Enhance our technical capability for in-house software delivery

With the formation of our digital delivery function, we must enhance our tools, standards and processes to ensure we can deliver excellent digital services to our users. The following principles will guide the development of our software delivery function:

- **Following GDS advice**, where it exists and is applicable, will assist in creating uniformity across government services, allow HMCTS to benefit from proven methods and expert advice and expedite the approval process where compliance is required. GDS peers will be involved in the governance process, and as required throughout the SDLC.
- **Build for the cloud**, to enable us to respond faster to the changing environment in which we operate and reduce costs. To maximise the cloud offering, Platform as a Service (PaaS) products will be preferred.
- **Approved technology stacks**, covering programming languages, runtime environments, frameworks, libraries and repositories, will be constantly reviewed to ensure HCMTS uses the most appropriate tech stacks for its needs:
    - A 'horses for courses' approach, allowing delivery teams to use the most suitable technology stack and propose new technology stacks when required
    - Focusing on a core selection of technology stacks to drive expertise, efficiency, uniformity and an increased time-to-market
- **Development/testing tools** will be provisioned to ensure the productivity of coders is maximised and to develop expertise in using tools such as Integrated Development Environment (IDE), Source Control, API Inspection, Packet Analyser and Package Manager. Reasonable effort will be made to accommodate coders that can increase their productivity using tools different to the ones provisioned. Sufficiently resourced local development environments (e.g. laptops) using modular components (e.g. containers) will allow staying mission-focused and maximise time invested.
- **You build it, you run it**. Teams will have the autonomy they require to own the end-to-end process, from building through to supporting their applications. CI/CD pipelines and Release strategies, such as Canary, Blue/Green and Versioned, combined with phased rollout approaches like Private Beta and Feature Flags, will allow delivery teams to practice Agile better. Infrastructure as code will allow teams to spin up environments as required and maintain them.
- **Just enough design** will be done throughout the SDLC in line with Agile and within the guardrails provided by the Technical Guidance Library and the technical governance. Software design will into account Domain Driven Design, Integration Patterns and Distributed Computing, Software Architecture Patterns, Design Principles, Design Patterns, Programming Paradigms and Software Quality Attributes. Fitness functions will measure implementation against their design artefacts and the wider 'guardrails', and automated to continuously assurance and discover the estate.
- **Secure design and coding** will assure Information Asset Owners, users and other stakeholders that their data is in safe hands. Designing with security in mind from the outset will reduce risks and allow operating in an Agile manner. Code made publicly available will follow a robust process covering code quality and security.
- **Emphasis on code quality** will be assisting to reduce the learning curve of new joiners and increase the versatility of our coders. To achieve code quality, a combination of automated tools, regular reviews and upskilling will be used.
- **Prototyping** will be used to adopt a more Agile environment, reduce risks and increase stakeholder engagement.
- **Shift left testing** will be practiced, involving testing from the initial stages of the development in order to ensure that development efforts are focused on the requirements and increase the quality of our products. Automated testing will be preferred, and where possible designed to run in 'headless' mode and within the release pipeline.
- **Rapid application development (RAD)**, will allow the delivery function to increase its capacity and address additional target audiences. RAD will be treated as a first-class citizen within the delivery unit and seen as software development for all intents and purposes. RAD will be delivered using a wide range of tools including Robotics Process Automation (RPA), Workflow, Business Process Modelling and Notation (BPMN), Extract Load Transform (ETL) and scripting. Digital Delivery will serve as a Centre of Excellence, guiding Citizen Developers across HMCTS using low-code technologies.