Software Engineering Principles

Automate by Default

Implemented on (date)	
Approved by (name & role)	Head of Software Practices (Fahad Anwar) In consultation with TAG (Technical Advisory Group).
Last review on (date)	13/03/2024
Reviewed by	TAG (Technical Advisory Group)
Next review due on (date)	
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	Final Draft
Published version link	

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

Automate by Default

Tasks that *can* be automated *should* be automated by default. The choice *not* to automate should always be a conscious decision.

Rationale

Automation of well-defined tasks, such as deployment and testing, improves speed, consistency and reliability compared to manual execution. The automation code can also be valuable as a form of documentation for the correct process or desired outcome.

Implications

- Automation requires an upfront and ongoing investment that usually pays off quickly but must be balanced with delivering business value.
- When evaluating COTS (Commercial Off The Self) packages or SaaS (Software As A Service) solutions, the ease with which they can be automated should be a significant factor in their selection.
- Automation may benefit from use of specialised tools, frameworks, or languages, in which case teams will need to acquire or develop the appropriate knowledge & skills.
- Software and processes that are difficult to automate will need to be changed or replaced if they require too much effort or are not amenable to being automated.
- Automated tasks/processes that are complicated but infrequently executed can become a source of risk, to the extent that people decide to perform the task manually because they don’t have confidence in the automated scripts. Design for Testability must be applied so that the automation can be regularly proven to work, so that it can be relied upon when needed. Disaster Recovery processes are a good example of this.
- Automation scripts must be treated in the same way as other software and conform to these engineering principles.
- Zero touch for production enables least privilege and reduces the risk of misconfiguration.

Questions to be considered

- Is there already automation that can be reused?
- What tasks & processes can be automated in the CI/CD pipelines?

- Effort saved in running automation vs effort required to implement the automation?
- How automation serves other software development process?
- Am I doing this task regularly? For example, is there a bug or certificate rotation that happens regularly. These are candidates for auto healing and automation.
- Will the automation be used? Are you going to NEED it don't overengineer automation.
- Have you documented all decisions regard automation in ADR/other.
- Have we considered how to support/maintain the automation tools/work and what is maintenance cost.
- Have you looked in the Software Developers Platform (or any other central developer's repository) for the new automation scripts/tools you want to create (as there might be a solution already exist).
- If automation tools/scripts, you have created can be used by other teams, have you shared them either through SDP or using any other forum?

Motivation of the Principles

- https://www.gov.uk/guidance/the-technology-code-of-practice
- https://engineering-principles.jlp.engineering/
- https://www.gov.uk/government/publications/security-policy-framework/hmg-security-policy-framework
- https://google.github.io/building-secure-and-reliable-systems/raw/ch05.html#zero_touch
- https://google.github.io/building-secure-and-reliable-systems/raw/ch14.html#rely_on_automation

Principle Enforcement

G6 Program Managers through Technical Leads or Sr. member of respective software development team is responsible for monitoring and enforcing principle compliance.

Exception process

It is acknowledged that situations arise in which the Principal Detail as above may not be able to be met. Where this is the case, a documented Principal exception must be sought from the Principal Owner (specified in the table on the title page).