GitHub Usage Policy

ONS Source Code Management Policies

Policy controls	Details
Implementation date (when this first became ONS or UKSA policy)	20/05/2024
Approver name	Fahad Anwar
Approver role	Head of Software Engineering
Board Approval Name	Technical Advisory Group (TAG)
Last review date	05/03/2025
Next review due date	01/04/2026*
Policy owner name	Fahad Anwar
Policy owner division	Digital Services (DS)
Contact email (person or team email)	fahad.anwar@ons.gov.uk
Link to published version of policy (if applicable)	

*: Note that the policy is evolving, and regular reviews of specific sections are carried out in the regular (every other month) bases by the TAG as necessary.

Contents

Policy review record

Details of every review of the policy document

Review number	Reviewer name	Review date	Brief summary of review outcome
01	TAG	17/09/24	Amendment to the GitHub Usage Policy
02	TAG	05/03/25	Amendment to the GitHub Usage Policy

Version history

Details of every change made to the policy document.

SharePoint version no.	Amended by	Version date	Summary of what has changed
	Fahad Anwar	05/06/24	Corrected typo in clause 4.3
	Fahad Anwar	30/09/24	Updates to Section 2,5 & 6 (See amendments document for details)
	Dom Ford	31/03/25	Updates to section 2, 4,5 and 6. (See amendments document for details)

1. Policy Statement

1.1 This GitHub Usage Policy outlines the best practices and guidelines for using GitHub within ONS (referred to as “the Organisation”).

1.2 The use of GitHub for version control, collaboration, and project management (for public repositories) is encouraged and governed by the principles and rules set forth in this policy.

2. Scope

2.1 All employees of the UK Statistics Authority and Office for National Statistics.
2.2 Any external parties delivering services into the UK Statistics Authority and Office for National Statistics (unless by agreement)

3. Introduction

3.1 This policy aims to:
3.1.1 Maximise the benefits of GitHub by providing clear guidelines and best practices, the policy ensures that users can effectively leverage GitHub for version control, collaboration, and project management
3.1.2 Promote responsible use by encouraging users to adhere to ethical standards and best practices when using GitHub.
3.1.3 Ensure compliance with legal and regulatory requirements by mandating that users prioritise security and adhere to all relevant security policies, preventing the exposure of sensitive data and ensuring the protection of intellectual property rights
3.1.4 Establish clear guidelines by outlining responsibilities and restrictions to minimise potential risks associated with the use of GitHub, such as security vulnerabilities and inaccuracies

4. Background

The GitHub Usage Policy is designed to align with the strategic objectives of ONS by promoting the responsible and effective use of GitHub. This policy aims to maximise the benefits of GitHub while mitigating potential risks, ensuring compliance with relevant legal and regulatory requirements, and establishing clear guidelines for its use within ONS

4.1 **Alignment with Business Strategy**:
4.1.1 The policy seeks to leverage GitHub to enhance productivity, improve coding practices, and support the development of high-quality software within ONS.
4.1.2 By outlining responsibilities and restrictions, the policy aims to minimise potential risks associated with the use of GitHub, such as security vulnerabilities and inaccuracies.
4.1.3 The policy encourages users to adhere to ethical standards and best practices when using GitHub, ensuring that the generated code is reliable and secure.

4.1.4 By outlining responsibilities and restrictions, the policy aims to minimise potential risks associated with the use of AI-generated code, such as security vulnerabilities and inaccuracies.

4.2 **Legal Context**

4.2.1 The policy ensures that the use of GitHub complies with all relevant legal requirements, including data protection and intellectual property laws.

4.2.2 The policy mandates that users prioritise security and adhere to all relevant security policies, preventing the exposure of sensitive data and ensuring the protection of intellectual property rights.

4.3 **Regulatory Context**

4.3.1 The policy aligns with regulatory standards and guidelines applicable to the use of version control and collaboration tools within the public sector.

4.3.2 The policy establishes a governance framework for the use of GitHub, including roles and responsibilities, monitoring and enforcement mechanisms, and procedures for reporting and addressing policy breaches.

5. Policy Detail

5.1 Account and Profile

5.1.1 Account Creation:

5.1.1.1 GitHub users representing the Organisation can do so with either their personal GitHub account or one created specifically for Organisation work.

5.1.1.2 An Organisation email address (e.g. @ons.gov.uk / @ext.ons.gov.uk) is required to be associated with any GitHub account undertaking Organisation work.

5.1.1.3 Multiple email addresses can be associated to a single GitHub account, so in the case of personal accounts, the Organisation email address can be added as a secondary address alongside the primary personal email. This is required for accurate onboarding and offboarding.

5.1.2 Email privacy:

5.1.2.1 GitHub email privacy must be enabled to prevent Organisation email addresses from being divulged when performing Git operations. This document explains how to do this. Note: there are separate configuration locations for web-based operations and command line / Integrated Development Environment operations.

5.1.3 Profile Information:

5.1.3.1 GitHub users must be mindful of the information they share in public GitHub profiles. It is possible to have very little information populated in the

public profile and this is recommended for dedicated Organisation accounts.

5.1.4 **Publicly available information:**

5.1.4.1 GitHub users should familiarise themselves with the Social Media Policy. With the ability for much information to be in the public domain on GitHub, this policy is largely appropriate.

5.2 Managing movers/leavers/joiners

5.2.1 **New Joiner:**

5.2.1.1 For a new joiner to the GitHub ONSdigital organisation, the Technical Lead or nominated Sr. team members must raise a ServiceNow ticket.

5.2.1.2 This request is to send an invitation to join ONSdigital organisation. Users must create their own GitHub account before submitting this request to receive an invitation to join the ONSdigital GitHub organisation.

5.2.2 **Leaver:**

5.2.2.1 If an individual leaves ONS then the Technical Lead or nominated Sr. team members must also remove them from the ONSdigital organisation.

5.2.3 **Audit**:

5.2.3.1 GitHub audits joiners and leavers via the audit log.

5.3 Repository Management

5.3.1 **Repository Creation:**

5.3.1.1 By default, all new repositories must be created within the ONSdigital GitHub organisation or one of ONS GitHub recognised organisations.

5.3.2 **Repository name:**

5.3.2.1 Repository names should be clear, descriptive, and adhere to naming conventions of (use hyphens or underscores for readability. Avoid special characters, spaces, or uppercase letters in repository names). Repository names should be clear, descriptive, and adhere to naming conventions of (use hyphens or underscores for readability. Avoid special characters, spaces, or uppercase letters in repository names).

5.3.3 **Private/Internal Repository Information:**

5.4 Repositories should be created as public by default unless there is specific need for them to be internal or private. If there is a requirement that the repository should be Private OR Internal, then the repository must include a PIRR.md (Private/Internal Repository Reasoning Record) file.

5.4.1 **CODEOWNERS file**:

5.5 All repositories must include a CODEOWNERS file that contains the GitHub Team name(s) or the GitHub ID(s) of the team(s)/individual(s) responsible for that

repository. The CODEOWNERS file must be located in either the root of the repository, a ".github" directory, or a "/docs" directory.

5.5.1 **README File:**

5.5.1.1 All repositories should include a README file that provides an overview of the project, installation instructions, and usage guidelines. Markdown formatting is encouraged for clarity.

5.5.2 **Licensing**:

5.5.2.1 A license file (e.g., LICENSE) must be included in each public repository to specify the terms of use for the project. Choose a license that aligns with the project's goals.

5.5.3 **.gitignore:**

5.5.3.1 Utilise a .gitignore file to specify which files or directories should be ignored by Git to prevent unnecessary files from being committed to the repository.

5.5.4 **Documentation:**

5.5.4.1 Maintain comprehensive and up-to-date documentation for projects, including usage instructions, installation guides, and troubleshooting tips.

5.5.5 **Archiving Repositories**:

5.5.5.1 If an application is no longer used in production and there is no update on the repository for the last 01 year, the Technical Lead or nominated Sr. team member should archive the repository, Update the README file accordingly and resolve any open security alerts before arching the repository.

5.6 Collaboration and Communication

5.6.1 **Team Names**:

5.6.1.1 Use clear and descriptive names for teams and organisations to help members understand their purpose.

5.6.2 **Team Roles**:

5.6.2.1 Every team must have one or more members with the "maintainer" role assigned. These individuals are responsible for the repositories owned by the team and for maintaining the team membership in GitHub.

5.6.3 **Access Levels:**

5.6.3.1 Only teams should be granted access to repositories. This makes it clear why access is needed and helps to ensure people have correct permissions within team moves. If there is an explicit need for an individual to be granted access this should be agreed with Technical Lead or nominated senior team member.

5.6.3.2 There are four levels of access used for Team structure in ONS:

5.6.3.3 The default role that has the following relevant permissions:

5.6.3.3.1 Create Repositories, manage repository settings for their created repositories.

5.6.3.3.2 Create GitHub Teams and manage Teams that they have created.

5.6.3.3.3 See all organisation members and teams.

5.6.3.3.4 Hide comments on commits, pull requests and issues that they have write access to.

5.6.3.4 Software Engineers/Team Leads use the standard Member role, but with specific admin privileges granted on a repository-by-repository basis by existing repository admins or organisation owners. For a full list of permissions, see the GitHub documentation.

5.6.3.5 GitHub Organisational Owners have complete administrative access to the GitHub organisation; limited to few members only but not less than two people in the team. For a full list of owner permissions, see the GitHub documentation.

5.6.3.6 Security Management staff have Permissions to manage security alerts and settings across the organization, as well as read permissions for all repositories in the organisation. For a full list of Security Manager role permissions, see the GitHub documentation.

5.6.4 **Code of Conduct:**

5.6.4.1 ONS encourages respectful and inclusive interactions on GitHub. Adopt and adhere to a code of conduct to ensure a welcoming and harassment-free environment for contributors.

5.6.5 **Collaboration Workflow:**

5.6.5.1 Collaborate effectively by using pull requests for code reviews and discussions. Use issues and discussions to track and discuss project changes and enhancements.

5.6.6 **Unsolicited pull requests:**

5.6.6.1 If we get an unsolicited pull request from outside of the organisation it must not be merged. It should be raised to the Technical Lead or nominated senior team member.

5.6.7 **Branching Strategy:**

5.6.7.1 Implement a clear and effective branching strategy (e.g., feature branches, release branches) to organise and manage code changes.

5.6.8 **Branch Protection rules:**

5.6.8.1 Branch Protection rules must be used to enforce security policies by preventing accidental branch deletions, enforcing code reviews, and requiring successful automated checks before pull requests can be merged etc.

5.6.9 **Signed Commits:**

5.6.9.1 Use of signed commits is recommended, so other people can verify that your work comes from a trusted source. If a commit or tag has a GPG or S/MIME signature that is cryptographically verifiable, GitHub marks the commit or tag as verified.

5.7 Security and Compliance

5.7.1 **Privacy Settings:**

5.7.1.1 Review and adjust your privacy settings on GitHub to ensure that your contributions and personal data are appropriately protected.

5.7.1.2 **Security**:

5.7.1.3 Regularly review and update dependencies to address security vulnerabilities. Utilise automated security tools and services to identify and remediate vulnerabilities.

5.7.1.3.1 All public repositories must have secret scanning enabled.

5.7.1.3.2 All public repositories must have push protection enabled.

5.7.1.3.3 All public repositories must have dependabot enabled.

5.8It is strongly advised that all development team should have dependabot Auto-Pull request feature enabled on all their repositories.

5.8.1.1.1 Branch protection must be enabled on all repositories.

5.8.2 **Secret Scanning Alerts Service Level Objectives (SLO)**:

5.8.2.1 5 working days to resolve any secret scanning alerts from the time of creation.

5.8.3 **Dependabot alerts SLO**:

5.8.3.1 The following are SLO for dependabot alerts:

5.8.3.1.1 05 working days for CRITICAL category dependabot alert from the time of creation.

5.8.3.1.2 15 working days for HIGH category dependabot alert from the time of creation.

5.8.3.1.3 60 working days for MEDIUM/MODERATE category dependabot alert from the time of creation.

5.8.3.1.4 90 working days for LOW category dependabot alert from the time of creation.

5.8.3.2 These SLO will be effective from the date of this policy publication date.

5.8.4 **Implementation**:

5.8.4.1 The above security features might not be enabled on day one of publishing this policy, however the aim is to have all above security features enabled within 12 months of publication of this policy.

5.8.5 **Private / Internal repositories**:

5.8.5.1 Providing GitHub Advance Security features are available all above security features should be enabled on Private and Internal repositories as well.

5.8.6 **Preventative Controls:**

5.8.6.1 All public visibility repositories MUST have a preventative control in place to limit the risk of secrets from being accidentally leaked.

5.8.7 **Secrets:**

5.8.7.1 Secrets may be leaked from being present in non-public visibility repositories that are changed to public. As such, GitHub organisations MUST have at least one of the following:

5.8.7.1.1 Secret leak prevention coverage of non-public repositories.

5.8.7.1.2 Secret leak detection coverage of non-public repositories.

5.8.7.1.3 A formal process for the change of repository visible from non-public to public which includes a full inspection for the presence of secrets, prior to the change of visibility.

5.8.8 **Compliance:**

5.8.8.1 Ensure that all GitHub activities and repositories comply with applicable laws, regulations, and organisational policies. Be aware of any data protection or export control regulations.

5.9 Continuous Integration and Build

5.9.1 **CI/CD Pipelines**:

5.9.1.1 Where appropriate use GitHub Actions to implement continues integration and build pipelines to automate testing, building processes.

5.10 Reporting Issues

5.10.1 **Reporting Security Concerns**:

5.10.1.1 Report security vulnerabilities (excluding dependabot alerts) or suspicious activities immediately to the appropriate contact within the Organisation or follow the established incident reporting procedures.

5.11 Exception process

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

5.12 Monitoring and Enforcement

5.12.1 GitHub Service Owner must conduct annual audits of repositories to ensure compliance. Use automation to assist in auditing and reporting. Form an Audit Team from Software Community of Practice where members can be rotated.

5.12.2 Technical Leads or Sr. members of respective software development teams are responsible for monitoring and enforcing policy compliance.

5.13 Mandatory Training

5.13.1 **For all software engineers**:

5.13.1.1 Get started using GitHub to manage Git repositories and collaborate with others.

5.13.1.2 Creating and managing repositories

5.13.2 **For Technical Leads managing GitHub repositories/projects**:

5.13.2.1 Get started using GitHub to manage Git repositories and collaborate with others.

5.13.2.2 Creating and managing repositories

5.13.2.3 Build security into your GitHub workflow.

5.13.3 **Pluralsight Training Channels** (including mandatory training & more)

5.13.3.1 All users (Skill level – “Working”): Working with GitHub Source Code Systems.

5.13.3.2 Repository owners (Skill level – “Practitioner”): Working with GitHub Source Code Systems.

5.14 ONS Recognised GitHub Organisations

- https://github.com/ONSdigital
- https://github.com/best-practice-and-impact
- https://github.com/datasciencecampus
- https://github.com/IDPdigital
- https://github.com/integrateddataservice
- https://github.com/ONS-centre-for-crime-and-justice
- https://github.com/ONS-Health-Index-and-Projections
- https://github.com/ONS-SST
- https://github.com/ONSBigData
- https://github.com/ONSgeo
- https://github.com/ONSvisual
- https://github.com/open-sdg

5.14.1 If you believe there is a recognised GitHub organisation not listed above, please contact policy owner.

6. Breach of Policy

6.1 Violations of this GitHub Usage Policy may result in consequences, including warnings, access revocation, or other disciplinary actions, as determined by the Organisation.

6.2 Deliberate or repeated failure to comply with the requirements of this policy will be handled through the mechanisms outlined in the ONS Disciplinary Policy

7. Roles and Responsibilities

Individuals who have a role in this policy:

Role	**Responsible for**	**Accountable to**
Fahad Anwar (Head of Software Engineering)	Overall responsibility	Chris Penner (Deputy Director – Digital Services)

8. Supporting documents

Generative AI Guidelines	[Link]
GitHub Copilot Usage policy	[Link]
Source Code Management Policy	[Link]
Base GitHub Template Repository	[Link]