Key success factors and enablers for developing a successful open source community

**Keywords:** Open Source, DevOps, Statistical Software, Collaboration, Community

# Introduction

## Who is the SIS-CC?

The SIS-CC vision [1] is *a reference open source community for official statistics, focusing on product excellence and delivering concrete solutions to common problems through co-investment and co-innovation*. We are 14 members and several partners with common goals, sharing similar needs in terms of data dissemination and production, who are interested in mutualising costs and sharing knowledge, and committed to foster common standards. The community aims to provide an international collaboration framework to develop reference open source software for official statistics, guided by our foundations: Community-driven dynamics; Open source delivery; (macro)data lifecycle coverage; component-based and scalable architecture. The main output of the Community is the *.Stat Suite*, a standard-based, componentised, open source platform for the efficient production and dissemination of high-quality statistical data. The product is based on the General Statistical Business Process Model ([GSBPM](http://www1.unece.org/stat/platform/display/GSBPM/GSBPM%2Bv5.0)) and the Statistical Data and Metadata eXchange ([SDMX](https://sdmx.org/)) standards.

## Why go Open Source?

A strategic decision in 2018 by the Community Strategic Level group to make the *.Stat Suite* fully open source, and bring our common solutions to the world at large and increase our collective capacity to continue to deliver on our shared strategic objectives. A number of key influences led to this decision, including that to a large degree statistical organisations share common needs and similar processes, yet, official statistics are still a niche market driven by specific requirements especially in terms of quality and compliance with standards, and a strong culture of international, open collaboration. This probably explains why, thus far, no sustainable offering has truly emerged on the ‘official statistics market’. The community of official statistical organisations have to continue to join forces to co-innovate and co-invest in the next generation of practices and solutions to meet common challenges. Most of us will not be able to cope with these challenges alone. We have also seen that the diversity of organisations’ profile and priorities can be both a strength and an opportunity: for the stronger organisations to emulate each other and mutually leverage their investments, for the less advanced organisations to catch up, and bring in their qualitative contribution and creative ideas.

Based on the above, we strongly believe that the open source community model is the right way to fill this gap and here is where the SIS-CC has come in. [2] An open source community model means a lot more than just making source code accessible – as can be observed in many digital strategies, the complexity lies in the execution of it, the how, and sharing on lessons learnt can hence be invaluable to future similar undertakings.

# Methods

Although the main area of our collaboration focuses on the *.Stat Suite* source code and the management of it, there are a number of other topics that you must consider when moving to a fully open source mode of operation. These include contributions, legal aspects, documentation, rollout strategy, and the different communication channels needed to expose the project to a public audience.

## Do your research

As early as 2017, the Community started to analysis the possible paths to open source, considering various models such as hybrid, fully, or phased. At that time mainly driven by a number of constraints in the technology and supporting environment. We engaged an external expert, OSS Watch, a UK advisory service for issues relating to free software and open source software based at the University of Oxford. In August 2017 a report [3] was produced following a due diligence study of the ‘as is’ state of the source code repositories, operational aspects, governance, that provided guidance on a number of considerations such as licencing, communications, and managing contributions. In mid-2018, the community undertook its own investigations drawing on much of the considerations identified by the OSS Watch report. This then led to the strategic decision and a subsequent engagement with Open Knowledge Foundation (OKFN), who oversee the highly successful open source project CKAN [3], from which we took a lot of inspiration.

## Automate the delivery

Without a doubt the way in which the source code is managed and the delivery of it is fundamental to the success of an open source project. Concretely, we talk about Continuous Integration (CI) and Continuous Delivery (CD). CI is the best practice to follow to continuously develop a complex software with constant code checking to ensure new code does not break the existing code. In Continuous Delivery, it is important to approach the question “How can we easily manage the release of the project?” This process can, and must be to a certain degree, automated to gain maximum efficiency.

This brings us to the overall operational mechanism, referred to as DevOps [4], the set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the development life cycle and provide continuous delivery with high software quality. DevOps is complementary with agile software development.

## Harness the technology

Key to the success of a DevOps chain is the harnessing of ‘container’ technologies (Docker, Kubernetes) which allow for fluid provisioning (installation, update) of an integrated set of components constituting an application over the cloud. Our observation has been that few statistical organisations have embraced those technologies and practices – and the introduction of *.Stat Suite* has been an opportunity for many of them to learn and progress in this promising domain. The harnessing of those technologies, sometimes through a difficult dialogue with the organisations between business and technical teams; the adaptation of the IT policies and practices to this new world; the appropriate IT security measures to ensure frictionless delivery does not entail additional cyber-risks, these are all key enablers for the journey to succeed.

## Leverage other open source projects

Another important aspect, in harnessing technology in the context of an open source project, concerns the decisions on the integration of third party components – essential to leverage innovations in other community’s and avoid re-inventing the wheel. Functional, technical and legal considerations must brought to the Community’s Architecture Task Force in order to make decisions that engage the Community for the long term. Key decision points for SIS-CC have been, for example: to rely on Eurostat sdmxsource project [5] for the development of core APIs; or on Apache SolR for the common search engine, Facebook react.js as JavaScript development framework, etc. Developing connectors with other open source projects (such as Drupal, WordPress) and market standard tools (Excel) is another important aspect.

## Streamline the development process

It is natural for developers to adapt the single-sided development cycle in a non-open source project. However, in an open source project, it is necessary to differentiate between the two cycles – the parts performed by the development community within the project and the parts performed by the user community that have adopted the software. In an open source project, you need to ensure that you test from the point of view of an external user on a fresh environment, which will help to further refine documentation and resolve any issues.

The development process of an open-source project can be divided into 3 different perspectives: Developers/contributors; Users; Project maintainers.

A key role in the project is that of **‘The project maintainers’** who collectively are responsible for orchestrating the open source project covering a large array of topics: 1) Communicating with developers and users; 2) Creating online content; 3) Facilitating and moderating discussions; 4) Responding to online queries; 5) Developing and enforcing policies and code of conduct; 6) Recruiting and on-boarding new community members. (Specifically for our project, it means either facilitating the member organisations’ contribution or totally outside developers); 7) Resolving conflicts. Animating a community-wide, constant conversation to share a common understanding of the project and the priorities, and have all stakeholders contribute to and influence it, is the foundation underpinning all of the above activities.

## Legal

Engaging early with the legal team ensures they are on-board, going open source is not creating undue liabilities, and IP aspects are properly managed. Licencing, of the software to be open sourced is an important consideration, but also the dependencies that could influence the decision on which licence to apply. You also need to consider managing contributions from a legal perspective, including possible Contributor Licence Agreements (CLA).

## Security

Engaging early with the IT security team is key to ease the process of making any software, let alone a strategic platform, openly available, exposing the source code to potential malicious attacks. Putting the right processes and safe guards in place (such as ongoing automated code vulnerability checks, penetration tests, as well as regular security audits by a third party), especially concerning the contribution to the source code, has greatly reduced the risk to our final deployment application.

## Documentation

This can become an open source project greatest asset to ensure ease of adoption, ease of use, and ease of deployment. When moving the project to open source, we need to be able to communicate the entire necessary document to both users and contributors. It is essential to deliver the messages in the right way for an increased chance of adoption of the software and building a successful and thriving open source community. We first need to distinguish between the needs of just users of the software and contributors (developers), because developers approach from a different viewpoint where they need to know and use the software before they decide they will contribute it, and secondly they need to know the technical structure for them to build upon after the decision to use has been made.

# Results

In June 2019 at the occasion of the SIS-CC annual workshop, we launched in full open source the .Stat Suite platform. Since then the project has evolved gaining wider recognition with multiple projects and initiatives taking shape including in **support of the global agenda** and SDG reporting, **supporting sectorial organisations** (e.g. ILO) in the dissemination of Labor Market statistics via SDMX Web Services, **implementation of .Stat Suite on the United Nations Global Platform infrastructure** to provide a low cost service (SaaS) offering and developing statistical infrastructure in low-middle income countries. A **developer Advocate role** put in place to act as the bridge between the core team (Project Maintainers) and the wider community, bringing much value add to the overall process and quality of them. A **.Stat Academy** (e-learning environment)is now starting to take shape to support the user community (**Data Toolers** and **Data Producers**) in the building of capacities to enable wider and quicker usage of .Stat Suite. All this is being enabled through the creation of open source assets that are easily shared and reused across different contexts, underpinned by the open source framework now in operation.

# Conclusions

## Key success and enablers

It is imperative to have well-defined and readable user documentation so new contributors and users gain an understanding about the software itself. Do not try to be perfect on launch, all open source projects need time to evolve and a key aspect of building a community is that the users feel part of the journey. Automate whenever possible, using tools and techniques (DevOps) to save time and reduce the delivery burden on the core maintainers. Harness the appropriate technologies and connect to other open source communities to avoid reinventing the wheel. Make sure the core team and the larger community are fully brought into the vision, address concerns and issues early; ultimately, they will be the ones who make or break the process of moving to an open source project. Finally, do not underestimate the need to communicate, be open and transparent about what you are doing and why you are doing it.

# References

1. SIS-CC 2020-2025 Strategy, Foreword by Paul Schreyer, Chair of the SIS-CC Strategic Level Group (SLG), Acting Chief Statistician, Director of Statistics and Data, OECD, <https://secureservercdn.net/160.153.138.219/0nn.103.myftpupload.com/wp-content/uploads/2020/04/SIS-CC-2020-2025-Strategy.pdf>
2. SIS-CC Move to Open Source - final report, by Scott Wilson, Rowan Wilson and Simon Whittemore, 18th August 2017
3. <https://ckan.org/>
4. <https://en.wikipedia.org/wiki/DevOps>
5. <https://ec.europa.eu/eurostat/web/sdmx-infospace/sdmx-it-tools/sdmx-ri>