CONTENTS

[WHATS INCLUDED 1](#_Toc423443000)

[OVERVIEW 2](#_Toc423443001)

[LOTTING STRUCTURE 2](#_Toc423443002)

[TIMESCALES 3](#_Toc423443003)

[KEY DELIVERY DATES 3](#_Toc423443004)

[CURRENT ROLES AND RESPONSIBILITIES OF THE CUSTOMER 4](#_Toc423443005)

[TEST & DEVELOPMENT REQUIREMENTS 4](#_Toc423443006)

[REQUIRED CAPABILITIES AND OUTCOMES OF THE SUPPLIER 4](#_Toc423443007)

[TERMS AND CONDITIONS 5](#_Toc423443008)

[EVALUATION STAGES, MINIMUM PASS MARKS & PRICE EVALUATION 5](#_Toc423443009)

# WHATS INCLUDED

Customer Requirements (this document)

Appendix A – Award Questionnaire (template to be completed)

Appendix B – Supplier Pricing Matrix (template to be completed)

Appendix C – Call-Off Contract (Part A&B) (Customer specific terms)

– Call-Off Contract (Part C) (Standard Terms and Conditions)

Any supplier invited to tender who has NOT returned their signed framework agreement for RM1043ii Digital Services 2 will NOT pass compliance check post-bid for this project and therefore their response will NOT be evaluated. Should any supplier have any questions regarding their status, please contact CCS via the eSourcing suite.

OVERVIEW

|  |  |
| --- | --- |
| CCS Project Lead: | Lucy McCormack |
| Customer: | Department of Health |
| Delivery Location: | London – no co-location but daily stand ups, regular catch-ups weekly, weekly reporting and sprint planning and sprint retrospectives |
| Phase(s): | Alpha, Beta and Live |
| Project: | DS02- 020 |
| Required Capabilities: | Include, but are not limited to: Software engineering and On-going Support  System Administrations and Web Operations |
| Subcontracting Permitted? | Yes  No |
| Supplier Partnering Permitted? | Yes  No |
| Contract Charging Mechanism (Alpha Phase): | Capped Time and Materials |
| Contract Charging Mechanism (Beta Phase): | Capped Time and Materials |
| Contract Charging Mechanism (Live Phase): | Capped Time and Materials |
| Tender Publish Date: | 12/08/2015 |
| Tender Submission Deadline: | 28/08/2015 |
| Proposed length of phase: | Alpha: 12/10/15 – 20/11/15; Beta: 23/11/15 – 06/01/2015; Live: 07/01/16 – 16/08/16 |
| Proposed Commencement Date of Project: | 12/10/2015 |
|  |  |

LOTTING STRUCTURE

## The Customer has structured this procurement as follows:

|  |  |
| --- | --- |
| **Lot 1** | Software Engineering and Ongoing Support and System Admin & Web Operations |

TIMESCALES

The Customer or CCS may change this timetable at any time. The Potential Provider will be informed by email if there are any changes to this timetable.

## It is the Potential Provider’s responsibility to monitor the online messaging facility (e-Sourcing).

|  |  |  |
| --- | --- | --- |
| **DATE** | **WHO** | **ACTIVITY** |
| 12/08/2015 | CCS | **Publish requirements to Potential Providers**  Clarification period starts |
| 17/08/2015 | CCS, Customer & Potential Providers | **Clarification Webinar 15:00**  Invite to webinar will be issued via the CCS eSourcing Suite. All questions and responses will be published via eSourcing Suite. |
| 24/08/2015 | Potential Providers | **Clarification Question period closes**  Please submit all clarification questions by 23:59hrs  Please note that we aim to publish all response to Q&A within 24hrs |
| 28/08/2015 | Potential Providers | **Submission Deadline**  Potential Provider must upload submission to the eSourcing suite by 12:00noon |
| 28 – 30/09/2015 | Potential Providers & Customer | **Demonstration, Testing and Scrutiny**  Supplier Presentations |
| 02/10/2015 |  | **Award Notification**  Publish Successful and un-successful Potential Providers. |
| 12/10/2015 |  | **Expected "Commencement Date" for Call-Off Contract/s** |

KEY DELIVERY DATES

|  |  |  |
| --- | --- | --- |
| PROJECT PHASES | START DATE | COMPLETION DATE |
| [Alpha](https://www.gov.uk/service-manual/phases/alpha.html) | 12/10/2015 | 20/11/2015 |
| [Beta](https://www.gov.uk/service-manual/phases/Beta.html) | 23/11/2015 | 06/01/2015 |
| Live | 07/01/2016 | 16/08/2016 |
|  |  |  |

# 

CURRENT SITUATION / BACKGROUND INFORMATION

**Current artifacts**

Product Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Area / Theme** | **Description** | **User Story** | **Priority  (High/Medium/Low)** |
| **Internal services** | Integration of internal services | As an end user I want to be able to access all of the tools I need to do my work in one place so that I can work more efficiently | **H** |
| **User centred design** | Review of design and UX | As an end user I want to have a high quality experience at a level I expect from external services so that I have confidence in the DH | **H** |
| **Editorial Process** | "Second eyes" approval process  Prompt for pages to be reviewed when a related page is updated | As an editor I want to be able to check and approve content that has been created/updated by other editors so that I can ensure the quality of content  As an editor I want to be confident that information related to the content I am editing is up to date so that there is only one version of the truth | **H** |
| **Online forms Security** | Increasing security level to OFFICIAL SENSITIVE | As a business owner (e.g. HR professional) I want to be able to collect information from staff in a simple and straightforward manner so that I can get the information I need to do my job | **H** |
| **Calendar / event booking** | Senior leader - book a 2 week Connecting placement at a front-line service (pick place & time) | As a senior leader I want to book my Connecting placement quickly and efficiently through the Intranet | **M** |
| **Content format** | Corporate calendar | As an event organiser I want to see what events the Department has scheduled so that I don't book my event for the same time  As an end user I want to see all the upcoming events run by the Department so I can find events I'm interested in and sign-up to attend | **M** |
| **In Your Building** | Option to add a date to a building news story | As an end user I want to be aware of building news on the day it's happening | **M** |
| Images | Wrap text around images | As an end user I want to read stories that are integrated seamlessly with images | **M** |
| **Emergency Banner** | Emergency banner - Triage process for what counts as an Emergency - Colour coding by severity - Link through to emergency post page with more info | As an end user I want to know about an urgent issue that affects me | **M** |
| **To Do List and In Your Building** | Ability to add pictures | As an end user I want to see rich content, with text enhanced by images | **L** |
| **Templates** | Add ability to feature YouTube video (and 'play button' oevrlay) in 'featured image' slot on news stories | As an end user I need to see that a story includes a video, so that I know the video will play when I open the news story | **L** |
|  | In WordPress dashboard, change "Checklist items" to "To do items" | As an editor I want to see menu items that correspond with items on the front-end | **L** |
| **Editorial Process** | Content templates | As an editor or content owner after go-live, I only need access to templates related to content input – for instance, ‘default page’ – but not ‘directorate’, so I can’t accidentally put content in the wrong format. | **L** |
| **Editorial Process** | Ability for a Lead Editor to delete editorial comments thread before a document is published | As an editor I need to be confident that internal discussions are not seen by contributers to the page | **L** |
| **Security** | Help DH simplify editorial roles: confirm what each active 'role capability' in the 'Roles' screen on WP entails | As an admin I need to be confident of which users have what levels of access | **L** |

**Project summary**

Since going into Live Beta in September 2014, the DH Intranet has changed the way its staff engage with information, each other, and the world beyond Whitehall (and Leeds). The new Intranet has delivered an improved service that is 4 times faster, 5 times smaller and 90% cheaper than its predecessor ‘Delphi’.

This has been a joint project between the Digital and Internal Comms teams and won the DH Recognition Award for ‘Working Differently and Better’ in January 2015.

We have a commitment to uphold the 26 points in the [Service Standard](https://www.gov.uk/service-manual/digital-by-default) and to continually iterate the service to meet evolving needs of users.

The development contract has now come to an end and this proposition seeks approval to go back to market for continuous development resource.

**Background**

In the 7 months since launch of the Live Beta on 10th September 2014 the DH Intranet has:

* Undergone 3 further Sprints to develop design, content and functionality in response to user feedback
* Achieved up-take of 2,196 users (out of 2,500 staff) who have created profiles and signed up for personalised email alerts
* Engaged 300+ staff in ongoing user testing, including all [Assisted Digital](https://www.gov.uk/service-manual/assisted-digital/index.html) users
* Been showcased as an exemplar digital project at the Ministry of Justice, Dept for Transport, Land Registry, Ministry of Defence, HMRC, Public Health England, NHS England, Construction Industry Training Board, Driver and Vehicle Standards Agency, Dept for Business Innovation and Skills, UK Trade and Investment
* Published the code openly for others to use on [GitHub](https://github.com/dhgovuk/intranet)
* 1,212,717 unique pageviews of the homepage, an average of 485 per user

Benefits realisation

* Achieved 61% user satisfaction rate, compared to 27% with Delphi
* Reduced the average time to find a page to 6 seconds from 20 seconds, saving £1.4m in staff time savings
* Saved £300k in annual development and running costs compared with Delphi

## Existing team roles

* Product Manager
* Delivery Manager
* Web Operations manager
* Business Analyst
* Content team (Internal Communications team)

## Current Technologies and Languages

WordPress platform PHP and MySQL

CURRENT ROLES AND RESPONSIBILITIES OF THE CUSTOMER

|  |  |
| --- | --- |
| **Role** | **Responsibilities** |
| Product Manager | Managing the product vision and product backlog |
| Delivery Manager | Managing the team’s resources, tracking and estimating projects |
| Web Operations | Hosting management, bug fixing, support and maintenance |
| Business Analyst | Analysing propositions, defining skill requirements and resource |
| Content team | Internal Communications team responsible for creating and publishing content |
|  |  |

CURRENT TECHNOLOGIES AND LANGUAGES

WordPress platform PHP and MySQL

REQUIRED OUTCOMES

There are a number of [ideal alpha](https://www.gov.uk/service-manual/phases/ideal-alphas.html) projects we wish to undertake at the beginning of a new continuous development contract. The user stories we want to explore (as noted in the [Product Backlog](https://docs.google.com/spreadsheets/d/1Lo7cadGZYXCMfZRF3kPwg5ghb6h2OpGsoh7Zy8PlUp8/edit#gid=1133512946)) are:

|  |  |  |
| --- | --- | --- |
| **User** | **User need** | **Expected research areas** |
| End user | As an end user I want to be able to access all of the tools I need to do my work in one place so that I can work more efficiently | Room bookings  People Finder (contact information) |
| Business owner | As a business owner (e.g. HR professional) I want to be able to collect information from staff in a simple and straightforward manner so that I can get the information I need to do my job | Online forms  Personal data security |
| End user | As an end user I want to have a high quality experience at a level I expect from external services so that I have confidence in the DH | Look and feel  Search |
| Editor | As an editor I want to be able to check and approve content that has been created/updated by other editors so that I can ensure the quality of content | Editorial workflow |

We propose running 6 further 2-week development Sprints over the course of 12 months, much of which will consist of the outcomes of the above research.

TEST & DEVELOPMENT REQUIREMENTS

We will require access to a test environment for User Acceptance Testing before any code releases can be made to the Live environment.

Inline with the Service Standard the code is published openly on [GitHub](https://github.com/dhgovuk/) and developers will be expected to release code updates after each iteration. We will grant access to the DH GitHub repository for this purpose.

REQUIRED CAPABILITIES AND OUTCOMES OF THE SUPPLIER

|  |  |  |
| --- | --- | --- |
| Required Capabilities and Outcomes of the Supplier | | |
| **Capabilities** | **Roles** | **Outcomes** |
| **Software Engineering and Ongoing Support** | **Developer (Word Press)** | Redevelopment of the existing platform according to prioritised user needs. Writing code, testing, adapting, maintaining and supporting to continually improve the service. |
| **System Administration and Web Operations** | **Security Specialist** | Analysis and implementation for storage of personal data. Help development team to build software that is secure and scalable. |

## 

THE METHODOLOGY

## [RELEASING SOFTWARE](https://www.gov.uk/service-manual/making-software/release-strategies.html)

<https://www.gov.uk/service-manual/making-software/release-strategies.html>

*How regular releases can reduce risk*

Releasing software comes with risks, so trying to minimise those risks is prudent. We do that in a number of ways:

* by releasing smaller chunks regularly it’s much easier to see what is going to change, and if something goes wrong it’s much simpler to roll that change back and undo it
* doing something regularly makes the case for investing in automation easier, removing much of the potential for human error and making each release the same
* if you’re doing something several times a day you tend to get better at it
* As well as reducing risk, being able to release early and often also helps products improve quickly, by reducing a potential barrier to quick experiments and rapid iteration.

It is important to think about how you release changes to a running application as early in the products development as possible. This is because it affects how software is developed and [tested](#id.ou6pjbx4lget) and how a product may be supported.

Being able to release software on demand is important. 6 monthly or longer release cycles are dangerous. Not only do new features rarely see the light of day but fixing known problems have to fit within a rigid release schedule.

Note that it’s important to make the distinction between releasing regularly and the ability to release all the time. The application should always be in a state where it could be released, that means quick changes can be made when needed. As an example changes to the software running GOV.UK are made on average 5 times per day.

In order to do that you have to consider:

* [your approach to testing](#id.ou6pjbx4lget)
* the quality of low level code – approaches like [continuous integration](#id.8a134m8w30a3), where code is tested constantly, and test driven design, can be helpful
* using the same tools and release processes for both the [development and production environments](#id.yr11ov9ge944) - this way the software and tools will be well understood and will have been run thousands of times before the first public launch

Although tools, potentially including commercial tools, are required to aid rapid releases the discussions should not start with what tools should be used or procured but with the needs of the service and the product team.

Finally consider the following two measures of a system; mean time between failures and mean time to recovery. A very traditional approach involves focusing completely on reducing the time between any failures happening, by hopefully improving the quality of the overall system. But problems will always happen at some point, so focusing some effort on reducing the time taken to fix problems that do occur can often be much more cost effective as well as improve the overall system uptime.

## 

## [TESTING IN AN AGILE ENVIRONMENT](https://www.gov.uk/service-manual/making-software/testing-in-agile.html)

<https://www.gov.uk/service-manual/making-software/testing-in-agile.html>

*What testing your service might look like*

It is important to recognize why we are testing in the first place, and that is to build the best quality system we can, that does what the customer requires, at a cost that everyone agrees we can afford (cost being money, business change, risk etc.). Too often, the focus of testing is to validate what has been produced and that alone, when in actuality it should be more about the following 7 concepts:

* Building quality in
* Everyone is responsible for quality
* Fast Feedback
* Tests are an asset of the product
* Faster delivery into production
* Clear and consistent view of testing
* Optimise value

### 

### Types of testing

The most noticeable difference with testing in an Agile world is that the majority of your test effort will be focussed on automated tests. These tests run in Continuous Integration (C.I.) which means that they form part of your code base and every time you make a change to your code, your tests are automatically run. This gives you immediate feedback on the quality of your code and helps prevent bugs being found at a later stage when they are expensive and complicated to resolve.

* Code Testing
* Exploratory Testing
* Load & Performance Testing
* Penetration Testing
* Accessibility Testing
* Crowd Sourced Testing
* Test Your Ideas

## [THE DEPLOYMENT PIPELINE](https://www.gov.uk/service-manual/agile/continuous-delivery.html)

## <https://www.gov.uk/service-manual/agile/continuous-delivery.html>

What happens to code between it being written by a developer, and deployed to production? We refer to this process as the deployment pipeline.

### The commit stage

Whenever a developer checks into [version control](#id.owokywyuffs2), a suite of tests is run against the latest version of the code. At this stage, any quick, easy-to-identify defects such as compile errors or unit test failures are caught. If the tests pass, the code progresses to the next stage.

### Shared sandbox environment

The code is deployed to a shared sandbox environment, where everyone involved in the project can observe it. The sandbox should be similar to production as far as is practical: for example, if production uses Postgres, the sandbox should also use Postgres and not another database such as MySQL or sqlite.

Every commit is considered a potential candidate to be released into production. The sandbox environment is the first environment where the application is deployed and run. This is the first stage where it can be visually inspected for quality by anybody on the team. The purpose is to identify any defect which means the application should not be deployed to production. If such a defect is found, this version of the code stops here; otherwise, it can proceed to further specialist testing environments.

### Specialist testing environments

There may be a need for other testing environments, to enable testing for specialist requirements such as load and performance testing, penetration testing, or accessibility testing. How many environments are needed will depend on the requirements and constraints of individual projects.

If code is determined to be of satisfactory quality, it can now proceed to the live production environment.

### Production environment

Once code has passed the commit stage, been deployed into the shared sandbox environment, had any necessary specialist testing run on it, it is considered suitable to go live. Deploying to production should be done in the same way as deploying to any other environment – using the same scripts, same [configuration management](#id.spj3di2271t0) tooling, and the same version of the code.

This ensures that when code is released to production, you are not doing it for the first time; you are instead performing an operation which has been validated at each stage throughout the deployment pipeline.

## 

### [Version control](https://www.gov.uk/service-manual/making-software/version-control.html)

<https://www.gov.uk/service-manual/making-software/version-control.html>

*Ensure the team can collaborate on code*

All software development projects must use a version control system. Version control allows you to track changes to code over time, meaning that you can quickly step back to an earlier version where necessary and you can annotate your changes with explanatory details to help future developers understand the process. Version control will also provide tools to audit who has made changes to the code and what has changed.

### Commits

Those updating the code should make small, discrete ‘commits’ of changes that are grouped according to their intention. They should be committed with a clear message explaining what the intention of the change was and (where appropriate) providing links to any supporting information such as development stories, bug reports, or third-party documentation.

## VERSION CONTROL SYSTEMS

At GDS we prefer to use a distributed version control system. This means that everyone involved in the process has a full copy of the code and of its history. This makes it easier for developers to create ‘branches’ in their code to explore new features or approaches without treading on the toes of those working on different aspects of the service. We use Git, which is one of the highest profile options.

It also provides extra resilience; if the network is unavailable the developers can continue to work and make small incremental commits, merging their changes back with everyone else’s at a later date.

### Not just code

It’s a good idea to also use version control for other aspects of your work, not just code. We use the same version control tools to manage the Service Design Manual as we do our code, and the Government Digital Strategy was also produced that way.

## 

## [CONFIGURATION MANAGEMENT](https://www.gov.uk/service-manual/making-software/configuration-management.html)

<https://www.gov.uk/service-manual/making-software/configuration-management.html>

*Manage a team's approach to configuration*

Your system is likely to be much larger than a single application, relying on other supporting infrastructure components. Even a simple application probably requires some configuration, to provide database credentials or a web service endpoint for instance.

In order to build robust, scalable and portable systems this configuration data should be well managed.

### Management tools

Configuration management tools help with documenting and maintaining the configuration and dependencies of a software system. Although this could be done using hand-made software, it’s common to use existing tools.

Three examples of existing open source configuration management tools are CFEngine, Chef and Puppet.

|  |  |
| --- | --- |
| Infrastructure as code | One approach to managing configuration is to describe the configuration and the software dependencies in code. This brings with it all the advantages of programming in general, including:   * testability * reusability * executable documentation * common and constrained language to describe a problem domain * Once described in code the infrastructure configuration is executed against the servers, networks and software in question. |
| Build for portability | Moving software systems between providers can be difficult and time-consuming. Even with compatible providers and simpler procurement rules it’s possible to lock yourself in through technical inertia alone.  Configuration management encourages a deep understanding of the configuration of the system and this can be used to move software easily between providers. |
| Use the same tools for development and production | A common problem in software systems is seen when code written by a development team works on their machine or a test environment but not on the production environment. A common cause of this is differences in configuration – different versions of software, different types of database or application server. This can be avoided by using the same tools for both development and production environments. |

## 

## [DEVELOPMENT ENVIRONMENTS](https://www.gov.uk/service-manual/making-software/development-environment.html)

<https://www.gov.uk/service-manual/making-software/development-environment.html>

*Early infrastructure needs for agile projects*

As software developers, the environments we use every day matter greatly. Below are a set of guidelines for development environments to enable the exemplar projects (service transformations committed to in the Government Digital Strategy) to:

* test software choices to prove they are valid
* experiment quickly with new approaches
* produce and test software in a production-like architecture
* develop rapidly and iteratively
* continuously test and monitor software during development

Although this document does not describe the capabilities and characteristics of a production environment, there is a general presumption that any production environment should enable the exemplar project development teams to:

* deploy updates to the system rapidly and iteratively (ie at least daily)
* continuously test and monitor software in production

### 

### Required

The essential capabilities of the development environment without which the development team will not be able to operate, are:

* **Current availability** - A service that is already operational and able to onboard customers very quickly (typically within 5 working days)
* **Internet connectivity** - Both incoming and outgoing internet connectivity. This should also facilitate remote management
* **Self service provisioning** - We should be able to remotely provision new machines ourselves to meet our needs as they arise, without the need to phone, fax or email anyone, and therefore require a self service method of provisioning virtual machines and storage
* **Suitable range of virtual machine options** - Support for 64 bit architectures and a range of virtual machine sizes at least up to 4 cores, 16GB RAM and 300GB disk
* **Run own operating system** - The flexibility to run whatever operating system is deemed suitable for the project, rather than just a limited subset of those supported by a vendor
* **EU-based data centres** - We would prefer to store data in the EU, and ideally within the UK, therefore we require development environments to be hosted only in EU-based data centres
* **Service Level Agreement** - A suitable SLA should be in place with the service provider (whether internal or external), with at least a 99.5% uptime guarantee
* **Development team access** - Approved development team members should have root access to manage virtual machines (eg to install & configure software)

### Desired

Optional capabilities which would make a marked difference to the production of the services, are:

* **Provisioning API** - The provisioning of virtual machines, storage, load balancing, etc to be available via an API. Any API should have a suitable authentication mechanism in place, and should be accessible to development team members via the Internet (optionally through a VPN)
* **Create virtual machine templates** - To speed up provisioning we would like to be able to store virtual machine templates from which new machines can be launched
* **Firewall and load balancer service** - If available a managed firewall and/or load balancer service may be used
* **Configurable private network** - We require the ability to manage internal networks, each consisting of specific groups of virtual machines. This should allow for some virtual machines not to be internet accessible
* **Virtual Private Network** - We may choose to expose parts of the service via a Virtual Private Network. The infrastructure service should at a minimum not prevent this and may ideally provide a suitable managed service

### 

### [Information security](https://www.gov.uk/service-manual/making-software/information-security.html)

<https://www.gov.uk/service-manual/making-software/information-security.html>

*Ensuring user data stays secure*

GOVERNANCE

Suppliers should ensure they follow the Service Manual guidance in regards to governance for service delivery

<https://www.gov.uk/service-manual/governance/introduction-to-governance-for-service-delivery.html>

**Digital Leader:** Adam Bye [Adam.Bye@dh.gsi.gov.uk](mailto:Adam.Bye@dh.gsi.gov.uk)

**Budget holders**:

* Adam Bye, Deputy Director Digital
* Mandy Dryden, Head of Internal Communications

**Delivery team**

Product Manager: Lisa Scott, DH

Delivery Manager: Sarah Wood, DH

Web Operations Manager: Francis Babayemi, DH

The Delivery team will report to Nayeema Chowdhury, Head of Digital Services, who will own the Product Roadmap for the Intranet and DH Internal Services. The Delivery team are empowered to make decisions about day-to-day service development, with ownership of the Product Backlog ultimately lying with the Product Manager.

**GOVERNANCE PRINCIPLES**

Both the supplier and the customer should follow the [6 principles for governing service development](https://www.gov.uk/service-manual/governance/governance-principles) to help create the right culture within the service development environment. They are:

1. [Don’t slow down delivery](https://www.gov.uk/service-manual/governance/governance-principles#dont-slow-down-delivery)
2. [Decisions when they’re needed, at the right level](https://www.gov.uk/service-manual/governance/governance-principles#decisions-when-theyre-needed-at-the-right-level)
3. [Do it with the right people](https://www.gov.uk/service-manual/governance/governance-principles#do-it-with-the-right-people)
4. [Go see for yourself](https://www.gov.uk/service-manual/governance/governance-principles#go-see-for-yourself)
5. [Only do it if it adds value](https://www.gov.uk/service-manual/governance/governance-principles#only-do-it-if-it-adds-value)
6. [Trust and verify](https://www.gov.uk/service-manual/governance/governance-principles#trust-and-verify)

**HOW TO GOVERN**

The service manual provides guidance on [how to govern across the life of a service](https://www.gov.uk/service-manual/governance/governance-across-the-life-of-a-service) with advice for each phase including:

* managing approvals and funding your digital service
* supporting delivery and managing assurance
* supporting your team
* how to scale up

**GOVERNANCE ACROSS THE LIFE OF A SERVICE**

<https://www.gov.uk/service-manual/governance/governance-across-the-life-of-a-service.html>

There are [5 phases of service delivery](https://www.gov.uk/service-manual/phases) — discovery, alpha, beta, live and retirement. Governance is important across all these phases and ensures the transition between them is seamless. People who govern must anticipate problems that could affect delivery and make sure it isn’t slowed down.

**MEETINGS**

<https://www.gov.uk/service-manual/agile/features-of-agile.html#standard-meetings>

Agile projects have 4 different types of regular meetings:

* [daily stand-ups](https://www.gov.uk/service-manual/agile/features-of-agile.html#daily-stand-up)
* [sprint planning](https://www.gov.uk/service-manual/agile/features-of-agile.html#sprint-planning)
* [sprint reviews](https://www.gov.uk/service-manual/agile/features-of-agile.html#sprint-review)
* [retrospectives](https://www.gov.uk/service-manual/agile/features-of-agile.html#retrospectives)

**ASSURANCE**

Assurance by Agile teams

<https://www.gov.uk/service-manual/governance/self-assurance-by-agile-teams.html>

Assurance is built into [Agile ways of working](https://www.gov.uk/service-manual/agile/index) with regular checkpoints and opportunities for feedback during service development. This means that assurance is proactive, ongoing and helps keep the service on track. The [governance principles](https://www.gov.uk/service-manual/governance/governance-principles) for digital services outline how to keep governance work on track in a similar way.

A [phase based approach](https://www.gov.uk/service-manual/governance/governance-across-the-life-of-a-service) to service development helps to increase the chances of success with in built assurance throughout each stage. Assurance should support delivery ([only do something if it adds value](https://www.gov.uk/service-manual/governance/governance-principles#only-do-it-if-it-adds-value)) and be proportionate to each phase.

**REPORTING**

<https://www.gov.uk/service-manual/governance/setting-up-the-right-reporting.html>

Teams should set up reporting in a way that won’t cause any extra work. This includes:

* visual management – teams should use their walls to capture everything they need for delivery
* face to face meetings, like the [stand-up](https://www.gov.uk/service-manual/agile/features-of-agile#daily-stand-up) and [show and tell](https://www.gov.uk/service-manual/governance/what-to-expect-from-the-show-and-tell)

This reporting should provide all the information that’s needed for good governance.

TERMS AND CONDITIONS

Please note that Customer specific Terms and Conditions apply to this agreement. Please refer to the Call-Off Contract Part A, for further information. Please note that these terms will supersede the standard terms within Call-Off Contract Part C Call-Off Terms and Conditions

EVALUATION STAGES, MINIMUM PASS MARKS & PRICE EVALUATION

## Evaluation will follow the approach below:

## Technical & Cultural evaluation

* Demonstration, Testing and Scrutiny

## Pricing evaluation

MINIMUM PASS MARKS:

## In order for Potential Providers to progress they must achieve or exceed the Minimum Pass Mark, as defined in the Award Questionnaire.

|  |  |
| --- | --- |
| Stage 1: Technical & Cultural evaluation | All Potential Providers who achieve the required Minimum Pass Mark for a Lot will be added to the Short List, and will be eligible to continue to in the Further Competition. |
| **Stage 2:** Practical Demonstration, and Scrutiny of the resources proposed by the supplier | Suppliers who meet the Minimum Pass Marks specified for Part A Supplier Confirmation, and Part B1 Written Submission; will be required to complete Part B2 Practical Demonstration of a particular skill (specified within the Award Questionnaire) in order to evidence capability.  Supplier resources will be required to respond to the Scrutiny questions stipulated within the Award Questionnaire. Each shortlisted Supplier must achieve the Minimum Pass Marks identified in the Award Questionnaire to continue to in the Further Competition. |
| Stage 3: Pricing evaluation | For each Further Competition the Customer has a choice as to how they wish the pricing to be evaluated. In this instance the Customer has specified Combined Evaluation as their chosen price evaluation method. For more information please see the Evaluation Guidance document held on the e-Sourcing suite. Please note that pricing will only be evaluated for those shortlisted suppliers that have met the Minimum Pass Marks for the preceding evaluation stages |