Great State

SERVICE DEFINITION

Service Design and Blueprints

GREAT STATE

At Great State we’ve been delivering change through digital for the public sector for over 20 years. Through our work with clients including the Royal Navy, Ministry of Defence (MOD), Her Majesty’s Prison and Probation Service (HMPPS), Office for Students (OfS), London Fire Brigade and the British Army, we have developed a deep understanding of the Government Digital Service (GDS) methodology.

We apply deep insight to help clients bring new products, services and experiences to market, at pace and with scale. We specialise in digital strategy, product innovation, experience design, and digital engineering.

We are passionate advocates of the Digital Service Standards, so much so that its agile approach has become central to our core principles and delivery framework. Our methodology for delivering and optimising digital products and services is based around agile fundamentals, and adapted based on your individual needs and requirements. Putting the user at the heart of everything we do shapes our work with all of the clients, not just those in the public sector.

Our engagements vary client-to-client, from end-to-end product development, technical builds that require complex integrations in-line with government security classifications, to providing specialist expertise just when you need it. We help in-house teams build better products, faster, and equip them with the knowledge to build better products in the future, upskilling their teams as part of the journey.

We favour a one-team philosophy to deliver technical innovation, compelling user experiences and digital transformation – our relationships with clients are open and transparent and allow for us to collaborate and work together as a joint team.

1. SERVICE OVERVIEW

Service design methodologies bring objectivity and structure when solving complex challenges. We champion a user-centred, research-led approach embedding service design principles throughout Discovery, Alpha, Beta and Live. We work collaboratively with our clients to iteratively redesign/create end-to-end, multi-channel services, presenting a complete picture of how user experience will be delivered.

1.1. SERVICE FEATURES

- User research-led problem solving, design thinking and capability mapping
- Quantitative and qualitative user research to understand users' needs
- Highly iterative, collaborative and user-centred approach throughout project lifecycle
- Strategic approach to multi-channel cloud service definition and design
- Creation of end-to-end service blueprints visualising current and future journeys
- User experience design using wireframes, prototype tools and service mapping
- Iterative design and development of complex services and products
- Tested prototypes/proof of concept and feasibility feeds into ongoing design
- Technology assessment and architecture considering emerging cloud technology
- Identify future MVPs and plan for implementation

1.2. SERVICE BENEFITS

- Balances user needs, business strategy and technology to inform design
- Design end-to-end across multiple touchpoints for a seamless service experience
- Risks managed by bringing technical decisions into early definition activity
- Aligns systems, tools, processes and technology around shared service vision
- Shared understanding of the current/future end-to-end service through service blueprints
- Built-in measurement of success to ensure value and constant iteration
- Collaborative approach shares knowledge and improves skills and capabilities

- Engagement across organisation – everyone working together to achieve vision
- Comprehensive approach ensures design consistency across projects and programmes
- Shareable and actionable digital blueprint outputs

2. DETAILED SERVICE DESCRIPTION

2.1. OVERVIEW

Service design methodologies bring objectivity and structure when solving complex challenges. We champion a user-centred, research-led approach embedding service design principles throughout Discovery, Alpha, Beta and Live. We work collaboratively with our clients to iteratively redesign/create end-to-end, multi-channel services, presenting a complete picture of how user experience will be delivered.

We tailor our approach for our clients and their specific needs. The scale and focus of an activity within a methodology is shaped according to key parameters including the type of the problem and the value its resolution will bring. Our methodology will always ensure that user needs are comprehensively understood and assumptions are tested early to ensure only the best solutions continue into design and development.

2.2. OUR PROCESS

Our process focuses on solving problems and testing hypotheses. Multidisciplinary teams work together to provide holistic and informed conclusions based on user research. We focus on user behaviours, needs and motivations through multiple user research techniques and analysis.

Our approach is flexible and adaptable so can be practised within the context of any project delivery methodology, although is well-suited to Agile methodologies in line with GDS. Our user-first method aligns with service standard principles.

We work in an iterative design sprint cycle to ensure we can test and learn as we go – we regularly engage with users to define and validate our solutions. We create excellent user experience based on clear, user-led evidence. We liaise with wider stakeholders and SMEs such as IT teams to ensure services align with any technical or architecture requirements. We deliver outputs such as strategic personas, user journeys, prototypes, experience maps, service blueprints and user stories.

2.3. EXPECTED OUTCOMES

Founded in user research, we ensure we design for what users actually need, not what people think they need. We deliver high quality products and services that are developed efficiently and quickly, utilising a method that aligns closely with service standards, and practices which support successful passing of service assessments.

A user-led approach brings value at all stages of agile delivery, and helps to engender a culture of objectivity and user-centred design within your organisation. Our collaborative and coaching approach also facilitates knowledge sharing and upskilling with internal teams/stakeholders, creating more sustainable, self-sufficient teams for the future.

3. OUR APPROACH

3.1. HOW WE WORK

All the work we do within the public sector aligns with the Digital Service Standard, ensuring that we:

- Work using agile methodologies including user-centricity and iterative development
- Understand user needs, based on thorough and ongoing research and testing
- Employ a multidisciplinary team, and we have a large pool of resource to draw from
- Work in the open, including all source code, where commercially viable
- Use open source solutions and common platforms where possible
- Ensure effective measurement and ongoing performance analysis
- Ensure AA/AAA accessibility standards are met
- Adhere to the Technology Code of Practice
- Work using phased delivery (e.g. Discovery, Alpha, Beta, Live)

We tailor our approach for our clients and specific requirements are always considered. We take a pragmatic and flexible view of working, ensuring close alignment with organisational structure, cultures, budget and stakeholder needs.

3.2. PROCESS AND ACTIVITIES

The following details our broad Agile delivery approach to working on a government or public sector brief. This approach has been developed over time and is fully in-line with GDS requirements. However, every project is different, and we work with organisations to refine an approach tailored to their needs.

3.3. PHASED DELIVERY

Following Agile principles and in-line with the GDS service standards, every project is approached using a phased delivery methodology, unless the specific project requirements dictate otherwise.

3.3.1. PRE-DISCOVERY

We often start with a Pre-discovery – this is a quick turnaround phase to identify the problem to be solved and areas for exploration through research, insight gathering and data analysis. Often, we need to engage wider stakeholders to help shape and focus the brief (for example, technical recommendations that impact a solution used across an organisation).

The typical activities include:

- Establishing business drivers
- Identifying SMEs
- High-level technical research
- Wide user research and data analysis into problem area(s)
- High-level user journeys and opportunity maps
- Prioritisation of requirements for Discovery

Possible outputs include a business case, concepts/visuals to inform Discovery, findings and recommendations report and approach for future phases. This phase isn't necessarily a requirement for every project and where this phase isn't undertaken separately, a number of the above activities would take place within Discovery.

3.3.2. DISCOVERY

A Discovery phase helps us to better understand the problem we are trying to solve. It allows us to understand user needs, and those of the organisation – agreeing business ambition and what success looks like. We interrogate the problem, gathering insights across audience, trends, brand and technology to inform the design process, shape the solution and understand impact. To do this effectively we conduct primary user research, but it is critical to contextualise this activity with insight from the organisation, the needs of internal stakeholders, technical constraints and the wider landscape. According to the specific challenge, this activity could include workshops, policy inductions, shadowing, analytics reviews, technical audits and competitor reviews.

In addition, we set the appropriate governance, communication frameworks and regular ceremonies, and agree ways of working. This can be flexed over time but is important to have a solid foundation in place from the outset.

A Discovery phase is generally planned for eight to ten weeks, depending on the scale of the project.

The first aspect of this phase would be a kick-off where we undertake a number of set tasks designed to ensure that we are fully aligned with the project aspirations, initial requirements are validated, and project risks are documented. This would normally include:

- Immersion and review of any work to date
- Pre-discovery requirements/outputs validation
- Agreement on ways of working, governance, communication, ceremonies, co-location, project tools
- Team chemistry session
- High-level risk management
- Stakeholder mapping
- Set-up of shared tools and systems
- Planning for the remainder of Discovery, including user-research requirement

Once the kick-off is complete, we move into the remainder of the Discovery activities, which would typically include:

- Stakeholder interviews to better understand longer-term goals and product vision
- User research: to establish user needs and to potentially challenge organisational assumptions
- Desk research and competitor analysis (if relevant)
- Technical audit and architecture review
- High-level requirements gathering and prioritisation (if not already been established)
- Research analysis to shape insights - identifying how business outcomes will be achieved by meeting user needs, and shaping scope for an Alpha

At the end of this phase, we will understand whether the project should proceed to Alpha, the areas that we would like to focus on, how we will measure success and the shape of the team required.

3.3.3. ALPHA

During the Alpha phase, we will work to understand how a new product or service needs to be designed and built, based on user needs and business requirements that have been captured in the Discovery phase. If Discovery establishes the problem to be solved, then the Alpha tells us how we should solve it.

The Alpha phase has prototyping at its core. In order to understand the most effective way to achieve the business outcomes identified in the Discovery phase, we test design hypotheses through user testing and technical proof of concepts.

Working in iterations, we can design prototypes, test, analyse and rework within a week. This is a collaborative process involving Service Designers, User Researchers, Designers, Content Designers, Copywriters and Developers, and our clients, to ensure that all design outputs are shaped by user needs, are technically feasible, and achieve the desired organisational outcomes.

The prototypes that we develop are of end-to-end elements of the user journey. They will be developed with the purpose of answering a specific research question or design hypothesis. The level of fidelity of the prototype will be proportional to the value of the question that we are answering. This means prototypes could be made from paper, lo fi design software or code.

An Alpha will typically last between 8-12 weeks, with a week-long 'inception' (planning) phase to start and ending with a week-long 'conclusion' (write-up).

In addition to the user-research led elements of the phase, we will also: Investigate any technical requirements and run proof-of - concepts as required

- Make final decisions as to CMS solution and cloud-based hosting, to enable Sprint-0 environmental set-up
- Validate any proposed IA, usually with by using Treejack testing with users
- Understand any copy and/or content requirements
- Plan for accessibility and assisted digital, ensuring that we have testing with users with a variety of access requirements
- Undertake a review of goals/measures of success to enable thorough analytics
- Establish data requirements
- Prioritise Beta requirements including non-functional requirements and constraints (this will provide the scope for the Beta written as user stories and acceptance criteria – in this phase this can be done to Epic level or more detailed user stories. If detailed user stories are not done in Alpha, they will be done as part of a Sprint O in Beta).

By the end of the Alpha phase we will know whether the product or service should proceed to Beta. We will also have determined the approach, addressing the structure, scope, technical and people requirements for Beta. Most importantly we will know how we are able to meet user needs with validated and tested end-to-end journeys.

3.3.4. BETA

Once we have completed a thorough Alpha phase, we move into Beta development. Usually working in Scrum, we ensure we are following the GDS requirements of iterative development, regular deployments and consistently following an Agile approach. We typically have a two to four week Sprint 0 to set up environments, finalise the backlog, run initial sprint planning and produce a level of UX/creative to feed into sprint development. We then move into two-week sprints, with a multidisciplinary team, with regular cycles of user testing, and releases.

As a standard, development will continue until there is confidence that the service can be deployed to Private or Public Beta, usually following a successful GDS assessment.

3.3.5. LIVE

Once a product or service has been set live, we work with clients to ensure that there is ongoing support in place, a plan for future optimisation and development, an ongoing testing schedule and effective measurement or agreed goals.

We learn from wide-ranging insights and develop and implement on-going optimisation plans/continuous improvement against a product roadmap. We support our clients in setting up their internal capability with training and team development and provide ongoing support.

Some of the activities we undertake once a product or service is live are:

- 360-degree data analysis, measure against KPIs and provide insight
- Develop product roadmap for continuous improvement
- Product backlog refinement and prioritisation
- Smooth transition to internal teams
- Collate/analyse user insight from Beta
- Ongoing user testing
- Multi-variant testing
- Managed services – SLA support
- Consultancy service

3.3.6. THE DELIVERY TEAM

3.3.6.1. JOB ROLES

In order to better deliver outcomes, it is imperative that we have the right team in place. Resource requirements change as the phases progress, but we endeavour to ensure that the same core team remains allocated to any project through its entire lifecycle. Our teams are truly multidisciplinary, and are able to work flexibly as required (e.g. UX and design, full-stack development).

Typical roles:

- Delivery Manager / Scrum Master
- User Researcher/s
- UX Consultant / Service Designer
- Technical Architect
- Business Analyst
- Designer
- Content Designer
- Developer/s
- Quality Assurance
- Digital Analyst
- Data Analyst
- Data Scientist
- Data Engineer
- Product Owner/Manager

In addition, to ensure we offer the required level of support from a client relationship management perspective, we will provide a dedicated client service resource.

3.3.6.2. ONE TEAM PRINCIPLE

We don’t like to work for our clients, but with them. Better outcomes are achieved if the project team is made up from a mix of client and agency resource. While this is not always possible, we encourage clients to provide roles beyond the standard Product Owner and/or Service Manager. Not only does this approach ensure better alignment of project goals throughout each phase but negates the need for extensive handover and/or upskilling at the end of the contract period.

3.3.6.3. CO-LOCATION

Following Agile principles, we will actively seek co-location opportunities on projects where possible. We believe that when the team work physically together then it leads to better communication, problem solving and ultimately, outcomes. When full-time co-location isn't viable, then open working practices and communication tools such as Slack, Teams, Miro, Trello and Azure DevOps are used depending on what is right for project and client.

3.3.7. PROJECT CEREMONIES AND COMMUNICATIONS

Successful stakeholder communications can be the differentiator when building a successful product or service. Large-scale central government projects may have large stakeholder groups and multiple third parties working towards a common goal. To ensure alignment and successful delivery, there are several tools and methods that we employ, regardless of the scale of the project.

These can include:

- Regular show and tells
- Blogging/week notes
- Open working
- Whole-team retros
- Scrum of scrums
- Open Slack channels (or equivalent)
- Azure DevOps
- Stakeholder working groups

We ensure the appropriate level of communication and governance is in place to foster collaboration and transparency.

3.3.8. GDS ASSESSMENT PROCESS

We understand the importance of successful GDS assessments, having taken several projects through the process, successfully passing. We work to the principle of 'always assessment ready' and undertake regular internal sessions and retros to ensure that we are working fully in-line with GDS standards throughout the lifespan of any project. We have worked with many government teams unfamiliar with the process and have successfully led them through assessments.

3.3.9. PROJECT HANDOVER

At the conclusion of any project, we will undertake several steps to ensure that the client is able to take on further project development, and business as usual (BAU) work. This is a standard step regardless of whether Great State is providing ongoing support. Working using a ‘one team’ approach should reduce the requirement for a large ‘end of project’ handover, but the following steps will be taken regardless:

- Full systems training as required (e.g. CMS)
- Stepped upskilling and training across all relevant roles
- End of project retrospective and/or project wrap up
- Comprehensive documentation
- Provision of all project documentation, including research findings
- Access to code repositories

4. TECHNICAL

4.1. TECHNICAL SELECTION PROCESS

4.1.1. PLATFORM SELECTION

Commencing in the kick-off phase, we will work closely with the client to understand the best solution for the product or service we are developing, ensuring this decision is based on several factors:

- User needs
- Budgets
- Longer-term requirements and technical roadmap
- Other systems and integrations
- The Technology Code of Practice
- What we have learned from any prior proof of concept work and previous experience

We work following the principle of **Last Responsible Moment** when it comes to making technical decisions and recommendations. This works hand in hand with agile, iterative development – it means that we avoid committing to a specific technology path too early in the lifecycle of the project. Making decisions too early can lead to rework, or even reimplementation. By delaying some of these decisions responsibly, it means that we will have more information available as the project goes on to make the right decisions, at the right time.

4.1.1.1. OPEN SOURCE

Open source software underpins a vast number of business-critical software and services around the world. It benefits from the scrutiny and rigour associated with being in the public domain. The large volume and scope of open source software also means that very often solutions to problems already exist and have been scrutinised and battle tested.

We carefully select open-source software based on a number of criteria, in line with the GDS guidelines. Some examples of the criteria we use are:

- Whether the software enables us to meet the needs of our end users
- Whether the software is under active development
- Whether any support available or offered
- The scalability and flexibility of the software

4.1.1.2. OPEN STANDARDS

We advocate the use of open standards wherever possible. Open standards have similar benefits to open-source software – they have been designed collaboratively by a wide range of stakeholders, and they enable interoperability between different systems and components.

They enable us to build robust products and services that can easily be extended and integrated with other services, in a well-understood way.

4.1.1.3. ENTERPRISE

We have extensive experience working with enterprise software covering a wide range of user needs. This includes content management systems, CRM systems, middleware and service buses, and other more use-case-specific software. This software is evaluated under similar criteria to open-source software, including:

- Whether it meets the needs of our end users
- Budget
- Whether the product is under active development and maintained
- Availability of support and SLAs
- Interoperability with other products and services

4.1.2. CLOUD HOSTING SELECTION

We have a cloud-first philosophy – the cloud brings with it many benefits, including:

- On-demand resources that can scale based on utilisation and need
- Resilience and redundancy
- Reduced operational overhead
- Cost effectiveness due to the economies of scale offered by cloud providers

We primarily build solutions on Azure and AWS, two of the largest cloud providers. We consider several factors when selecting a cloud hosting provider, including:

- Whether it meets the needs of our end users
- Budget
- Availability of on-demand services that mean we don't need to reinvent the wheel
- Security considerations around storage of data and personal information
- Uptime and availability considerations

We also have extensive experience with on-premises hosting and hosting services within the Government private cloud.

4.2. ITERATIVE DEVELOPMENT

Iterative development is a cornerstone of how we ensure we are delivering high-quality software that meets the needs of the end users.

We typically deliver work in sprints, following the Scrum methodology.

This means working in multi-disciplinary teams and delivering a working iteration at the end of each sprint. We hold a regular retrospective meeting at the end of each sprint to feed in what has been learned into subsequent sprints.

The multi-disciplinary nature of how we work means that we feed in learnings from user research, development, testing (user testing, performance testing, etc) such that we have a large, broad base of information to draw upon. It means decisions are informed, and we can react swiftly and pivot and adapt as needed. When combined with the last responsible moment principle, this means that we are very well-placed to deliver a working product or service that meets the requirements of the end users as those requirements evolve.

4.2.1. VERSION CONTROL

We use the industry-standard version control system, Git.

All code is stored in the version control system and we follow the *gitflow* workflow which is widely adopted in the industry.

The version control system allows us to iterate fearlessly, as we have the history of how the product has changed over time and can revert changes at any point if necessary. It also facilitates collaborative working.

4.2.2. CONTINUOUS INTEGRATION AND DEPLOYMENT

We use a continuous integration system, Azure DevOps, to regularly validate that our software builds correctly.

This enables us to know immediately if there is a problem with any recently committed code, and act immediately to resolve it. Our systems will not allow us to deploy code that has fundamental problems.

We also use the continuous integration to run automated tests – giving us another layer of confidence around code quality. If the tests fail, the code will not be deployed.

Code that has passed our automated tests is then deployed to a series of non-production environments. As we use the same tool to perform all our deployments, we know exactly what is being deployed and can track every line of code back to an item of work. The servers are always in a known state – because our system tracks what has been deployed and to where.

The system enables us to make small changes, often, which gives us a tight feedback loop – invaluable for being able to deliver working software quickly.

We advocate zero-downtime deployments – users should not need to suffer because things are happening behind the scenes. The typical pattern we employ is blue/green deployments – seamlessly swapping out application servers whilst the service continues to function as normal for end users.

5. QUALITY ASSURANCE AND TESTING

Quality assurance and testing is an intrinsic part of our process and the responsibility of the entire project team to ensure success. Our team of ISTQB-BCS certified (ISEB) test analysts take an iterative approach to testing.

Firstly, understanding the requirements (functional and non-functional) and acceptance criteria. This is captured in a test plan with the approach, roles and responsibilities, browser/mobile device specifications, entry/exit criteria, areas of risk and assumptions.

The key areas of testing carried out by our quality assurance team covers application code, compliance performance, security and privacy. We use automated and manual executed scripts and exploratory testing, and all items found will be raised in Azure DevOps. We cycle through this process throughout the entire development lifecycle until all changes have passed.

We implement performance tools which continually monitor the solution, providing us with a granular level of metrics and the ability to troubleshoot and identify technical bottlenecks, to maintain high levels of optimisation against agreed benchmarks.

5.1. MONITORING

Automated monitoring tools will be installed and configured to provide 24/7/365 monitoring. These include:

- **Server/host level monitoring** - monitors server resources such as CPU, RAM and disk usage.
- **Application/code level monitoring**- monitors code level performance of services, providing insight into areas for optimisation.
- **Website level monitoring** – monitors application uptime and performance of services.

5.2. BUSINESS CONTINUITY AND DISASTER RECOVERY

Great State ensures that with each solution a robust Disaster Recovery and Business Continuity Plan (BCP) for the delivered internal infrastructure and data is achieved. The overarching aim of the BCP is:

- Minimise interruptions to normal operations
- Limit the extent of corruption and damage (data and physical)

- Minimise the operational impact of the interruption
- Deliver BCP redundancy in the event of disruption to enable operations
- Provision of smooth and rapid restoration of the service

Our team of cloud hosting architects and support specialists works with our clients to define and implement a disaster recovery model to fit your requirements. Disaster Recovery is a coordinated process of restoring data, infrastructure, and systems required to support business operations.

There is no single solution that fits every client's disaster recovery requirements and our process ensures we consider the following:

- Risk tolerance
- Acceptable amount of down-time
- Acceptable amount of data loss
- Intangibles - impact to reputation, loss of customer trust

6. SUPPORT

After completion of work we will provide a warranty period whereby any issues raised are resolved quickly. Following this warranty period, we can provide a Service Level Agreement to ensure that any issues which arise can continue be dealt quickly and efficiently.

This comprises of two elements; proactive monitoring - of services to identify and isolate potential problems before they become issues and Reactive response - to resolve any unforeseen problems which may arise.

Support includes the following key elements:

- Event management access management
- Request fulfilment
- Applications management
- Problem management
- Support service desk for telephone and email support
- Website monitoring
- Incident management
- CMS platform support and maintenance
- Solution support and maintenance
- Monthly reporting

7. ACCREDITATION, SECURITY AND INFORMATION ASSURANCE

Great State fully conforms with the UK Cabinet Office policy for information security and assurance, therefore ensuring the process and mechanisms are in place and adhered to, in order to deliver secure and reliable ICT systems in accordance with UK government policy.

These include but are not limited to:

- JSP 440
- Security Policy Framework (SPF)
- ISO 27001 (2013)
- PCI DSS – Payment Card Industry Data Security Standard
- ISAE 3402 – International Auditing Standard
- UK Data Protection Act 1998 and EU Directive 95/46/EC
- Safe Harbour (Sharing data with US)
- Adherence to the new Government Security Classifications (GSC) – April 2014
- UK Cyber essentials and Cyber essentials+

Copies of Great State internal security policies are available on request.

Great State has in depth experience of working with DSAS (Defence Security and Assurance Services) and PSYA (Principal Security Advisor) to ensure full system compliance is achieved.

Deliverables include:

- Security and penetration testing (and supporting documentation)
- Impact assessment (Business, Privacy etc.)
- RMADS (Risk Management and Documentation Sets)
- Sy Ops (Security Operating Procedures)

We continually monitor and look at ways to improve against the standards set and maintain additional compliance with the International Standards ISO 9001 for Quality Management and ISO 14001 for Environmental Management and gold members of The Cyber Club.

7.1. ORDERING AND INVOICING PROCESS

Great State provides each client with a dedicated account handling team. This team will coordinate the scope and the cost of the project or implementation directly with the client. A full technical and functional specification and estimate will be drawn up for the project and Great State will require sign off of the scope of this work and a Purchase Order to proceed with work.

Invoices will be broken down into work stages and are typical invoiced monthly on the satisfactory completion of each phase. Great State has a standard 30 days' payment term for all invoices.

7.2. TERMINATION TERMS

Typically, Great State requires, as a minimum, a 3-month notice of termination all other terms and conditions for termination are as stated in the G-Cloud 13 Framework and Call-Off Contract.