United Nations ECE/CES/2023/23

Economic and Social Council

Distr.: General
2 June 2023

English only

Economic Commission for Europe

Conference of European Statisticians
Seventy-first plenary session
Geneva, 22–23 June 2023
Item 3 of the provisional agenda
Moving towards open-source technologies – strategic and managerial perspective

Transforming statistical workflows to use open-source technology at the United Kingdom Office for National Statistics

Prepared by the United Kingdom

Summary

This document shows how the Office for National Statistics (ONS) and the wider analytical community in the United Kingdom government is transforming the way it does analysis by moving to large-scale adoption of open-source tools.

The use of open-source tools by ONS is part of a wider agenda around reproducible research. By adopting principles from reproducible research and software engineering good practice, ONS has improved the quality, transparency, and resilience of government analysis. However, this process of transformation comes with significant challenges. The paper explores these issues, and how they are being tackled.

The document is presented to the Conference of European Statisticians’ session on “Moving towards open-source technologies – strategic and managerial perspective” for discussion.

I. Benefits of open-source technology

1. The government of the United Kingdom made a commitment to make its code open source by default at the Open Government Partnership summit in Paris in 2016. The government Central Digital and Data Office[1,2] advocates this policy to improve transparency, flexibility, and accountability. The only clear exceptions are around storage of keys and credentials, algorithms where openness could compromise essential security, for example in fraud detection or disclosure control, and code that could reveal the details of unreleased policy before it is announced.

2. Open-source technologies have the potential to deliver three significant benefits for national statistical offices. The first is improved transparency. Tools are freely available, and open sourcing our code when it is safe and appropriate to do so means statistical processing is fully transparent, verifiable, and reproducible. Further, writing our code using open-source tools means that anybody can run, re-use, and challenge our processes. This enhances our capacity for robust audit and scrutiny.

3. The second is improved efficiency. Open code is easier to share, reducing duplication of effort and maximizing the potential for re-use and collaboration between teams internally, across wider government and with external partners.

4. However, we argue that open-source languages like R and Python are only truly transformative when combined with a wider set of tools and practices to ensure reproducibility. Realizing the maximum value of open-source technologies for analysis in government – the third benefit – is fundamentally linked to also adopting practices from software engineering and reproducible research[3]. This is what sets open-source tools apart from traditional analysis software. They support and enable software engineering good practices like automated testing, continuous integration, and version control. This makes it easier to build workflows that are fully tested and audited, but also changes the role of the analyst from operator to designer.

5. To realize this third benefit fully requires changes to the way we think about doing analysis. In practice, it means treating analysis workflows as software products rather than collections of manual steps. It means creating and running those workflows using open-source tools to ensure transparency and adopting software engineering good practices to make sure that workflows are tested, documented and transparent. Finally, it is likely to involve multidisciplinary teams that bring together analytical and software development competencies.

II. Analysis as code: ensuring reproducibility with software engineering good practices

6. Since 2017, analysts in the United Kingdom public sector have made their analysis more efficient and reproducible using a methodology called Reproducible Analytical Pipelines (RAP)[4,5]. They advocate writing analysis as code and using open-source tools, version control software, and dependency management.

7. Government analysts have found that they can use the same digital capability they have gained to improve reproducibility to deliver better analysis. They improve quality through automated data validation and testing. They make their analysis more impactful through interactive data presentation, more prompt and less costly by removing manual steps, and more powerful through advanced analytics.

[1] Central Digital and Data Office, 2017, Be open and use open source.

[2] Central Digital and Data Office, 2017, Open source guidance

[3] The Turing Way Community, The Turing Way: A handbook for reproducible data science, https://doi.org/10.5281/zenodo.3233853

[4] https://dataingovernment.blog.gov.uk/2017/03/27/reproducible-analytical-pipeline/

[5] https://analysisfunction.civilservice.gov.uk/support/reproducible-analytical-pipelines/

8. Delivering reproducible analysis through open-source technology is a strategic priority for government analysts. In 2022, the Government Analysis Function[6], the network for all civil servants working in government analysis, published its strategy for Reproducible Analytical Pipelines[7]. The strategy sets out to embed reproducible analysis using open-source technology as the default approach to analysis in government, learning lessons from the effective deployment of open-source tools to support rapid analytical decision-making in the COVID-19 pandemic.

9. By building analysis as software, we can draw on software engineering best practice to provide immediate benefits:

- Higher quality analysis and reduced risk, with quality assurance built into all parts of the process
- More efficient and reliable processes
- Improved transparency and greater confidence in the analysis from producers, managers, and users
- Improved business continuity and knowledge management
- Analysis that is easier to adapt and reuse.

10. In the United Kingdom, we have defined a “RAP minimum viable product”[8] which sets out seven critical properties to meet before analysis qualifies as reproducible:

(a) The analysis must eliminate or minimize manual steps like copy and paste, point and click or drag and drop operations. All manual steps must be fully documented.

(b) Analysis should be built using open-source software which is available to anyone, preferably Python or R.

(c) Analysts must deepen technical and quality assurance processes with peer review to ensure that processes are reproducible.

(d) Analysis must guarantee an audit trail using version control software, preferably Git[9].

(e) We should open source our code by default unless it is unsafe or inappropriate to do so, using file and code sharing platforms.

(f) Analysts must comply with the United Kingdom government and departmental good practice for quality assurance.

(g) Analysis software must contain well-commented code and have documentation embedded and version controlled within the product, rather than saved elsewhere.

A. Analysis as code: the challenges

11. Realizing these benefits requires significant changes to the way we do analysis. In moving to wide-ranging adoption of open-source technology we have encountered three important challenges. We need to make sure that analytical teams have the right skills. We need to build the right culture to support these new ways of working. Finally, we need to make sure that analysts have access to the tools and platforms they need to enable full use of open-source technologies and the software engineering practices that make them transformative.

[6] https://analysisfunction.civilservice.gov.uk/
[7] https://analysisfunction.civilservice.gov.uk/policy-store/reproducible-analytical-pipelines-strategy/
[8] https://github.com/best-practice-and-impact/rap_mvp_maturity_guidance/blob/master/Reproducible-Analytical-Pipelines-MVP.md
[9] https://git-scm.com/

IV. Supporting the transformation to analysis-as-code

12. In this section, we discuss the three challenges in more detail, how we are working to overcome them, and what we have learned as we transform.

A. Developing the right skills

13. The skills needed to enable effective use of open-source technology cut across three areas of capability. Analysts themselves need the right skills to build and assure analytical workflows in code. Managers and leaders of analysis must be confident about managing the delivery and assurance of that analytical software. Finally, organizations need to understand the skills that they will require to work in these new ways so that they can recruit the right people.

14. The United Kingdom government analysts still perform analysis with closed source tools like Excel, SAS, or STATA. While most analysts are comfortable with writing procedural scripts, many are less familiar with good practices in programming such as modular code and the use of functions and classes. There are still important knowledge gaps when it comes to adopting software engineering good practices like version control, unit testing, managing dependencies or continuous integration technologies.

15. We are tackling the analytical skills gap at ONS in four ways:

(a) We have developed a modular learning pathway so analysts can build the skills they need to work in these new ways. It introduces RAP concepts, principles and practices, coding with R and Python and software engineering approaches like clean code, command line basics, Git version control, modular programming and writing functions, unit testing, packaging and documentation and continuous integration;

(b) We are building these new skills into our professional analytical competency frameworks so that there is a clear expectation that they are critical;

(c) We have a central team who support RAP transformation. They provide guidance, standards, tools, and policies and offer consultancy support to help analysts to transform the way they work;

(d) We have set up communities of practice for ONS and the wider Analysis Function to provide peer-to-peer support and review, facilitated by our central team. These offer regular meetups as well as online discussion forums.

16. The learning pathway provides a solid foundation, but we have found that developing the full capability to build and support reproducible pipelines requires hands-on practical experience gained through mentoring and peer support. Analyst teams typically require mentoring support for 2 to 3 months while they work through an initial project to build their skills.

17. The ONS RAP central team works with statistical analysis teams in the business to enable them to become self-supporting. This is a “hub and spoke” model of deployment, with the central team supporting and mentoring other analysts rather than building and maintaining the analysis for them.

B. Building the right culture

18. When implemented fully, the “analysis as code” approach of RAP significantly reduces technical debt by building in practices that maximize transparency and resilience. However, successful deployment requires the right organizational culture.

1. The building blocks for sustainable, open-source analysis

19. We have identified six components that must be in place for RAP transformation to succeed. We use them as the basis of a memorandum of understanding to set out expectations for business areas that need support to transform their work to use open-source tools:

(a) Is there commitment and trust from senior managers? RAP projects succeed when senior managers are committed to transformation and ensure that time and resource is available to do the work;

(b) Is there commitment from team members? RAP projects succeed when the team support the project and are committed to delivering the outcome;

(c) Is there enough time for team members to contribute? RAP transformation projects take up to three months, and team members typically need to spend at least 20 per cent of their time every week on transformation work to embed knowledge transfer;

(d) Is there a process to transition to Business as Usual (BAU)? RAP projects need to work towards a sustainable business outcome. Having a clear plan from the business area for transition to BAU and long-term maintenance is essential for making the projects sustainable and maximises the chance of delivering early benefits;

(e) Is there a base level of technical understanding? RAP transformation involves mentoring and just-in-time learning, but teams need a base level of coding knowledge before they begin. Experience has shown that where this is missing, teams struggle to cope with the amount they need to learn. ONS teams are required to complete the learning pathway before embarking on a RAP project;

(f) Do the team have the right tools in the right place? RAP transformation requires that team members have the tools they need to deliver the workflow and the right technology platforms for the project.

2. Building a sustainable approach to Reproducible Analytical Pipelines

20. Senior leaders and managers of analysis have a critical role in the successful deployment of open-source technologies. They work as champions by promoting a “RAP by default” approach and making it clear that they value and support transformation. They must advocate the value of coding skills in analytical practice. Finally, they have a key role in making sure that teams have the time and resource to build and sustain the transformation.

21. RAP workflows need to be resourced and maintained like any other analysis. ONS teams who have adopted RAP practices have found that while transformed, automated workflows take significantly less resource to maintain than manual ones, planning in enough resource for ongoing maintenance is critical and not always anticipated at first by managers.

22. Workflows need to be resilient to staff turnover and single points of failure. This is a significant ongoing challenge while we are still transforming and building up capability because of the risk of creating technical debt if analytical teams lose key individuals with scarce skills before transformation is complete. Transformation is most resilient when we build RAP capability across an entire team, so that if one person leaves expertise does not leave with them so that projects become unsustainable.

23. We operate different RAP deployment models across ONS and have yet to settle on a preferred approach. Some business areas are growing capability across their analytical teams, with RAP skills embedded in lots of different areas. Others have set up specialist groups to provide support for automation and RAP across an entire business group and build and maintain those products in collaboration with analytical teams.

24. Both models have pros and cons. The first tends to take longer to scale but is likely to result in a larger pool of capability if we can manage risks around staff turnover, so that many analysts can build, maintain, and update workflows. The second depends on the capacity of the specialist team to build and maintain the pipelines and their analytical colleagues to specify and assure the outputs. It is more vulnerable to capacity limits, as the ability to build and maintain processes relies on a small pool of experts.

C. Access to the right tools and technology

25. Supporting open-source technologies for statistical analysis requires the right IT platforms and infrastructure. Most statistical outputs require some form of data engineering, analysis, or research. The Office for National Statistics is working towards updates of its cloud-based IT platforms that will fully support reproducible analysis using open-source technologies.

26. This is an ongoing activity. Current research and production platforms at ONS were built before RAP requirements were fully understood and do not support all of them. We are focusing our efforts on making sure that the next generation of computing platforms enable analysts to implement RAP workflows.

27. We have identified a set of high priority requirements to enable full support of RAP on our future platforms as we build towards them. These are set out in Appendix A.

V. Assuring the quality of open-source workflows: guidance, standards, tools, and peer review

28. Teams require support as they transform their analysis to open-source, reproducible workflows. Training and consultancy support were described above. Here, we focus on other resources.

A. Guidance, standards and tools

29. We have developed a set of government-wide standards, guidance and tools to support analysts in implementing RAP. We disseminate them through a public facing GitHub site[10] where we manage them as products, so that analysts from across government can request new features or collaborate with us in improving them. Key resources include:

(a) Quality Assurance of Code for Analysis and Research[11]. This manual sets out all the practices which analysts need to undertake to build reproducible workflows. It also provides guidance for managers about how to quality assure such pipelines;

(b) Govcookiecutter – a software project that provides standard templates for open source code projects;

(c) The RAP minimum viable product;

(d) Guidance on open sourcing analytical code[12]. The guidance is designed to help analysts evaluate how they could benefit from open sourcing their code and how to do so safely.

B. The role of peer review

30. Peer review is a standard component of software development good practice, and we advocate its use in all open-source projects. We also recommend pair programming, where two analysts collaborate to build a workflow, as a way of engaging in peer review from the very start. We have developed standard templates for code review in ONS, and version control software like Git makes peer review easy to manage within teams.

31. A wider challenge comes in setting up and maintaining a community of practice to facilitate peer review across teams or between analytical areas in different government departments. While communities of practice now exist within ONS and through our cross-

[10] https://github.com/best-practice-and-impact

[11] https://best-practice-and-impact.github.io/qa-of-code-guidance/intro.html

[12] https://analysisfunction.civilservice.gov.uk/policy-store/open-sourcing-analytical-code/

government RAP Champions Network[13], there is more work to do to set up routine peer review processes.

C. Maturity models for open-source analysis

32. While the RAP Minimum Viable Product defines a baseline for open-source analysis, it represents a low bar for what we would expect to see when our capability is mature. Developing maturity models for both open-source analysis and the platforms that support them that we can use to benchmark and monitor progress will be part of the next iteration of our guidance.

[13] https://analysisfunction.civilservice.gov.uk/support/reproducible-analytical-pipelines/reproducible-analytical-pipeline-rap-champions/

Annex

Platform requirements to support Reproducible Analytical Pipelines

We set out high priority technical requirements to enable analysts to build reproducible analytical pipelines on an analysis platform below.

- Dedicated support for the open-source coding languages prioritized by the organization. For ONS, this is Python and R.
- Access to all supported versions of Python and R, and the flexibility to create virtual environments containing the required software versions and re-use them in successive sessions.
- Version control of coding projects using Git software, with controlled access to GitHub, Gitlab or similar for version control, continuous integration and making code publicly available.
- Access to all the standard package repositories for Python and R (for example CRAN, PyPI, conda), usually through a local, secure mirror site such as Artifactory.
- Access to standard integrated development environments (IDEs) such as RStudio or Visual Studio Code.
- Access to public and internal hosting for documentation.
- Distributed computing resources and tools for using them such as Apache Spark.
- Orchestration tools to enable end-to-end automatic execution of pipelines, such as Apache Airflow.
- Distinct workspaces for development, research, and production that analysts can create by self-service.
- Secure data storage including relational database support and big data storage, with standard ingest and egress routes.