Software Engineering Principles

Performance Importance

Implemented on (date)	20/05/2024
Approved by (name & role)	Head of Software Practices (Fahad Anwar) In consultation with TAG (Technical Advisory Group).
Last review on (date)	16/05/2024
Reviewed by	TAG (Technical Advisory Group)
Next review due on (date)	16/05/2025
Principle owner (name)	Head of Software Practices In consultation with Technical Advisory Group (TAG) representing Software Engineers, Cloud Division, TISS and Security Division.
Principle owner (division)	Digital Services and Technology (DST)
Main point of contact (name)	Fahad Anwar Software Engineering Head of Practice
Status	Approved

Principle Review Record
This Review Record is to be completed on each time a review is conducted. Its purpose is to maintain a record of reviews, recording who conducted the review, the date of the review and the outcome of the review (fit for purpose, amendment required, principle no longer required, etc).

This principle is to be reviewed annually.

Review No	Review Conducted By	Review Date	Review Outcome
01			

Amendment Details

Date	Amendment Summary	Amended by	Version

RASCI (For detail please – RASCI Information document)

Responsible	G6 Program Managers through Technical Leads, G7 and SEO's
Accountable	Head of Software Practices
Supportive	Head of Cloud Functions (Amazon, GCP, Azure) SAIM SIRA Software Engineering Community of Practice (SE-CoP)
Consulted	Technical Advisory Group (TAG) representing Software Engineers, Cloud Division, TISS and Security Division.
Informed	Senior Leadership Team Software Engineering Community

	SAIM SIRA Design Authority Chair

Performance Importance

Our systems meet or exceed their users' expectations of performance. Degradations in performance are monitored, investigated, understood and either remedied, or accepted as appropriate in the business context.

Rationale

Systems that are fast and responsive for users enable them to carry out tasks more efficiently, and for the end user-facing areas of our business there is a strong correlation between slower systems and a negative impact on the user's response.
We must maintain a continued focus on the impact of changes to the performance of our applications or we risk impacting user experience, our ability to scale cost-effectively, and the stability of our systems.

Implications

- Set a performance criteria as early as possible and incorporate testing of this into the software delivery pipeline. Teams should have an awareness of how their systems have been engineered to meet performance criteria, and the means to avoid accidentally introducing changes that breach the expected user experience.
- Incorporate the analysis of performance tests into the release process. This is typically through frequent automated testing using short component performance tests which leverage mocks/stubs. Understand what tools are available to help with the data collection and analysis of these results.
- Choose the appropriate performance technique for your workload and platform. For example, client-side performance testing tools are more appropriate for applications with a frontend. Longer "soak" tests are only required for platforms that are long-lived, and "stress" tests are more appropriate when addressing specific stability concerns or risks.
- Frequent or continuous performance monitoring and testing should be implemented for the performance-sensitive components of the systems. Integration or end-to-end performance testing should be adopted cautiously due to its high cost and complexity of maintaining fully integrated and always available testing environments - consider whether upstream dependencies can be adequately mocked under a variety of load profiles to validate any performance risks.
- Consider using release strategies that allow performance of significant changes to be validated using a subset of end users and rolled forward or back quickly - such as canarying or dark launching with feature toggles.

Questions to be considered

- How critical is the performance tracking for your application (understand the impact to the end user/business)
- What is the dependence of performance of difference services/components (do you need end to end performance testing)?

Motivation for the principles

- https://www.gov.uk/guidance/the-technology-code-of-practice
- https://engineering-principles.jlp.engineering/
- https://github.com/otto-de/tech_manifest
- https://www.gov.uk/government/publications/security-policy-framework/hmg-security-policy-framework