration spaces	Interactive areas that encourage creative, group working. These should include ample wall space for story boards, flip-charts and video screens. This is where many agile meetings will be held such as sprint planning, show and tells.



**Figure 23: Agile Collaboration Space**

Multiple individual monitors

Multiple monitors assist the development team to pair programme and collaborate, leading to better quality software.

Large screen monitors

Large screen monitors display analytics performance data in real-time. This supports responsive decision making, prioritisation of requirements and fluid development.



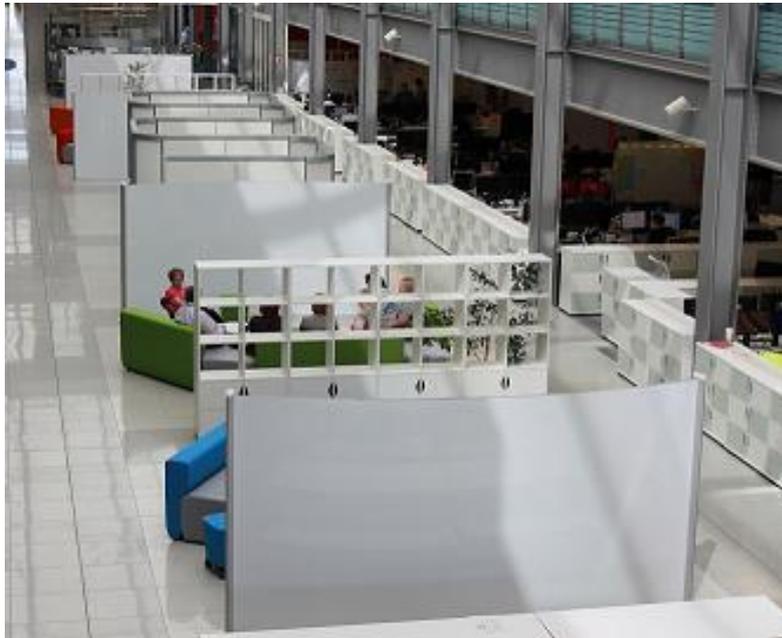
**Figure 24: Large Screen Monitors to support show and tells**

User Research Labs

Dedicated labs host face-to-face and remote user research. Key facilities include devices to track eye movements and facial expressions, technology to monitor mouse cursor movements, and adjacent rooms to view and record user reactions. These labs support user-centred design activities.

White boards

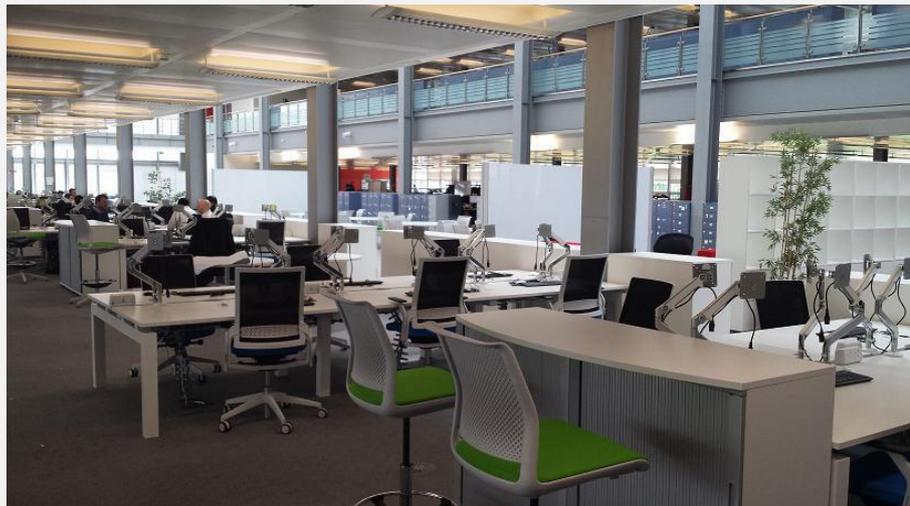
Boards used to track Scrum progress (e.g. Scrum/Kanban boards), facilitate collaboration in agile ceremonies, and help embed agile group working.



**Figure 25: White Boards at the DDCN**

Desk layouts

Align desks in straight lines for huddles. Remove dividers to increase direct communication and joint working among the team.



**Figure 26: Desk Layout at the DDCN**

Laptops

Laptops are needed as team members move around for delivery ceremonies. They require sufficient compute power to run local, SCM and build environments.

Lockers

Installing space-efficient lockers rather than desk pedestals can increase open space for team huddles and daily stand-ups.

Private meeting rooms

Include some private meeting rooms in the centre for any sensitive/confidential discussions.

Network connectivity

Open and unrestricted access to the internet at any place in the DDC to support agile, collaborative working (e.g. Show and tells). Sufficient reliability and performance required for heavy development usage.

Development specification	Set teams up with an appropriate development specification, with access to the relevant environments, tools and repositories. A standardised specification is important to support consistent delivery approaches. Choose between the responsiveness of a physical workstation and the security and flexibility of using virtual machines.
Touchdown workspaces	The fast-paced nature of agile tends to mean that Scrum team members engage in shorter, quicker and more intense bursts of work. Touchdown workspaces offer a different workspace to their desks, providing more variety and allowing them to iterate in an informal and open setting.

## 4.13 Communications

Develop a communications strategy and plan during mobilisation to structure engagement with internal and external stakeholders. Communications will help support the DDC on two levels:

- **internal communications** facilitate engagement within the DDC and with other teams within the organisation. This will help to build the awareness and interest of the people who will work in or with the DDC
- **external communications** facilitate engagement between the DDC and stakeholders outside of the organisation. This serves to support recruitment, increase the profile of the DDC, publicise new services and improve your organisation's reputation.

### 4.13.1 Establishing a communications strategy and plan

A communications strategy will establish the communications tools and techniques to support the business objectives of the DDC. The strategy should answer three questions:

1. **What business outcomes can be supported by communicating information on the DDC?**  
Define what can be achieved through the use of communications
2. **Who do we need to communicate with?** Key internal and external stakeholders (organisations, groups and individuals)
3. **How should we communicate with them?** Propose potential communications channels.

Form a communications plan informed by the above strategic targets. This plan should outline the date, frequency, stakeholders, description, channel, priority, owners and actions of communications activities. Communications are particularly important as the centre evolves in order to build internal and external publicity to support recruitment, raise awareness of new services and develop the organisation's public image.

### 4.13.2 Communications during mobilisation

Communications are particularly important during the initial development of the DDC. The table below details some suggested communications activities in during the Mobilisation of the DDC.

Communications objective	Suggested activities	Key stakeholders
<b>Internal</b>		
Attract internal recruits with the right skills and experiences	Communicate open roles at the DDC with internal candidates (e.g. emails to those who have shown interest; advertising on department intranet)	All internal staff interested/ available for staffing
Engage internal staff working in or with the DDC with the vision of the centre	Progress and delivery updates on department intranet Newsletters for digital community	All internal staff working in/with the centre

Build an understanding of the DDC's capabilities internally / use as blueprint	Host visits to the DDC for internal senior management to show what the centre can do, the benefits of both it and agile, and the front-door process	Internal management
Instil a collaborative environment within the DDC for all staff working in the centre	Encourage cohorts and communities to share knowledge and start building networks with each other through collaboration sites, e-mail distribution lists and confluence	All staff working in the centre
<b>External</b>		
Attract people externally with the right skills and experience	<ul style="list-style-type: none"> <li>• Work with the Press Office to advertise roles on social media and gov.uk</li> <li>• Review/edit/approve communications created by the DDC People team</li> <li>• Vet advertising by external agencies</li> </ul>	Local and national talent pools Universities
Build awareness of the DDC with key external stakeholders (e.g. MPs, GDS, external suppliers/SMEs)	<p>Communicate with local MPs informing them of opening</p> <p>Organise DDC visits for representatives from GDS</p> <p>Attend regional events (e.g. "Town Hall" events and GDS supplier conferences) to engaged with SMEs</p>	MPs GDS External suppliers and SMEs
Increase public confidence in the department	Press releases on gov.uk or via the GDS blog, focusing on modernity of department's capability, reduction in IT costs and creation of new jobs	General public

### 4.13.3 Ongoing communications

Certain communications activities can support the DDC on an ongoing basis. These can be managed by the continuous improvement team. Activities to consider include:

- **ongoing recruitment** - as the centre grows and recruitment continues, communicate with potential internal employees and external applicants
- **celebrating success** - celebrate both project success (e.g. if a digital service goes live) and individual success (e.g. promotions)
- **marketing** - as you release services, communicating the DDC success can help build your organisation's reputation and position the centre as a blueprint for adoption elsewhere
- **hosting visits** - other organisations may want to visit to understand scaled digital delivery - in these instances how to triage requests for and hosting of visits should be considered e.g. key messages, activities, resource requirements, and materials
- **sharing learnings** - publish learnings, guidance and best practice on digital delivery in order to help others who are looking to set up a DDC or who are facing issues in transitioning to agile e.g. public blogs, articles on GOV.UK, digital/IT websites
- **service-specific communications** - product owners should work with the Communications Team and/or continuous improvement team to develop and deliver a communications plan for each new service - the public need to understand the how the service can help them, and how to use it.

### 4.14 Work prioritisation

DDCs offer a new, fast and effective means of releasing high quality services to the public. Once the DDC is operational it is quite possible that the demand for new services from across the business will outstrip capacity. You should establish an effective process of managing the digital delivery pipeline across business areas. When projects have been selected for digital delivery, the approach for identifying and prioritising the

requirements for delivery is different in agile, involving different stakeholders. Additionally, when solutions are being delivered across multiple delivery teams (either within the DDC or alongside legacy system delivery teams) the delivery approaches, timelines and stakeholders need to be managed to ensure work is prioritised effectively. This section outlines approaches to effectively manage and prioritise work across the delivery lifecycle.

#### **4.14.1 Digital pipeline management**

The majority of digital pipeline management occurs in pre-discovery before the project is allocated to a DDC. The objective of pre-discovery is to understand whether there is potential value in taking forward a new digital service by conducting a quick, light-touch evaluation and securing approval to proceed i.e. is there a user need for the service, and is it best met by a digital service? This is a decision that should be made by a programme-level governance group (e.g. a Digital Programme Board), that includes digital leadership (e.g. CDO), business leadership, architecture leadership, and DDC leadership (with a view of supply).

*HMRC found that user needs were not always best met by a digital service for some of the initial services proposed by the business. We recommend establishing an effective evaluation process in Pre-discovery to establish that any user need is best met by a digital service before commencing more costly Discovery and Alpha delivery stages.*

Different business groups will each have their own priority services for digital delivery. To limit ongoing debate between these areas on the relative priority of individual services, an effective approach is to allocate a set supply/number of teams within the DDC that will focus on digital delivery for each area (e.g. by budget, existing services, customers etc.). A clearly defined digital delivery supply for each business group will help them prioritise which of their projects they want the DDC to deliver. Appointing a lead product owner for each area of business will further help this prioritisation. The lead product owner should have overall ownership for the digital product vision, and ability to prioritise the delivery of different digital services and macro functionality within their business group. Lead product owners can also exist at a DDC or cross-DDC level (see Section 6).

We recommend the following activities for effective pre-discovery pipeline management:

1. Business groups propose projects for DDC to deliver in periodic programme board meeting (or alternative governance group).
2. High level evaluation of idea, including light-touch user needs analysis conducted based on available data/insights. This will establish what the user need is for the service. High level estimate and very high level solution design diagram developed.
3. Agree at programme board that the idea is 'digital' and falls within scope of DDC.
4. Business groups prioritise their digital services based on pre-defined criteria (e.g. value).
5. Solution agreed by architecture group (or alternative owners of IT solutions in organisation).
6. Secure programme board approval to proceed following light-touch evaluation.
7. Seek funding from Investment committee/funding source for discovery and alpha at a minimum. Avoid funding requests for one development phase at a time as this can delay delivery. You may also find it useful to agree programme-level funding of all digital delivery projects that do not exceed a certain value/number of users. This will limit the number of time-consuming funding requests you have to make.

#### **4.14.2 Prioritising work during delivery**

Following pre-discovery approval to proceed with delivery, appoint a product owner (if they do not currently exist). The product owner will oversee full end-to-end delivery of the service, including management of requirements in a product backlog of user stories. During discovery the Scrum team, led by the product owner, will define the MVP for the service. While the finer details of the MVP may change throughout delivery, these should be the highest priority requirements.

At the start of each sprint, the product owner will identify the requirements that they want the team to deliver. The team will then decide how best to deliver those requirements in the sprint. At the end of each sprint during the show and tell, the product owner will sign off on the requirements that have been delivered and re-prioritise the list of requirements in the product backlog with a view to achieving the MVP. Once the MVP has been delivered, the team will deliver the other, lower-priority items on the product backlog, in line with

the product owner's prioritisation. To support work prioritisation, it is important to define a realistic MVP. Given that requirements change in agile, your MVP should not typically exceed 70% of requirements that can be delivered in the release (and in most cases will be much lower). The objective is to release the MVP as soon as possible.

The product owner should consider not only the MVP, but also the ongoing operability of the service. Release of functionality in itself is not an outcome. The focus has to be on delivering and increasing the value of the service to end users. Live operations and development activities should be prioritised accordingly.

#### **4.14.3 Operational prioritisation of work of teams within the DDC**

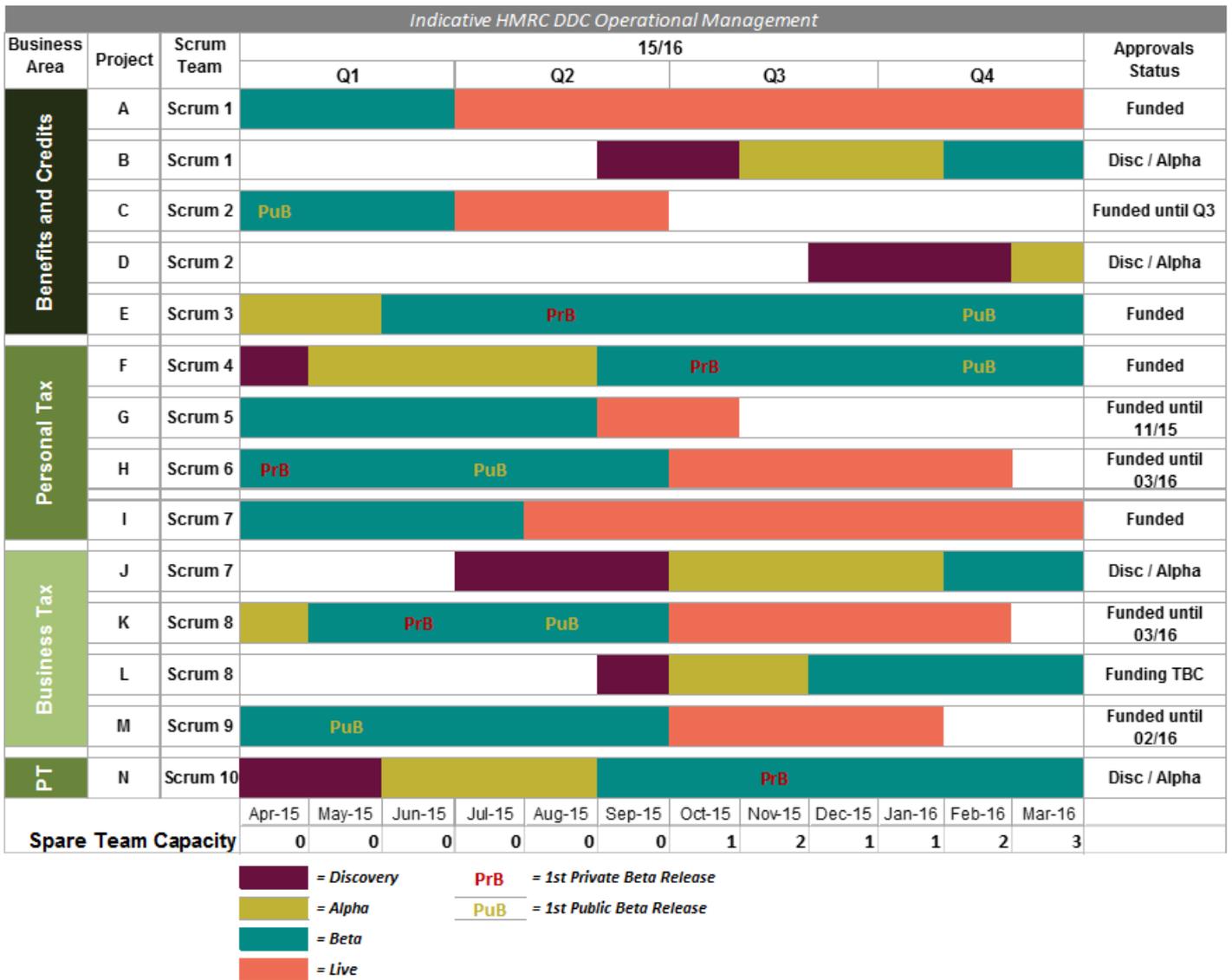
The DDC Operations Managers are responsible for overseeing delivery across all DDC teams. They also allocate work to teams within the DDC based on pre-discovery approval to proceed. Operations Managers at HMRC as far as possible allocate projects to DSMs and Scrum teams with experience in the relevant business area and Heads of Duty systems. This helps build expertise in subject matter and the technology estate. They use the estimates generated in pre-discovery regarding the effort involved in delivering the service or product. They identify which team is responsible for that area of tax and check whether that team has the capacity to deliver the change. However, the decision to allocate a project to a specific DDC or team is primarily based on capacity, and movement between business areas is possible. The specialist functional leads in design, user research, architecture etc. can help to ensure development principles and standards are met and deliver a consistent quality of service, even if teams move between business areas.

*HMRC teams conform to service-agnostic digital platform standards and principles in Design, User Research, Architecture and Development. Functional Leads / authorities oversee each specialist area. This enables teams to move between business areas while continuing to deliver high quality services.*

During DDC mobilisation, while development infrastructure and delivery processes are evolving, your operations managers (informed by DevOps) should make priority decisions on which teams should have access to environments if space is limited. The same applies for test data and any other operational process priority calls.

Figure 27 is an **indicative** booking chart used by HMRC's operations management to track delivery timelines, funding approvals and team capacity. You can see that Scrum teams can potentially work on more than one service at the same time, for example to continue some limited development and management

of a live service, while also commencing development of a new digital service (see teams 1, 2, 7 and 8 below). In addition, this demonstrates that although HMRC align teams to areas of the business, they may be asked to work in different business areas depending on work prioritisation and team capacity. In the example below, Team 7 is simultaneously managing a live personal tax service, while also developing a new business tax service.



**Figure 27 – Indicative operations management team booking chart**

## 5.0 OPERATING AND CONTINUALLY IMPROVING A DIGITAL DELIVERY CENTRE

This paper outlines the steps needed to develop and scale digital delivery in an organisation. Whereas Chapter 4 outlined the priority activities needed to mobilise and start delivering quality digital services at scale, this chapter looks at what else is needed to operate and continually improve a DDC. The areas identified in Chapter 4 will remain important as the DDC scales and stabilises. However, after addressing those priority activities, some focus should turn to the questions discussed below. Digital service delivery is highly flexible, involving a series of behaviours, practices and approaches that can change and adapt to each organisation's requirements and capabilities. As such, some organisations may approach the activities in this paper in a different order. However, whichever order you follow, developing a culture of continuous improvement is a key part of successful digital delivery and should be considered throughout the life of any DDC.

### 5.1 Continuous improvement

Inevitably, issues affecting delivery will arise when you establish a new DDC with a large number of new people using new processes and new technology in a new environment. A central continuous improvement team can help with teething issues. This team are responsible for:

- identifying issues and risks that impact team delivery and the operation of the centre
- designing and implementing mitigating solutions
- delivering continuous improvement solutions to increase the quality of services and service delivery activities
- setting standards and ensuring consistency and quality of services (e.g. design, architecture, development).

In addition to the dedicated continuous improvement team, everyone in the DDC should take responsibility for continuously improving the centre. This will shape a DDC that best fits its team members.

#### 5.1.1 DDC continuous improvement team overview

Establishing the initial DDC teams and demonstrating that they can deliver can help build DDC mobilisation momentum and buy-in. It also provides the opportunity to test the approach and to learn lessons, with support from the Mobilisation Team. Yet, while an aggressive ramp up of the initial teams can be beneficial, people might find a number of common issues affecting their delivery (such as process, governance, tools, infrastructure, team forming, learning and more).

Rather than continuing to ramp up at a constant pace, we recommend pausing or reducing the speed of this. You should start a stabilisation and continuous improvement programme, led by a dedicated continuous improvement team. They'll be responsible for identifying delivery issues and risks, and for designing and implementing improvements to benefit all delivery teams. Some of the ways we've identified these issues include floor-walking and spending time with individual teams, attending Scrum-of-Scrums, attending show and tells, establishing communities, conducting DDC surveys and communications.

When the DDC reaches capacity and initial delivery issues have been resolved, the continuous improvement team should continue to operate, leading initiatives to improve the ways of working and delivery processes. The team oversees the design and development of all digital development activity. This involves setting standards and ensuring consistency and quality of service design, user research, architecture, development and other activities.

The continuous improvement team develop their own product backlog of issues and corresponding solutions, and manage this in much the same way as other agile deliveries (i.e. re-prioritisation, Stand Ups etc.) The continuous improvement team should possess a similar skill-set to the Mobilisation Team (they may even include some of the same people). In addition, any functional leads in the centre (e.g. head of design, lead Scrum master, lead architect) should form part of the continuous improvement team. These

*HMRC DDCN focus continuous improvement backlog user stories on the following “users”:*

- *The individual*
- *The team*
- *The community*
- *The DDCN*
- *The wider HMRC.*

functional leads may dedicate varying degrees of their time to continuous improvement activities, depending on their service delivery commitments, but they should allocate some time to focus it.

Finally, the continuous improvement team should be supported by every individual working in the DDC. Team members at every level should be part of building and shaping the DDC in order to foster the collaborative, solution-oriented and flexible culture needed for successful agile delivery. Teams and communities should be involved in the continuous improvement activities, encouraging input and

allocating ownership of individual items. We recommend that the continuous improvement team possess some or all of the following skills and experience:

- DDC functional leadership to set DDC-wide standards and principles (architecture, Scrum master, design, user research, development and testing)
- Technical digital development
- Business change
- Delivery management
- Agile delivery
- Business transformation experience including operating model design, process optimisation, capability development, change management
- Learning and training
- Support from **everyone** in the DDC – team members must retain ownership for shaping their DDC.

There is a key difference between the operations management team and continuous improvement team. The operations management team focus on the **development of the service/product**, overseeing delivery progress and collaborating with the team. In contrast, the continuous improvement team focus on the **development of the team**. They seek to provide all the support and capabilities that teams need to be as effective as possible, thus supporting an efficient DDC and high-quality digital deliveries.

### 5.1.2 Continuous improvement team areas of focus

The specific activities carried out by the continuous improvement team depend on the major issues impacting delivery at each DDC. The table below outlines some areas in which continuous improvement teams have helped HMRC and other projects.

Area	Continuous improvement action	Outcome
Standards and principles	Set delivery standards for all services e.g. design, user research, development, architecture. Where there are multiple DDCs, standards will be set by cross-DDC Functional Leads in a centralised continuous improvement team. DDC-specific Functional Leads ensure these standards are met in their local DDC.	Consistent, high quality services that have the same look and feel, shared components, and assurance levels.
Reporting	Development of reporting processes and metrics that transition from Waterfall approaches and align with agile metrics, ways of working and required business value.	Governance metrics that deliver real business value, enable effective decision making and limit time spent by teams developing additional metrics.
Process embedding	Continuous improvement and iteration of process documentation. Working with team members to reflect learnings in processes, and leading dedicated process walk-throughs with teams. Hand over ownership to Delivery Team Process Champions.	Consistent understanding of end-to-end delivery process and ways of working across all teams. Process ownership with Delivery Teams to ensure they shape their way of working.

Deployment pipeline	Collaborate with delivery teams and DevOps to design the deployment pipeline. Document in centralised Technical Reference Guide, including key activities at each stage. Embed this process in each Scrum team.	Clear process to develop and deploy software in an agile environment. Shared understanding of deployment pipeline activities and tooling across all teams.
Agile development approach	Work with Scrum masters to identify development approaches that can be improved e.g. test-driven development, pair programming, code quality practices, acceptance criteria. Work with teams to design and implement initiatives to address areas for development e.g. brown bags, guidance, training, coaching.	Increased utilisation of agile best practice development techniques. This will build capability and confidence of team members, and ultimately lead to better quality software.
Learning and training	Identification key skills to develop across the DDC e.g. through the use of a digital competency model. Design and implement learning initiatives to build competency levels. Support / lead the implementation of an effective capability development framework.	Increased proficiency in skills, behaviours and technologies that were hindering delivery progress / software quality. Establishment of a structured approach to learning and capability development.
Delivery management	Implementation of reporting and delivery management activities and metrics that align to the agile way of working. Support new teams as they are evolving, and provide the insights management need without impacting team delivery activities.	Increased delivery efficiency, clarity on delivery progress, de-risked delivery and improved decision-making.
Test approach	Define and document a consistent approach to testing for all DDC teams. This includes test environments, test activities, tooling, data, test management / oversight.	Shared understanding of all the information needed to effectively and efficiently test and assure new digital services.
Communities	Implementation of role-specific communities e.g. test community, product owner community. Communities meet regularly to share information and discuss issues impacting multiple teams.	Economies of scale provided without sacrificing Scrum team autonomy. Information, best practice and learning shared across teams. Specialist issues addressed by 'the experts'.

### 5.1.3 Further expansion v internal capability uplift

We recommend a dedicated continuous improvement capability for all new DDCs. Improving the efficiency of the delivery approach will benefit all teams. As your DDC begins to stabilise, focus can shift to:

- further ramp up and expansion (possibly even to a new DDC), and/or
- increased internal capability and reduced reliance on suppliers.

This decision will inform the direction and activities of the continuous improvement team.

## 5.2 User experience

In section 4.2 we highlighted some challenges associated with moving to user-focused service delivery. This section outlines the activities and considerations needed to deliver a great user experience in digital services.

### **5.2.1 User experience (UX) professionals in the agile team**

Building UX capability within the DDC should be considered from the moment that teams are formed. The delivery teams must have sufficient UX experience to support better design decisions, and ultimately, a better UX. Consider rotating UX trained resources through the delivery teams to spread expertise and experience across the DDC.

UX designers and researchers should be embedded into the sprint (even if not full-time team members). UX specialists will use the sprint time to prepare their tests and processes. Trying to get developers or QAs to build in UX specific activities may not be realistic if they already have a full load of development tasks for each sprint. However, you should encourage all team members to consider UX and support UX designers and Researchers throughout delivery.

Sprint teams run to aggressive schedules, with a focus on developing working software. Successful UX requires the development team to allow the UX members to assist in tasks like design reviews, Scrum meetings, and the design of the User Interface (UI.) The team need to understand the role of UX in the project. Try to establish collaborative working relationships where the UX resources are consulted and engaged with the development team.

### **5.2.2 Delivering a high-quality user experience**

User research should be planned into the sprint cycle, with tasks set for each sprint that work towards getting that input. Lean UX techniques like wire-framing and sketching are very useful in bringing requirements to life, and capturing feedback from the end user. Teams should focus on developing prototypes that can be tested on end users to get their feedback as early as possible. Agile teams need to understand that it doesn't need to be a polished end product in order to get user feedback. We recommend getting user feedback early and often.

Collaborative, engaged UX resources in a team help to drive quality decisions and set quality standards with the development team. You should agree design standards in order to move from a developer-centric to a user-centric method of working. For example, whilst a developer may think the product is ready, the UX designer should verify the product meets their expected standards. Therefore it is very important that agile teams engage their UX resources at the product/project strategy stage. All members of the team should understand the required UX quality and what a final product should look like. Employing a design lead/authority to oversee design activities, and set common standards and design principles will help to ensure a high-quality, consistent UX for all new digital services.

Scrum masters can also support high quality UX. Firstly, they need to create a joint expectation of quality on the project, and the UX exit criteria for the product. Secondly, they should ensure that the UX specialists are working collaboratively with the developers throughout the sprint. Co-location helps, but irrespective of the communication channels, UX resources need to engage with the development team continually. Thirdly, UX goals need to be set for the team, planning when customer feedback and engagement is to be used. Connecting with the customer throughout the development lifecycle, and having the UX team members drive that engagement with the Scrum team, helps build a successful team.

## **5.3 Integration between digital and traditional delivery methods**

Digital service deliveries are easier to coordinate if all components of the end-to-end service adopt the same delivery methodology. As we have seen, agile is an effective way to deliver public-facing digital services<sup>23</sup>. However, existing commercials, governance, business criticality and software architecture mean that full end-to-end agile delivery may not be possible from the outset. Aligning different product design and delivery approaches is challenging.

---

<sup>23</sup> For more information on the UK government's view of agile, see <https://www.gov.uk/government/publications/the-green-book-appraisal-and-evaluation-in-central-government/agile-digital-and-it-projects-clarification-of-business-case-guidance> and <https://www.gov.uk/service-manual>

Many public sector Heads of Duty Systems follow Waterfall development approaches, receiving system requirements via the business interpretation of policy makers. Once the full scope of the project is understood, the work to deliver the project is organised sequentially. Changes to the plan are costly and disruptive.

Using agile, DDCs embrace change and avoid complications when requirements change. However, this flexibility and the speed of change must be carefully managed. Without large up-front design, teams are learning and experimenting with the application architecture and solutions. Refactoring the code is a natural part of the process to ensure quality, keeping the application current and removing technical debt. As this flexibility is visible to stakeholders, agile teams can, if not supported, find themselves adding features as a priority over refactoring. This results in unmanageable code, a decline in productivity and an increase in quality issues. Both methodologies have challenges which need careful management and understanding.

Integrating agile value principles and practices in a Waterfall setting can be successfully achieved provided:

- the difference in approach between project management and product management is understood<sup>24</sup>
- a plan is developed to clarify the boundaries between these two things, and
- management and teams communicate every step of the way.

There is no easy fix to bring together two opposing methods. The integration needs to be tailored around a good understanding of programme reporting needs, fixed delivery constraints, the flexibility of the development environment and capabilities of the delivery teams.

There are three elements to consider when integrating agile and Waterfall projects:

1. Enterprise-level project integration
2. Ongoing delivery integration and dependency management
3. Predictable core delivery dependencies.

### **5.3.1 Enterprise-level project integration**

We recommend appointing a project manager with overall responsibility for coordinating multi-team deliveries involving agile and Waterfall components. They're responsible for programme management, enterprise release planning and reporting. They should resolve programme and portfolio level issues without impacting the delivery team as far as possible. They should try to resolve enterprise level issues with your DDC management ahead of the team. When addressing cross-project issues, it may be necessary to work with the Scrum master to define the dependency and articulate the impacts. However, it is important to never let this affect the agile delivery team. If you need to involve the agile team, wait until sprint planning rather than disrupt software development.

The project manager should report the project's status to enterprise portfolio management, or other governance groups. They must act as a translator, converting the agile metrics into the accepted reporting structure of your organisation if these don't align. The focus in agile on transparency makes it easy for the project manager to find out which reporting structure is needed, including Earned Value<sup>25</sup>. As maturity develops, you should try to modify reporting structures to include some agile metrics. The project manager should work with the project sponsors to look after the scope, schedule and budget of the project, among other things.

Deploying software developed by the agile team in the enterprise production system can be challenging. The project manager, Scrum master and lead developer should work together to plan for their release. The ultimate goal of an agile team is to make deployment seamless so that it has little or no impact on the delivery team. It is likely, however, that they'll have to devote some part of their iteration to the planning, deployment and checkout of the release. If possible, a single person should be given this responsibility, with

---

<sup>24</sup> For example, there is likely to be more cost of change of requirements/scope for the Waterfall project than the agile project

<sup>25</sup> See Section 4.6. For more information on Earned Value as an agile reporting metric, see Suliaman, 2007; Rusk, 2009

the remaining team allocating a small portion of reserve time (reducing their time available) for the iteration so any defects can be resolved swiftly.

### **5.3.2 Ongoing delivery integration and dependency management**

The ideal situation for agile projects is an autonomous Scrum team without dependencies. Although a dependency-free situation is rare, the aim is to reduce these. There are a number of techniques you can use to effectively manage dependencies with other agile or Waterfall teams.

During agile development, the Scrum master is the first person you go to resolve any issues. They're responsible for intra-team communication (be that with an agile or Waterfall team), for coaching the delivery team, and for helping prioritise work when the team is facing challenges. Generally the Scrum master is responsible for the health and wellbeing of the agile team. Delivery impediments that the Scrum master cannot resolve for the team are escalated to the project manager. The project manager also acts as a protector for the team, keeping the complexities of the outside organisation from impeding progress. To achieve this delicate balance, the Scrum master and project manager need to have a very good working relationship.

Some common challenges agile teams have when working with interdependent Waterfall teams include:

- Test environment availability and the demands to align testing with a Waterfall delivery
- Test data provisioning from a head of duty system to support the Sprint
- External stakeholders unused to agile ways of working and working with the team (e.g. Through product owner)
- Integration details such as API schemas and functionality delivered with limited time to test and implement
- End-to-end support from other teams to confirm outcomes, reset data and/or services

Given the challenges, it is essential for successful delivery that the developers and QAs in the agile teams work closely with their counterparts in dependent teams (irrespective of their delivery approach) to help prioritise activities and implement process improvements.

The Scrum master and the project manager should work together to alleviate these challenges and coordinate agile deliveries with dependent teams (agile or waterfall) through a number of forums:

- Joint sprint planning (with attendance from a representative of the waterfall team)
- Iteration coordination
- Joint show and tells
- Joint retrospectives
- Scrum of scrums

In particular, we recommend using a Scrum of Scrums where dependencies exist to ensure that teams work together and prioritise activities jointly. It should be held at least weekly, and more often as required.

Suggestions to help you plan and deliver an effective Scrum of Scrums meeting include:

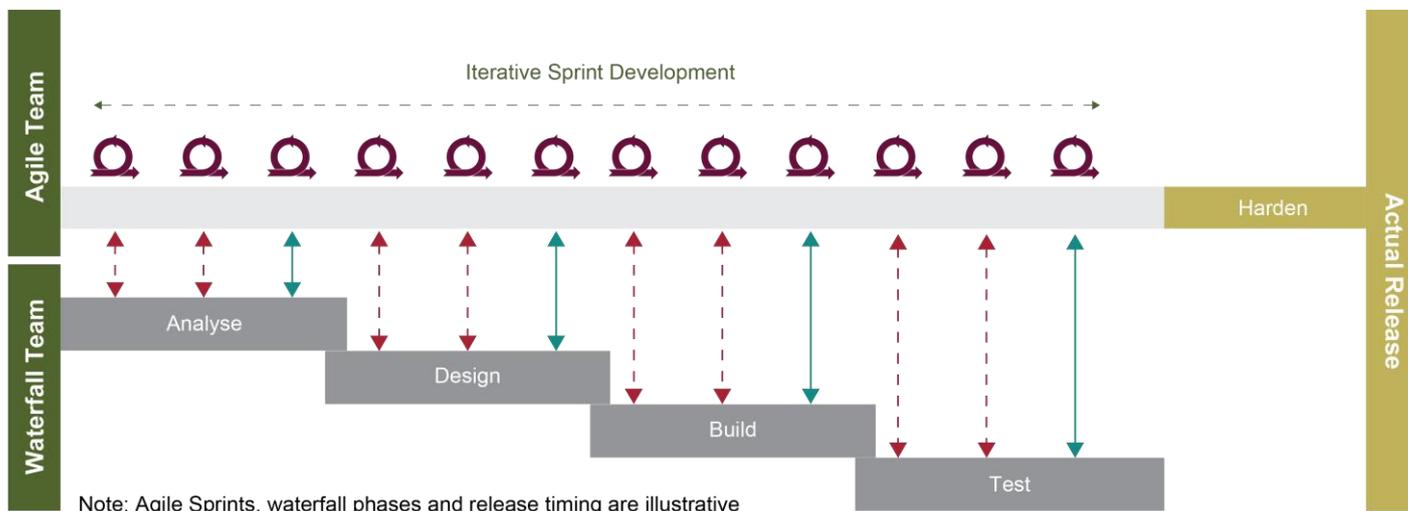
1. During sprint planning, identify the features that need contributions from several teams during the upcoming sprint (e.g. with a dependency matrix).
2. Focus the Scrum of Scrums meeting around these dependencies. Dismiss the Scrum of Scrums if there are no relevant dependencies (this is a good thing).
3. Teams with dependencies should send one or two delegates to the Scrum of Scrums. The delegates should be the people with greatest knowledge of the dependencies – in many cases this may be developers rather than the product owners and Scrum masters.
4. Agree the frequency for the Scrum of Scrums. Sometimes the teams need a daily Scrum of Scrums, sometimes one or two Scrum of Scrums per week are sufficient. The teams should decide during sprint planning.
5. Limit the Scrum of Scrums to 15 minutes if it is done on a daily base (weekly Scrum of Scrums may have longer time-boxes.)
6. Visualise the state of the features with dependencies (e.g. with a dependency matrix.)
7. Let one of the team Scrum masters moderate the Scrum of Scrums.

8. Discuss the following three questions (similar to the one used in the daily Scrum):
  - *What did my team achieve since the last Scrum of Scrums regarding the features with dependencies?*
  - *What impediments occurred in my team regarding the features with dependencies?*
  - *What do the teams plan to do until the next Scrum of Scrums regarding the features with dependencies?*

You can also use the following other approaches to manage agile and waterfall delivery dependencies:

- Face-to-face Inception event at start of the project for all dependent teams involved in delivery - this will define roles and responsibilities, building understanding of each team's ways of working, challenges and constraints
- Clearly document how and when the teams will:
  - Identify and notify the other of a dependency
  - Agree how the dependency will be met
  - Provide continuous visibility of progress e.g. Through shared jira boards, jenkins dashboards and confluence pages
- Attend the regular daily Scrums of dependent teams
- Pair cross-team developers during development of the integrating components, or have each team develop the testing stub for the other team
- Early testing of dependencies through the use of shared testing environments and early integration testing
- Cross team review of acceptance tests (combined with behaviour or test-driven development, this enables early identification of misunderstandings between teams delivering either side of an interface)
- Appoint a member of each team with responsibility to ensure their team learn about the other, e.g. The legacy IT team learn agile ways of working from the other and look for opportunities to apply relevant practices to their work
- Teams may form 'Team of Team' task forces for features with dependencies - this involves bringing together the relevant team members to collaborate and jointly solve a common issue; you can establish a Scrum of Scrums for each task force; these are often starting points for reorganising team structures to improve team autonomy.

Figure 28 below demonstrates typical high level interaction patterns for dependent agile and waterfall teams. It shows the level of integration and joint working varies based on the level of interdependency between teams. At a minimum, teams need to interface and integrate while working towards a common large release. Where low levels of dependency exist, some interaction is required at key milestones, such as the end of a Waterfall phase. High levels of dependency mean there must be frequent interaction between the teams. Using some of the techniques above during each sprint helps you to effectively manage high levels of dependency.



Scenario and Key	Description
Independent Team Interactions	Mostly independent teams with few dependencies may come together at the end before a common release or release independently depending on end user impacts.
Low Dependency Team Interactions	Teams with low dependencies should synchronise at key milestones to ensure the proper integration of components, in addition to the interactions in scenario 1.
Independent Team Interactions	Teams with high dependencies should synchronise at the team level on every Sprint boundary at a minimum, in addition to the interactions in scenario 1 and 2.

**Figure 28 – Coexistence of agile and waterfall delivery teams**

### 5.3.3 Predictable core delivery dependencies

“Predictable core delivery dependencies” refers to the planning, management and delivery of any project components that an agile delivery team needs to be in place to start delivery. These are the things that must be established between project approval at portfolio level and allocation to a DDC, and the point at which the delivery team can start. One prime example is getting new hardware or providing environments for the project. It is not reasonable to expect that new hardware and environments can be ordered at the start of an iteration and installed during it, especially if the iteration lasts only one or two weeks. Where longer lead times are required for these predictable elements, you should allocate sufficient delivery time in advance of the sprints for which they are required. Other examples include software licenses, printed materials, office or estate components, and legacy components being delivered under Waterfall.

In most of these cases, it is likely that these predictable components will be delivered by a different team than the DDC delivery team. We recommend using the techniques in sections 5.3.1 and 5.3.2 to manage these dependencies and align the teams.

## 5.4 Supplier management

If you choose to work with suppliers of experienced digital practitioners in the DDC, an effective management approach is needed. This becomes even more important where partnerships with multiple suppliers exist across a number of sites.

### 5.4.1 Working in partnership

Forming a cohesive, self-managing delivery team is key to success in agile projects. Accordingly, at a team level, we recommend building a culture where suppliers are seen as partners rather than the traditional client-supplier relationship. Delivery teams should be integrated, working towards a common goal. You should remove any distinction between supplier and permanent DDC resources in the Scrum team itself

(notwithstanding the product owner who sets team direction, and the DDC management being in place for any escalation points). You can build this partnership by:

- Supplier and permanent resources working in each team
- Common escalation points
- An integrated management team in the ddc, with the presence of both suppliers and permanent resources
- Shared communications, best practice, meetings, training and communities
- Agreed terms of reference and ways of working.

Work with suppliers who understand the vision of the centre and what it is trying to achieve. Instilling a flexible and integrated approach to delivery will help foster the culture and collaborative ways of working that support digital delivery. You should recognise that suppliers will be learning too, and that having everyone work towards the common goal will help achieve this vision. Employing a range of suppliers – including large consultancies, SMEs and independent contractors – can help to bring different strengths and capabilities to the DDC. Different people are attracted to different types of organisations, so a mixed group of suppliers can provide a diverse range of skills and perspectives. Experienced suppliers or independent contractors can also be well placed to support your recruitment, assessing and identifying new team members.

Certain suppliers may be more appropriate for some projects than others i.e. because a particular skill-set (e.g. UX) or toolset is needed. In such cases, the product owner should play a key role in team formation, working with DDC management to agree which suppliers to staff in the team.

#### **5.4.2 Operational management of suppliers**

Some key activities to consider to structure the supplier relationship and maximise the value that suppliers can provide include:

- Developing a ramp-up and ramp-down plan, defining supplier roles to fill and how this will evolve over time
- Documenting expectations of capability development of ddc staff; implement techniques to facilitate knowledge transfer from suppliers to internal resources
- working together on recruitment - suppliers can help recruit the right people, ddc management should keep the supplier updated on changing resourcing requirements
- Holding weekly status meetings to discuss delivery progress, and identify any areas or team members that need greater support
- Agreeing day rates and how they will change over time - regular commercial meetings to discuss progress and validate ongoing commercial plans.

*HMRC found that using a Time and Materials commercial model provided more flexibility to assure the progress of supplier team members and identify people who need support, jointly agreeing steps to take to improve performance.*

### **5.5 Scaled platform vs. service model**

This section compares and contrasts the creation of a scalable platform rather than the creation of a range of discrete services. Questions covered include:

- When does it make sense to create a platform rather than building individual services?
- What pre-existing IP and framework architecture can be used?
- How does the platform evolve to accommodate changing requirements?

#### **5.5.1 What is a platform rather than a service?**

As an increasing proportion of customer services move to digital channels, it can be useful to establish common, consistent services, on a reusable and scalable environment which supports further development of services. Organisations want DDCs to conform to common platform standards in design, architecture, development and more, rather than approaching service delivery activities wholly independently.

This contrasts with the creation of individual services, typically fulfilling specific business functions. These functions may be reusable (e.g. a common API to check the validity of a car registration) but without an accompanying environment which can be further developed on, this does not constitute a platform.

A crude analogy would be the Apple Ecosystem, with individual apps being the equivalent of services, and the Store and Apple Development environment being the platform.

### **5.5.2 The benefits of using a platform**

When establishing a DDC you should consider a sensible degree of standardisation and reuse when developing applications. If the intention is to gradually increase the proportion of applications that will be developed, and you want a common look and feel, and architectural compliance, then a platform may be sensible. This is further reinforced if your strategy is to extend that platform to other related organisations and encourage them to develop using the same tools and methods.

The creation of a platform requires articulation of the development architecture, including the software development tools, standards, and architectural compliance that will be enforced. This doesn't mean you have to stop experimenting and piloting, but you do need this to help create applications and services that are consistent and compatible.

The consistency helps reduce the time taken to develop new applications, as developers get more familiar with using the same tools and environment, and the library of reusable components and services increases.

Platform-level leads/authorities in design or architecture as part of the continuous improvement team should help instil the reuse of existing components. They can also identify the creation of new ones, and enforce and update the consistent development tools, software products, and methods in use within the DDC. Service teams need to adhere to these standards. By 'coding for the wider good', they are displaying best practice as their work can be reused by other teams.

### **5.5.3 Re-use of existing IP and architecture**

As part of the mobilisation phase of a DDC, it's useful to have an initial inventory of existing tools and software components. You can use this to identify whether prior investment can be reused and applied within the DDC.

There are also significant external sources of IP which you can exploit. Resources such as the Open Source Foundation and the Scrum Alliance can help identify collateral, as can Government Digital Services.

There may also be opportunities to share platforms with sister organisations (e.g. arm's length bodies associated with government departments) or other enterprises. This is particularly true where there is significant overlap in the customer base and demographics of both organisations.

### **5.5.4 How does the platform evolve to meet changing requirements**

The more you use a platform, the more you need to accommodate the number of, sometimes conflicting, requirements it needs. Changing the technology products, standards, tools, and methods can have a profound impact on the developer and software base of the platform. This increases the importance of cross-team architectural leadership. You will need to take advantage of advances in the technology market without suffering a drop-off in service or large swathes of services being left on redundant technology.

Speeding up the development of digital services by using a digital platform means it has become even more straightforward to update or replace them, as long as they've been developed in a modular and well-architected way. As a result, you should establish an ongoing refresh programme, looking to review and replace services where there is a business case to do so (including cost of maintenance, and availability of suitable skilled resources).

One key development standard that supports effective scaling of a platform has been in ensuring that APIs are backwards compatible. This means that services coded for one version of the API will continue to function, without adjustment, with future versions of the API. This needs some extra initial development work from the team managing the API. However, if an incremental improvement in a service requires changing an

existing API, this won't affect all other services connected to that API. Backwards compatibility helps when scaling a digital delivery capability as the number of users of an API or service increase.

In summary, when considering the scale and scope of a DDC you should:

- identify how widespread the digital delivery will be within your organisation, and the opportunities for creating a consistent and extendable environment that may be of use by others - this will determine whether the costs and benefits justify investment in a platform rather than the creation of a series of discrete services
- Consider whether you can adopt or reuse any pre-existing platforms (or platform components), either from within the existing IT estate, or available more broadly
- Establish Functional Leads for Design, Architecture and Development within the continuous improvement team to lay out standards, tools, software, and methods; and to keep these elements maintained and refreshed as the market continues to evolve.

## 5.6 Live support model

Agile live support models vary between organisations. The most common approaches include:

1. The Scrum team continues to develop and maintain the service once it is live, even if development activity is minimal and they have moved on to a new development project. This approach is most suitable where a Scrum team is responsible for one service, or a series of closely related services. Development work focuses on expanding and iterating the core service.
2. The Scrum team hands over operation of the service to a dedicated Live Service Management Team, who are responsible for any subsequent development and maintenance of the service. Handovers should be limited in agile, and need to be managed carefully, involving a sufficient transition period.
3. The most mature DevOps organisations create platforms that provide operations with functions as services that Scrum teams can use to build and manage their services. These services apply automated security models, create environments and pipelines with appropriate management tools and interfaces. The operations function moves to a support team which looks after the platform's ongoing development, while also providing expertise on infrastructure, modelling, scaling and environment configuration. Scrum teams own their services using the operations services platform with the operations team as back-up.

HMRC uses a hybrid of options one and two outlined above. It distinguishes between live services that are active, maintained and retired, as indicated in Figure 29.

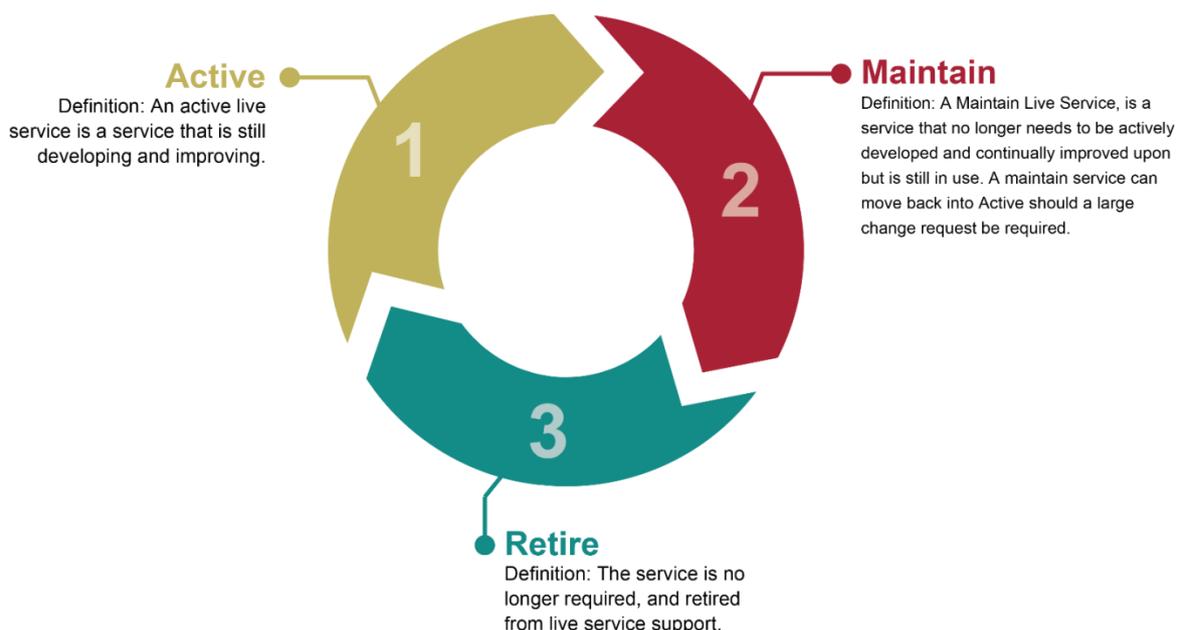


Figure 29: Lifecycle of a DDC Live Service

In this model, the delivery team continues to iteratively improve the service in beta and live. It reacts to evolving user needs and demands, and seeks to meet performance targets (key performance indicators) set during development. This is referred to as the **Active** Live Service. Once the service has reached a level of quality where significant changes to functionality and improvements are minimal, and less important than new incoming work, the service is passed onto a dedicated live support function. This monitors, maintains and conducts any small scale development on the service, ensuring it remains secure while in use (**Maintain** Live Service). Eventually, the service may be retired. User needs must still be met during any period of transition (**Retired** Live Service).

### 5.6.1 “Active” Live Services

*HMRC found that where core platform services are built, it can be helpful to have an over-sized Scrum team working on active live support. This provides increased capacity to manage incidents without hindering continual live service development.*

An Active service is still going through high levels of development e.g. throughout the beta phase. Support for this service should stay within the Digital Service Delivery Team, with new features, functionality and incident fixes forming part of the product backlog. It is continually monitored through support tickets, analytics and user research. The product owner and Scrum master prioritise both delivery and support Backlog items. Following Go Live, the team could have a 'Technical Debt' sprint to refactor, optimise and clean up any technical debt that has accrued as a result of business pressures to deliver (for example).

Teams actively monitor their services throughout beta and live (in conjunction with DevOps) until responsibility formally passes to a dedicated “non-development” live service Scrum team.

If following this model, encourage your delivery teams to adopt the mind-set that they are responsible for both the operation and performance of the Active Live Service. Any central DevOps team would most likely take responsibility for monitoring the operation of the platform as a whole. However, given the potentially high numbers of Active Live Services at any one time, it is more effective for the delivery team to lead operations and support activities for their Active Live Service, in addition to development. This would include any out of hours support.

### 5.6.2 “Maintain” Live Services

There is value in one Scrum team continuing to manage their live service indefinitely. However, in some organisations, this may be challenging if teams have to go on to develop a completely different service. In such situations, possibly involving different product owners, it is difficult to prioritise the work of the team effectively. HMRC has decided that, once a live service no longer needs to be actively developed and continually improved upon, it should be maintained by a dedicated Live Service Team.

Scrum development teams across their DDCs temporarily perform the role of the Live Service Team, or ‘non-development’ team on a rota basis. Once it’s decided to move a service into Maintain Live Support, the development team can move temporarily into a ‘non-development’ role. This is beneficial as they already have deep knowledge of the service, so are well placed to oversee its initial period as a Maintain Service, before handing over its live service management to another Scrum team. If it’s been decided that the development team will move directly onto another project, they should conduct a structured transition with the designated ‘non-development’ team that will take ownership of the service.

Maintain services are actively monitored through support tickets and analytics by the dedicated ‘non development’ Scrum team. Expertise of the original team can be called on where necessary. The service can return to active mode if a large change to the service is required. In such cases it would be treated like any other new project in the DDC.

### 5.6.3 “Retired” Live Services

Changes in policy may mean that a particular digital service is no longer needed, or new understanding may mean it can be provided through a different service. Digital services are aligned to user needs, so you should agree how these needs will be addressed once the service has been retired. Ensure the team provide details of what the change is, why it is being made, what users will need to do, and what will happen to their data.

## 5.7 Open source technology

You should look for appropriate open source languages and tools to avoid being tied into costly license agreements. Applications and services are developed in a de-coupled manner in order to facilitate a simple swap out of technologies. This allows you to keep the applications current, improve the code base and reduce license costs once an appropriate alternative open source solution has been identified. The use of open source is clearly articulated in the government's IT Strategy and Technology code of practice.<sup>26</sup>

A DDC should also consider contributing to the open source community by publishing the code it develops publically, for example in public instances of its code repository (such as GitHub). This is known as 'Coding in the Open'. HMRC, for example, has published different source code repositories – libraries, front-ends, services - on public GitHub<sup>27</sup>. When a repository contains security-critical code and/or configuration it should be changed to be as small a component as possible, with the non-critical majority open-sourced as a separate repository.<sup>28</sup>

'Coding in the Open' also encourages better quality code. For example, development teams need to separate code for sensitive functionality, such as security rules, from other code. This encourages good-practice modular development, making separate modules of code easier to understand, test and refactor independently of others.

Setting platform-level development standards will help ensure a greater number of your digital services and libraries can be publically published. If services are built to the standards and principles defined by design and development authorities, then it should be possible to publish many of them.

The HMRC DDCs are taking open source to its natural conclusion. Teams are now moving all development to 'Coding in the Open' which gives the tax-paying public visibility and access to the code we're writing as we write it. The open source community and connected businesses can now benefit from the volume of code contributed to public GitHub.

## 5.8 Security of digital services

Digital services need to be securely designed, developed and tested in order to protect them from both cyber-attacks and illegitimate use. The responsibility for maintaining service security sits both with any central platform or infrastructure teams (e.g. DevOps), and also on a delivery team level.

### 5.8.1 Cyber attacks from hackers

If your organisation is using a common digital platform to host digital services, then the security measures established as part of platform development should provide protection from cyber attacks for any services released on the platform. This should include protection from both untargeted and targeted attacks, such as Distributed Denial of Service (DDOS) attacks<sup>29</sup>. At HMRC, for example, it is the responsibility of the DevOps and platform teams to maintain platform-level security. If services developed at the individual DDCs adhere to the platform-level design, development and testing standards throughout delivery and before release, the service can benefit from the cyber security protection the platform provides.

### 5.8.2 Illegitimate users

Another cyber security challenge is the illegitimate use of digital services (e.g. fraud). We recommend instilling responsibility with the Scrum team for protecting the service against illegitimate use. Teams should

---

<sup>26</sup> <https://www.gov.uk/service-manual/making-software/open-source.html>

<sup>27</sup> <http://github.com/>

<sup>28</sup> For more information on Coding in the Open, see <http://hmrc.github.io/coding-in-the-open-manual/>

<sup>29</sup> [https://www.gov.uk/government/uploads/system/uploads/attachment\\_data/file/400106/Common\\_Cyber\\_Attacks-Reducing\\_The\\_Impact.pdf](https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/400106/Common_Cyber_Attacks-Reducing_The_Impact.pdf)

develop the service in such a manner that enables effective risk analysis. The team are responsible for monitoring their service and adapting it accordingly. Relying on a central DevOps team to monitor the service for illegitimate use can prove more challenging when they have multiple simultaneous services to monitor. Teams should consider what information they need to capture in order to identify and prevent fraud and illegitimate use. From its first release, and throughout service operation, teams should monitor service transactions, dropping the relevant data, such as API requests and responses, into audit logs. Ensure that a sufficient volume of data is captured this way to support and protect legitimate customer behaviour.

Whether measures are at the platform or service level, you should consider how services can be effectively secured without adding unnecessary and time-consuming governance for teams. As the GDS manual states, “Audit and verification of user behaviour should be used to ensure policy compliance instead of preventative measures which add cost and degrade productivity”.<sup>30</sup>

### **5.8.3 An effective approach to risk management**

Each digital service should have a designated risk owner, responsible for the operation of the service. When this person is the risk owner for a very complex system, with high levels of ongoing development, or a number of individual services, it can be difficult to have a clear view of all the possible risks, as well as the effectiveness of all the security activities. For example, while a risk owner may sign off a penetration testing activity, are they always confident it has covered all the possible threats. Do they always fully understand what has been tested?

You should engage the risk owner and team conducting the technical security activities early in the project. Focus the conversation on the outcomes and objectives for security activity e.g. rather than initially focusing on the volume of penetration tests to conduct, try to define the potential risks to the service, and agree on the most suitable measures and activities to mitigate them.

We recommend using experts from CESG Certified Cyber Security Consultancies<sup>31</sup> to help drive these conversations and structure your security activities. At HMRC, these have effectively brokered conversations between risk owners and technical teams. They can support the definition of business and service risks, and advise you on the appropriate security. Similarly, they can translate technical activities and outputs into business outcomes. This provides a more reasoned and informed approach to risk management.

---

<sup>30</sup> <https://www.gov.uk/service-manual/technology/security-as-enabler.html#trust-responsible-users-audit-and-verify>

<sup>31</sup> <https://www.cesg.gov.uk/scheme/certified-cyber-consultancy>

## 6.0 MANAGING MULTIPLE DIGITAL DELIVERY CENTRES

As digital delivery becomes the standard approach to UK public service development, departments need effective, large-scale, multi-location digital delivery capabilities to transform service delivery across their organisation. In many cases, one DDC will not be enough.

Chapter 4 outlined the necessary activities and considerations to set up and operate a large scale DDC. Chapter 5 highlighted further considerations for successful operation of a DDC. This chapter describes how to scale delivery across multiple DDCs.

### 6.1 Communications

Cross-site digital delivery needs effective communication between DDCs. A number of cross-site dependencies may exist, for example shared services provided at one location, or joint prioritisation of work. Teams need a quick way of managing these dependencies without delaying the agile delivery cycles. As in Section 4.10, the lines and methods of communication can be the same (for example instant messenger) but will need to be accessible by everyone at all your sites.

The following activities can improve communications across multiple sites:

- **on-boarding** – new teams visit other sites to build relationships (such as with DevOps or Platform teams) and learn from more experienced teams
- **operational management** – regular meetings between DDC operations management to share team delivery progress, discuss capacity and share issues
- **collaboration tools** – instant messenger tools can be used to quickly and effectively resolve common or dependent delivery issues such as on common platform components; wikis are good places to publish new learning, progress or information impacting other teams
- **cross-DDC communities** – when individual DDC communities have settled in, think about setting up regular cross-centre sessions to discuss common issues and set plans and strategies for the specialist area (for example, design)
- **Q&As** – sessions to ask shared services (such as DevOps), or more experienced teams, questions that will support delivery. These can be held electronically, for example, Google Hangouts
- **shared services communications channels** – set up and document the communication channels to cross-DDC shared services such as continuous improvement teams, DevOps, Live Support, and Platform - see Sections 6.2 and 6.3 for more information on how continuous improvement and DevOps can be used to support delivery across multiple locations
- **governance** – clear and common escalation and governance routes. Representatives from each centre should attend common governance groups (See Section 6.5).

### 6.2 Centralised continuous improvement function

Maintaining consistency and quality across a multi-centred digital delivery organisation is a challenge. Having resources in different locations makes communications more complex and collaboration more challenging. This section considers how to make continuous improvement the normal way of thinking across locations and improve the quality of products, delivery processes, and consistent design, development and architecture standards across centres. The central premise is to create a cross-DDC 'continuous improvement function'. This is an extension of the DDC-specific continuous improvement team outlined in Section 5.1, and comprises a centralised Scrum team with functional leads in each of the key delivery specialities. These functional leads are responsible for driving consistency, quality and continuous improvements in all parts of the process.

The continuous improvement team should aim to make delivery teams feel empowered to drive improvements. The centralised function should provide the necessary coaching and guidance to make sure that everyone involved in DDC delivery across all locations understands that continuous improvement is one of their key responsibilities.

As with the DDC-specific continuous improvement teams, the Cross-DDC team will own their own product backlog. They complete this by identifying issues that are affecting multiple teams and/or centres.

### 6.2.1 The role of the central continuous improvement team

The following table shows the typical roles and functions needed to fulfil the cross-DDC continuous improvement function. The final column shows whether these functions could be separate for each of your centres. These people could fulfil their cross-DDC continuous improvement role part-time, with their remaining time spent in the continuous improvement team of an individual DDC. The UX lead, for example, could be the continuous improvement champion for a centre and the overall UX functional lead across centres.

Role	Function	Possibly discrete?
UX design	Overall UX lead to ensure consistency of design from all Scrum teams.	Yes
Architecture	Provides technical direction for all development teams. Owns new technical architecture components.	Yes – if the technology stack is discrete, if not then should be central.
Development lead	Quality assurance and promoter of shared code across centres. Promotion of open-source where applicable.	Yes – Developers may need different types of support, so having an onsite person to lead their Community and support learning is key.
DevOps / Tooling	Ensures appropriate use of DevOps functions (and feedback loops) across all teams. Identification and roll-out of tooling.	Not usually - A centralised DevOps function would typically sit across all centres, though there are different models. See Section 6.3 below.
Scrum master	To ensure all Scrum teams are working in the most efficient manner.	Yes – all DDCs will need support from lead Scrum masters to embed agile ways of working.
Test automation	To support adherence to automated testing best practices. Ensures availability of training for the test community (QAs, end-to-end test managers) and secures feedback.	Yes – even if centres use the same test automation tools, an onsite lead helps build the community, drive learning and maintain standards.
User research	Helps to grow a culture where user needs is the focus of all teams within all centres. Works with user researchers to implement new User Research techniques.	Yes – teams need to embed user research within the teams. More important if work allocation is different across DDCs.
Lead product owner	Product owner sitting above all individual product owners for overall prioritisation within the organisation, or within an area of the business.	Unlikely to be DDC-specific, though could be business-area specific.

One common difference between the cross-DDC continuous improvement team, and continuous improvement teams at the individual centres is that the local teams may often have a greater focus on delivery management and business change. They have a full-time, hands-on role to stabilise the DDC and embed the new ways of working. The cross-DDC continuous improvement team supports this activity but has greater focus on maintaining cross-DDC standards and principles, with their time split understandably across centres.

### 6.3 DevOps

As stated in section 4.11, a successful DevOps function is a key component to a successful DDC. Across multiple sites, you should consider how best to scale the DevOps function. There are typically four options.

1. Create a centralised function to support multiple sites.
2. Set up local DevOps functions with DevOps experts as part of the local continuous improvement function.
3. Engage with a managed service from a third-party.
4. Build a DevOps platform to allow Scrum teams to self-serve.

### 6.3.1 Selecting the correct DevOps model for distributed digital delivery

The correct DevOps model for distributed digital delivery will vary between organisations. It is important to make an informed decision and not simply expand at each location to meet demand.

The table below shows typical pros and cons of the four options. A hybrid model could also be considered. Whichever option you select, DevOps should offer both 'business as usual' support and continuous improvement (focusing on improving the tool-set and deployment pipeline process).

Option	Pros	Cons
Create a centralised function to support multiple sites	<ul style="list-style-type: none"> <li>- Central team with central control</li> <li>- Ensures consistency across delivery</li> <li>- Cost efficient (no double handling)</li> </ul>	<ul style="list-style-type: none"> <li>- No local support for troubleshooting</li> <li>- DevOps may become isolated i.e. "just a support function"</li> <li>- Requires availability of high numbers of skilled DevOps team members. If in-house this level of capability may not be available</li> </ul>
Set up local DevOps functions with DevOps experts as part of the continuous improvement function	<ul style="list-style-type: none"> <li>- Onsite support that can engage with users and troubleshoot problems</li> <li>- Build stronger relationships at each site</li> </ul>	<ul style="list-style-type: none"> <li>- Lack of centralisation could lead to divergence in usage and an inconsistent approach</li> <li>- More expensive to employ additional staff</li> <li>- Level of capability may not be available in all locations</li> </ul>
Engage with a managed service from a third party to scale easily	<ul style="list-style-type: none"> <li>- Can scale as quickly or slowly as the business requires</li> <li>- Industry experts that have significant experience</li> <li>- Off the shelf capabilities that greatly reduce development and set up costs</li> </ul>	<ul style="list-style-type: none"> <li>- More expensive in the short term - May be contrary to strategic direction of the organisation (e.g. building internal capability)</li> </ul>
Build a DevOps platform	<ul style="list-style-type: none"> <li>- Price depreciates from first delivery</li> <li>- Infinitely scalable</li> <li>- Expertise of operations team grows as the platform develops.</li> </ul>	<ul style="list-style-type: none"> <li>- More expensive in the short term</li> <li>- Requires a specialised skillset</li> </ul>

We recommend completing a cost-benefit analysis of these options in consultation with industry experts. You should then document a strategic vision for DevOps based on the options above (or a hybrid e.g. Options 1 and 4).

### 6.3.2 Using DevOps to coordinate work across multiple sites

Regardless of the chosen approach, the DevOps function is simultaneously a development team, an operations team, a test team, a security team and a network team. This makes them a microcosm of the entire IT landscape. The cross-functional nature of DevOps means they are more reliant on tools to streamline and manage their different strands of work (architecture, delivery, development and test) than teams with just a single role. DevOps are also responsible for implementing the infrastructure and tool-set across the digital delivery organisation, working with local delivery teams to roll these out.

Collaboration tools help DevOps and service delivery teams work together effectively across multiple locations:

- Wikis allow teams to share architecture designs and straw-man approaches for comment and review; they can also be used to document models and experiments, and record and report on results  
Ticketing tools allow work to be specified, assigned, tracked and managed
- ChatOps tools allow for real-time collaboration and exposure of system events to co-located or distributed teams - ChatOps tools announce incidents, deployments, new code commits and configuration changes, allowing the teams to add comments and narrative to these events.

All of these collaboration tools should be integrated to enable cross-reference.

DevOps helps to coordinate work across multiple sites. Development and configuration in DevOps takes place in the source control code repository. Changes originate on the developer's local machine. They are committed to the source control repository, statically tested, built by a workflow and build system, and subjected to automated tests before eventual deployment. This applies to both DevOps and development teams, with the difference being that the DevOps output could be an orchestrated environment with a wide variety of applications, all integrated automatically.

Modern source control systems also enable a project to be "open-sourced" to a specific development community (see Section 5.7). This means that DevOps engineers can own a particular part of the platform and grant permissions to developers to contribute to it. The developers can choose which changes to merge into the trunk, which to ignore, and which require additional development or testing work. This enables development teams to drive the direction of a project, though strong leadership is needed to make sure development time is optimised.

DevOps should work with local delivery teams to roll-out monitoring and alerting systems with dashboards to present a real-time view of current service (see also Section 4.10). Combinations of system and business metrics help to communicate the performance of the system to key business stakeholders. Analysing these metrics with real-time system outputs in a time-series database allows developers, architects, QAs and DevOps engineers to better understand the performance of a live service, its patterns and behaviours, and helps to identify aberrant behaviour and risks.

Implementation of these tools and techniques by DevOps across multiple sites will increase streamlined development and effective live service operations. Strong leadership is also needed to develop and implement the DevOps strategy which underpins the DevOps activities.

## 6.4 Work Prioritisation

There are number of additional factors to consider when prioritising work across multiple DDCs. You may follow a consistent approach to project prioritisation, driven by a central governance group, irrespective of the existence of one or multiple DDCs, such as user/business value (see Sections 4.5 and 4.14). However, when deciding *where* to schedule the work there are other factors to consider, such as grouping of technologies, type of work, functions and capacity. Operations managers, working with the programme board, should be responsible for allocating projects to individual DDCs.

### 6.4.3 Pipeline management in a distributed model

You should centrally maintain an overall enterprise digital delivery product backlog or digital pipeline, following the approach outlined in Sections 4.5 and 4.14 (with overall oversight from a digital programme board). It may be that programmes of work (higher level user epics) could be handed to individual centres to own. These should be broken down and tracked as items within the Enterprise Digital Delivery product backlog.

Consider employing one or more lead product owners. A single lead product owner can have responsibility for the full portfolio of digital delivery. Alternatively, multiple lead product owners can exist, each responsible for managing a product backlog for a particular business area or location. This allows the organisation to adopt a scale-by-value delivery model to the product or service. A Lead product owner sets the overall vision for the value stream. Individual product owners take responsibility for separating out and refining the backlog items which their respective Scrum teams will deliver into the overall solution. At a macro level, the

lead product owner would prioritise the services to be delivered within his/her remit (for example, an activity that would previously have been conducted by a specific business group).

#### **6.4.1 The impact of multiple DDCs on the scheduling of work**

When prioritising and scheduling work, you may consider grouping projects by scope or technology. This can help centres build expertise and consistency of delivery, allowing them to specialise and become leaders in a specific business or functional area. This must be balanced, however, against creating 'functional silos' in different locations, deterring cross-centre collaboration. At times teams may need to work outside their area of expertise. The existence of common architecture, design and development standards (as established by the functional leads) should help this change in focus.

#### **6.4.2 Key questions to consider when allocating work**

The approach to allocating work between DDCs must not hinder the flexible and cross-functional culture of digital delivery. When scheduling work, you should seek to avoid creating functional silos and dependencies, while driving up economies of scale through increased specialisation and minimisation of rework. Each project should be considered individually.

There are two key questions to consider when allocating work across multiple DDCs.

##### **1) How are technology capabilities distributed?**

One option is to separate technical activities or tooling across the different centres. For example, Centre A may complete all analytics and social media analysis, whilst Centre B develops customer-facing web based applications. This can be useful where certain skills are rare in one location. Skilled people are critical to digital delivery, and may be difficult to recruit. Having a realistic staffing model is therefore important when scheduling all work. If local analytics resources are much easier to find in Centre A than Centre B, then it would make logical business sense to schedule the analytics work in Centre A. Building local expertise can drive best practices, but again this must not be done to create false dependencies and functional silos. This approach can be relevant to specific shared services such as DevOps, core platform development or transaction monitoring, with digital service delivery following a common technical approach across all sites.

##### **2) How is functional expertise distributed?**

We recommend that dependent service functionality is delivered in the same site. For example, where a single project involves development from multiple Scrum teams, it should be allocated to teams that work in the same DDC. This improves the ability to collaborate regularly, promotes consistency and builds common understanding of the product and user needs. The lessons learned should still feedback into the continuous improvement team to allow all benefits to be distributed across multiple DDCs. The benefits gained from functional grouping include higher consistency of solutions, development of subject matter expertise (leading to better user research and ultimately a better quality product), and facilitation of cross-team collaboration to solve complex problems.

We recommend considering technical and functional capabilities when scheduling work across multiple sites. However, it is also important to note that this may not always be possible. Organisational prioritisation and capacity should be taken into consideration. Having too much work in Centre A and not enough in Centre B is inefficient. At times, you may need to allocate work to teams outside their normal technical or functional area. The common platform model, and associated common standards, should help teams to move between different functional areas effectively.

## **6.5 Governance**

As an organisation grows its digital delivery capability across multiple centres, the stakeholder groups involved are likely to become increasingly complex. This section will describe the methods that you can use to make sure that governance requirements (including reporting, interactions and tooling) remain clear, precise and consistent. Given individual DDCs are part of a wider single business entity, all centres should

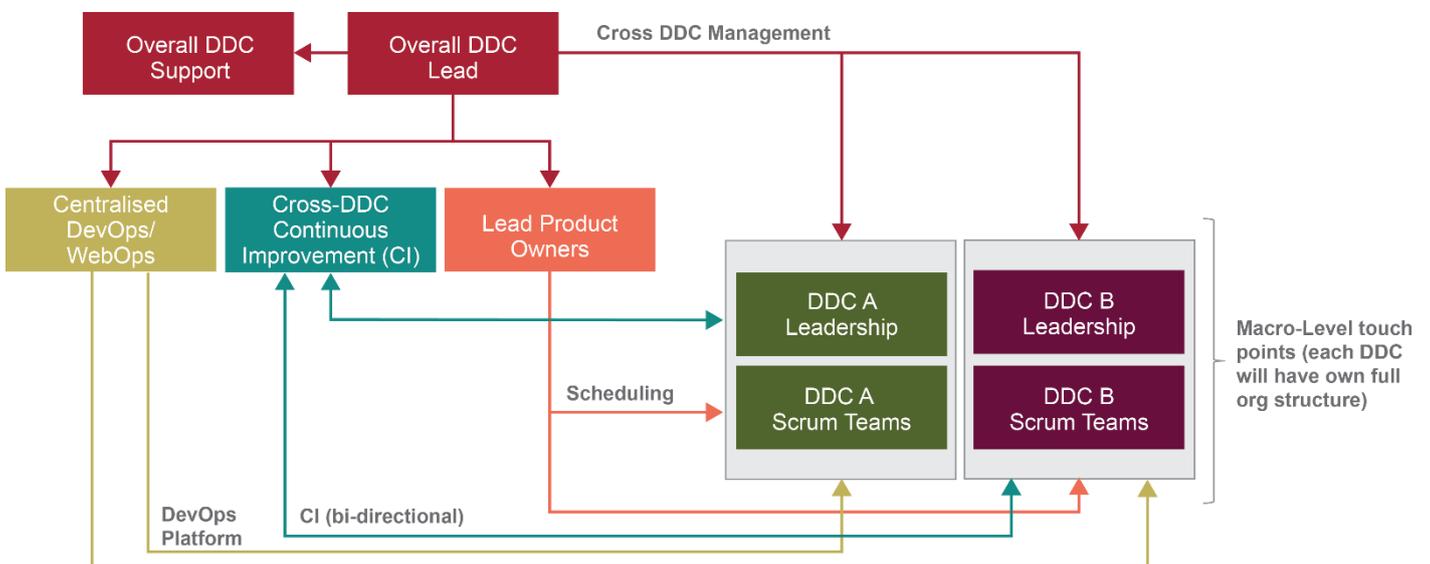
report up through one reporting line against the centralised enterprise digital delivery product backlog or digital pipeline.

### 6.5.1 Distributed DDC Organisation Chart

Section 4.5 gave a detailed view of effective governance for scaled digital delivery. This included governance arrangements within and external to the DDC. While most of these arrangements apply equally to delivery at one or multiple DDCs, we recommend the following additional functions when delivering across multiple sites:

- overall digital delivery lead (all DDCs)
- centralised DevOps
- cross-DDC continuous improvement function (including functional leads / authorities)
- lead product owner(s)
- overall DDC Support (including project management and commercials).

These functions interact with individual centres as follows:



**Figure 30: Indicative interaction between centralised functions and representative DDCs**

The representative organisation chart above highlights the summary interaction between the cross-DDC central functions and individual DDCs. This structure may vary on a case-by-case basis, but the key point is that central functions sit outside of the core DDCs (although the people who work in them may have responsibilities in both).

## 6.6 International expansion of Digital Delivery Centres

As distributed digital delivery capabilities mature, it raises the question whether UK public sector digital delivery can take place from overseas. The UK workforce is changing, with people increasingly likely to work from home and the number of temporary staff and contractors growing. Previous barriers to sharing data and IT services with third parties are shrinking. There is also a growing commitment to the development of APIs with third party software providers, some of which may be developed offshore. Traditionally, the public sector has been reluctant to offshore IT delivery, but as technology advances, attitudes change and the UK workforce becomes more fragmented, can public services be delivered from abroad?

Two obvious delivery models exist:

1. Where the Scrum team are distributed, or
2. Where full Scrum teams are located offshore.

In the first scenario, distributed agile delivery can be challenging, but has been proven to work at scale. Collaboration tools can be used to conduct virtual agile ceremonies and shared wikis and project management tools ensure that teams have the same view of progress irrespective of their physical location. Where the team are distributed it can be helpful to second or base experienced team members in the offshore location with less support.

In the second situation, a full team is located offshore and they collaborate with onshore product owners and/or operational management. Ensuring they have clear and regular communication with the product owner, if not co-located, is essential. We would recommend focusing on embedding the offshore team into the wider Digital Delivery Team. This includes participation in communities to build awareness of progress and challenges facing other teams. You should also consider how the offshore teams will conduct the necessary user research for their service. In both these situations, it would be helpful to bring the offshore team members into the DDC temporarily so they learn the desired ways of working and build relationships with experienced onshore team members before they return to their home country.

The growth in third party APIs, a commitment to 'coding in the open' and the Bring Your Own Device approach demonstrate that there is a desire for greater openness and collaboration. At a practical level, the technology exists. At HMRC, the Digital Delivery Teams have adopted a "work anywhere" culture, whereby team members use secure virtual private networks (VPNs) to work outside the internal network. This could be used by offsite and offshore teams to access the necessary environments, repositories, tooling and documents in different locations. Users who require access to sensitive data or the hosted platform would still need to connect to internal machines due to the security constraints of VPNs and personal devices.

## 7.0 CONCLUSION

Public expectations of government services are high. People interact with private sector companies via efficient, easy-to-use digital services, and they expect the same quality of service from public sector organisations. Simultaneously, the public sector is under pressure to cut costs and increase efficiency. Developing large-scale internal digital delivery capabilities is an effective way to fulfil these dual demands.

Building and operating DDCs is not easy. It requires change throughout the business and takes time to achieve, but the benefits are clear. This paper has discussed the key considerations needed to develop, operate and scale a digital delivery capability. Specific activities may vary between organisations, but this provides a clear framework to structure your digital delivery activities.

The experience at HMRC developing five DDCs comprised of more than 600 staff between 2013 and 2015 informs a lot of the content in this paper. But HMRC's digital delivery journey is not complete; further expansion is planned and new lessons are learned every day. We continue to adapt our approach accordingly.

Use this content to plan your DDC, then learn by doing.

*Accenture and HMRC co-authored this document with extensive input from Equal Experts and Capgemini, HMRC's other digital delivery partner organisations.*

## 8.0 GLOSSARY

Term	Description
Agile delivery	Iterative development and release of small chunks of functionality following development sprints that encompass all design, build and test activities
Alpha	Agile delivery phase. Alpha follows Discovery and precedes Beta. The objective of Alpha is to rapidly build and iterate solution prototypes based on user needs, and plan for Beta development.
APIs	Application Programme Interfaces (APIs) are a set of routines, protocols, and tools for building software applications. APIs allow applications to access the features or data of an operating system, application, or other service.
Beta	Agile delivery phase. Beta follows Alpha and precedes Live. Beta is typically the longest delivery phase and involves the design, build, test and release of a full end-to-end digital service to test with users, continuously iterating and improving on it.
Burn down charts	A tool to track the amount of work remaining in a sprint, release or project. The burn down helps to measure the progress of the team against timelines and is useful in helping teams visualise the end goal.
Burn up charts	A tool to track work completed in a sprint, release or project. The burn up helps teams understand their progress to an agreed scope level, and is particularly useful in visualising scope changes in agreed scope.
Business analyst	Scrum team member. Responsible for translating client requirements into a prioritised list of user stories in the product backlog, supporting the product owner with prioritisation.
Daily stand-ups	Daily team checkpoint during which each team member briefly explains 1) what they have done since the last stand-up, 2) what they are going to do before the next stand-up, and 3) what obstacles (if any) are in their way.
Developers	Scrum team member responsible for writing and testing software code in order to develop incremental features of a digital solution.
DevOps	DevOps (development and operations) is the alignment and integration of software developers and IT operations. DevOps teams design and implement the development approach, architecture, tooling and infrastructure needed to support streamlined delivery and operation of digital services.
Discovery	Agile delivery phase. Discovery follows Pre-discovery and precedes Alpha. The objective of Discovery is to start to research the needs of the service's users, understand how current services are performing, begin to define service requirements and plan for Alpha and Beta.
Epics	Large user stories that capture a significant amount of work and can be broken down into a series of smaller user stories.
Government Digital Service (GDS)	UK government agency leading the digital transformation of government. GDS work with other public sector organisations to make public sector departments and public services digital by default, simpler, clearer and faster to use.
Government Digital Service Digital by Default Service Standard	Service delivery guidelines that all public sector digital service deliveries should follow. Ensures teams build high quality digital services by imposing 18 criteria that a transactional service must meet in order to appear on GOV.UK.
Live	Final agile project delivery phase. Live follows Beta. Live services are operational and in use by the public. The service is improved continuously whilst Live, based on user feedback, analytics and further research.
Minimum viable product (MVP)	Delivery of the minimum functionality required for a product to start delivering some value to the end users/business, then iterating it further to add more features as needed.
Potentially shippable product	The team try to build 'potentially shippable products' in each sprint. This refers to outcomes (typically software) that are completed to a high degree of confidence

	and represent work of good quality that is potentially shippable to end customers at the end of a Sprint.
Pre-discovery	Initial agile project delivery phase. Pre-discovery precedes Discovery. The objective is to understand whether there is potential value in taking forward a new digital service idea by conducting a light touch evaluation and securing approval and funding to proceed.
Product backlog	The product backlog contains all requirements for the product in the form of user stories. It is managed throughout the delivery lifecycle by the product owner, with support from business analysts.
Product owner	Owns the service vision, budget and resources. Manages and prioritises the product backlog and has a clear understanding of how to translate user stories into product features.
QAs	Scrum team members who are responsible for ensuring that the service receives sufficient test coverage, access to test data and environments, and that requirements / user stories are testable. Assist developers in writing and executing test scripts in each sprint through test-driven and/or behaviour-driven development.
Release planning	Identifies the goal of a release, the overall features and functionality that a release should contain (once enough potentially shippable slices of the product have been developed within sprints), and any potential obstacles.
Scrum master	Ensures that the Scrum team are working in an agile manner and removes any impediments to progress.
Scrum of scrums	A technique to scale Scrum and manage dependencies across multiple teams to ensure that teams work together and prioritise activities jointly. It should be held weekly at a minimum, and more frequently as required, involving one representative of every impacted team.
Scrum team	The agile delivery team developing the digital service. Scrum is a type of agile methodology, though agile teams are often referred to as Scrum teams even if they are following a hybrid or different agile methodology (e.g. Kanban, XP). Also referred to as delivery team, agile team or development team.
Show and tell / Sprint review	Agile ceremony that takes place once at the end of each sprint. The team present the functionality that they have produced in the sprint to the product owner and other stakeholders, who are able to ask questions.
Sprint	Time-boxed cycle of development where the Scrum team works on iteratively developing potentially shippable slices of the overall product according to the top priority user stories in the product backlog.
Sprint planning	Plan “What” functionality is going to be developed during the sprint by considering the highest priority user stories in the product backlog, and “How” this will be done.
Sprint retrospective	Internal team meeting where the previous sprint is evaluated. Identify areas that went well and areas for improvement, turning these into actionable improvement measures to implement in the next sprint.
Test driven development	Development practice in which developers write an automated test case which defines new functionality. They then write the minimum amount of code to pass this test, and refactor the code to required standards.
User researcher	Scrum team member who leads user research activities within the team. Conducts interviews, tests prototypes and collects feedback from user groups. Works with business analysts to ensure feedback is integrated into user stories in the backlog.
User stories	User stories are system requirements structured from the perspective of end users. They outline what a user does or needs to do to use the service.
User centred design	Processes and practices whereby the needs of end users of a service are given extensive attention at each stage of the delivery lifecycle.
UX designer	Scrum team member who leads design activities within the team. Responsible for designing all elements of a digital solution that a user will experience (interaction points, visuals etc.)