Successful Tool Re-Use in Open Contracting

A Primer

THE ENGINE ROOM

OPEN CONTRACTING PARTNERSHIP

THE WORLD BANK
IBRD • IDA

A starting point for anyone interested in taking advantage of existing tools to publish or use open contracting data.

This primer is based on research commissioned by the World Bank, conducted by The Engine Room in collaboration with the Open Contracting Partnership from May to October 2019.

Research and writing
Grace Higdon, The Engine Room

Writing, editing and project management
Helen Kilbey, The Engine Room

Review and feedback
Julia Keseru and Zara Rahman,
The Engine Room.
James McKinney,
Open Contracting Partnership.
Kristina Aquino and Samuel Garoni,
the World Bank.

Design
Dimitri Stamatis

With thanks to all of those we interviewed for this project, who generously contributed their time and expertise.

Acknowledgement and quotes from this publication to be referenced as: The Engine Room. Successful Tool Re-use in Open Contracting: A Primer, November 2019. This publication is available at https://theengineroom.org and https://open-contracting.org.

Contents

Successful Tool Re-Use in Open Contracting: A Primer

This primer is a starting point for anyone interested in taking advantage of existing tools to publish or use open contracting data.

The steps in this primer are designed to offer organisations a clearer view on what tools are out there, how to find support, how to evaluate whether a tool is the right fit for their goals and context, and what will contribute towards the successful re-use of their chosen tool.

For support in implementing an open contracting initiative more broadly, practitioners are encouraged to get in touch with the Open Contracting Partnership directly.

The examples and quotations included in this primer are taken from interviews conducted as part of our research.

Section 1

What tools are out there?

What tools are out there?

There are a number of open contracting tools[1] out there. Not all of these, however, are good candidates for re-use. Many are complex tools created with a single use case in mind (e.g. the Ukrainian platform Prozorro), which makes re-use in a different context more complex.

In recent years, however, efforts have been made to create tools that lend themselves more easily to re-use. For the most part, these are smaller, less complex tools built for specific utility – that can be used alone or together – rather than large platforms.

Types of tools available for re-use

The Open Contracting Partnership maintains a continually-evolving directory of tools for re-use. The directory focuses on tools that can be used to create, use, visualise and analyse OCDS[2] data; most of the tools included offer a combination of these functions.

Many of the tools in the directory support the more technical aspects of working with OCDS data. These include tools designed to:

- **Create and publish OCDS data.** This category covers a variety of technical functionalities, such as data entry and merging data. **Example tools**: Contrataciones Abiertas and OCDS Merge.
- **Convert data**. These tools convert data from one format to another – for example, from JSON (the format OCDS requires) to CSV (in order to analyse the data as a spreadsheet) and vice versa. **Example tools**: OCDS Toucan and Flatten Tool.
- **Check the quality of data.** These tools are designed to check that a dataset conforms to the Open Contracting Data Standard, and flag where there are any issues. **Example tools**: OCDS Data Review Tool and jOCDS Validator.

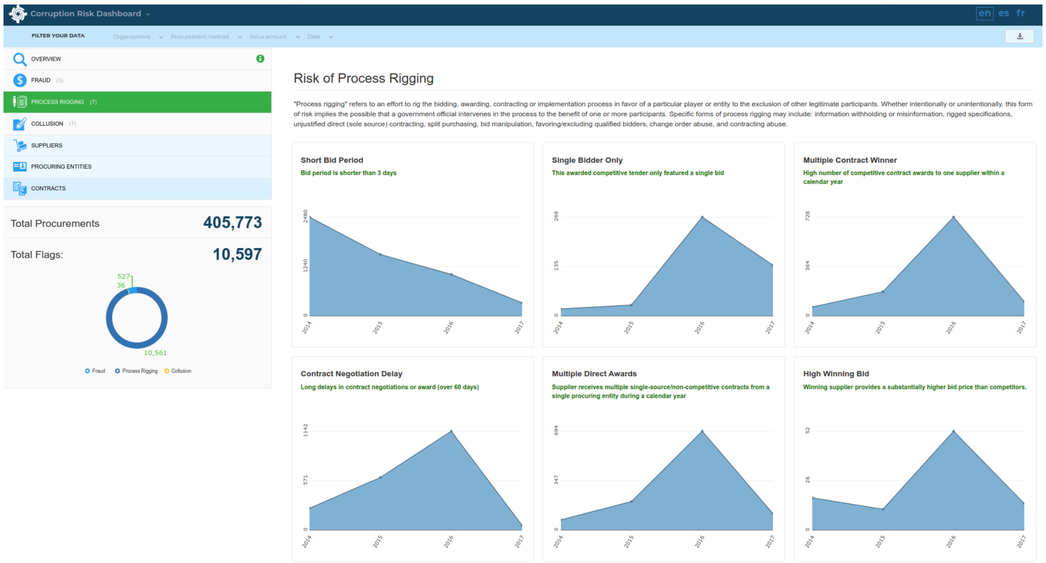
[1] Software, including web applications and command-line tools.

[2] OCDS refers to the Open Contracting Data Standard – an open data standard that formalises how contracting data and documents can be published in an accessible, structured and repeatable way. For more on the OCDS, see the OCDS website, managed by the Open Contracting Partnership, as well as the Open Contracting Partnership's website, which offers explanations and use cases.

Other tools are designed to interpret and tell stories about data sets. These include tools designed to:

- **Analyse OCDS data.** These tools can be used to collect and prepare data for specific analysis. **Example tool**: OCDS Kingfisher.
- **Visualise OCDS data.** Visualisations offer a way to make sense of data graphically, for example through graphs, charts, maps and diagrams. There are a number of tools available that are designed to both analyse and visualise OCDS data – **examples** include Tower Builder, the Corruption Risk Dashboard, and the Monitoring & Evaluation Dashboard.

Example Tool: Corruption Risk Dashboard



Total Procurements 405,773

Total Flags: 10,597

Risk of Process Rigging

"Process rigging" refers to an effort to rig the bidding, awarding, contracting or implementation process in favor of a particular player or entity to the exclusion of other legitimate participants. Whether intentionally or unintentionally, this form of risk implies the possible that a government official intervenes in the process to the benefit of one or more participants. Specific forms of process rigging may include: information withholding or misinformation, rigged specifications, unjustified direct (sole source) contracting, split purchasing, bid manipulation, favoring/excluding qualified bidders, change order abuse, and contracting abuse.

Short Bid Period
Bid period is shorter than 3 days

Single Bidder Only
This awarded competitive tender only featured a single bid

Multiple Contract Winner
High number of competitive contract awards to one supplier within a calendar year

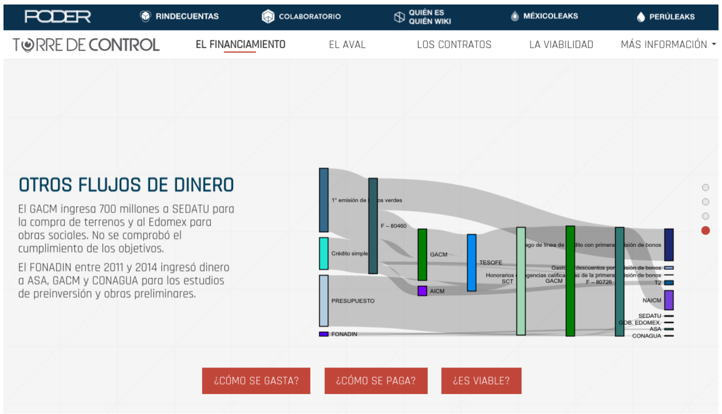
Contract Negotiation Delay
Long delays in contract negotiations or award (over 60 days)

Multiple Direct Awards
Supplier receives multiple single-source/non-competitive contracts from a single procuring entity during a calendar year

High Winning Bid
Winning supplier provides a substantially higher bid price than competitors.

Part of a suite of tools for data visualisation and in-depth analytics called the Open Contracting Explorer, the Corruption Risk Dashboard (CRD) is an open source tool that aims to help its users see where corruption might be at play in public contracting. Using a red-flagging approach, the tool visualises 10 indicators mapped to three different forms of corruption: fraud, collusion and process rigging. The code and documentation for the suite can be found on the tool's GitHub repository, and the tool author, Development Gateway, can be contacted directly for guidance on re-using the tool.

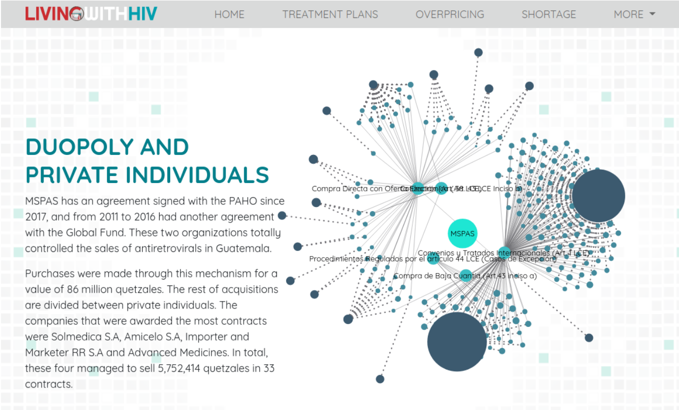
Example tool: Tower Builder



OTROS FLUJOS DE DINERO

El GACM ingresa 700 millones a SEDATU para la compra de terrenos y al Edomex para obras sociales. No se comprobó el cumplimiento de los objetivos.

El FONADIN entre 2011 y 2014 ingresó dinero a ASA, GACM y CONAGUA para los estudios de preinversión y obras preliminares.


LIVINGWITHHIV
HOME
TREATMENT PLANS
OVERPRICING
SHORTAGE
MORE
MSPAS
Compra Directa con Oferta Electrónica (Art. 43 LCE Inciso b)
Convenios y Tratados Internacionales (Art. 1 LCE)
Procedimientos Regulados por el artículo 44 LCE (Casos de Excepción)
Compra de Baja Cuantía (Art.43 inciso a)

DUOPOLY AND PRIVATE INDIVIDUALS

MSPAS has an agreement signed with the PAHO since 2017, and from 2011 to 2016 had another agreement with the Global Fund. These two organizations totally controlled the sales of antiretrovirals in Guatemala.

Purchases were made through this mechanism for a value of 86 million quetzales. The rest of acquisitions are divided between private individuals. The companies that were awarded the most contracts were Solmedica S.A, Amicelo S.A, Importer and Marketer RR S.A and Advanced Medicines. In total, these four managed to sell 5,752,414 quetzales in 33 contracts.

Tower Builder is an open source system that generates websites with data visualisations that mix open contracting and beneficial ownership data. The tool was originally built by Poder Mexico to follow the money in large contracting processes, and has been used to tell stories around plans for a new airport in Mexico and medicine procurement in Guatemala. The code for Tower Builder can be found on the tool's GitHub repository and the documentation for re-use can be found in both English and Spanish.

Section 2

Build a tool from scratch or re-use an existing tool?

Build a tool from scratch or re-use an existing tool?

Factors to consider

There are two options available to a potential open contracting tool implementer.

1. **Build a tool from scratch.** This is the default approach, especially for teams who have developers in-house.
2. **Use, adapt or replicate an existing tool.** This means using the code of an already-built tool or system for a new project, in a new context. Re-use might also include:
   - adapting the existing code to respond to a new set of needs
   - sharing knowledge and experiences around a specific tool or open contracting approach.

Which option is better? The answer here is, of course, *it depends*.

The table below offers a brief overview of some of the **advantages that each approach could potentially offer, given the right conditions**,[3] as well as some of the pitfalls that could be encountered. Section 4 gives more in-depth guidance on how to weigh up whether re-using a tool is likely to be the best option.

Building a tool from scratch

Potential advantages	Potential pitfalls
• Can ensure that the tool is suited to the needs of the specific context in which it is being built and that it conforms to specific legal parameters. • Can develop software in preferred programming language. • Can have a full overview of the project and code from beginning to end.	• Resulting tool will not yet have been tested in any context. • No pre-existing support community around the tool. • Can involve high development costs.

[3] The advantages and potential pitfalls listed here are based on the experiences of those we interviewed for this research, as well as prior research conducted by the Engine Room on tool selection in transparency and accountability initiatives.

Re-using an existing tool

Potential advantages

- Can take advantage of existing functionalities that fulfil a specific need (i.e. don’t have to reinvent the wheel), which can save time and money.
- Can potentially access an international network of people and organisations working on related issues (e.g. other tool re-users, tool authors and support organisations).

  “We could share in what we were doing - sharing code, knowledge, experiences - this was a rich experience. That was the great part.”
- Can take advantage of updates and improvements made to the original tool.
- Can access support in implementation and maintenance from people familiar with the tool.

Potential pitfalls

- Can encounter surprises half way through implementation due to, for example, missing documentation, unexpected hurdles hidden in the documentation or tool infrastructure, or out-of-date dependencies.
- Developers might have to work in a programming language or framework that they are not familiar with.
- Roles and responsibilities can be unclear between the original tool author and the new implementer (e.g. whose job is it to update the core code of the tool?).
- Tool might not be as suitable to the new context as originally thought, or might need more adaptation than expected.
- The costs involved in re-using the tool might be higher than expected.

Section 3

How to identify suitable tools

How to identify suitable tools

A step by step guide to help find tools that fit

This might seem obvious, but it stands to be repeated: every context is different. Procurement cycles, needs, priorities and environments are different, and these differences need to be taken into account when choosing tools. There is no one-size-fits-all, plug and play tool or system that will work everywhere, in every context.

Examples
In our interviews, we spoke to the architects of a large, complex portal for publishing federal procurement contracts in Nigeria, as well as to a data analyst using smaller tools to analyse large contracting data sets published by the government of Mexico.

Each worked within **different political and contextual frameworks** and had very different needs – both technical and in terms of responding to differences in procurement cycles between the two countries.

The following steps, adapted from The Engine Room's *Alidade* tool,[4] are designed to help an organisation figure out which available tools might be good candidates for their open contracting project.

3.1 Define the project's core objectives

What does the project want to achieve? Try and answer this question with a short, clear statement. Core objectives might be things like:

- Streamline efficiencies in the procurement process and deliver better value for money for the government.
- Create fairer competition and a level playing field for smaller enterprises.
- Provide an outlet for civil society and law enforcement to analyse government procurement data.

[4] Designed to help in creating plans for technology tools to use in social change projects.

- Monitor risk indicators that can help to identify potential wrongdoing or inefficiencies in contracting processes ('red flags').
- Drive quality standards higher for goods, public works and services.
- Promote smarter analysis of and better solutions for public problems.

For more support in setting goals, see the list of resources in Section 1.1 of *Alidade*: What is your project's objective?

3.2 Define the tool's intended users, and the needs of these users

- **Who is expected to use the new tool?** Create profiles of different user groups. In an open contracting project, this list might include government departments, journalists and civil society organisations.
- **Why will each of these user groups need or want to use this tool?** For each user group, write down a list of needs. For open contracting projects, these might include things like:
    - publish contracts
    - enter data
    - visualise data
    - compare specific data sets.

  "You really have to have user needs and understand what their interests are, why they want to use it."

- **What are the average technical skill sets of the target user groups? What are their levels of data literacy?** Try and answer this question for each user group.

3.3 Define what the tool is expected to do

- **What features does the tool need?** Create a list of technical functions. For an open contracting project, this might include:
  - produce data visualisations
  - convert JSON data to CSV file format
  - enable published contracts to be viewed online.
- **Divide the list of features into three categories:** essential (for the most important features), desirable (for less important features), and nice-to-have (for features that you could manage without).
- **How will each function allow the project to meet its objectives?** Assess each function one by one. For example, data visualisation might provide a way for civil society and law enforcement to analyse government procurement data.

3.4 Look for tools that have the features the project needs

- Start by going back to the Open Contracting Partnership's directory of tools. Do any of these tools have features that will help the project achieve its goals?
- Make a shortlist of the tools under consideration, and organise them in order of preference. The next section will offer an evaluation framework to help determine whether re-using any of these tools is likely to be successful.

To go through these steps in more detail, and for lists of supporting resources and advice on how to trial a tool with intended users, see *Alidade*.

Section 4

Are the right conditions in place for successful re-use?

Are the right conditions in place for successful re-use?

An evaluation framework

The following framework is designed to help an organisation evaluate whether re-using a specific tool is likely to be successful in a new context. The evaluation framework is organised according to four categories: tools and ecosystem, implementation team, context, and sustainability.

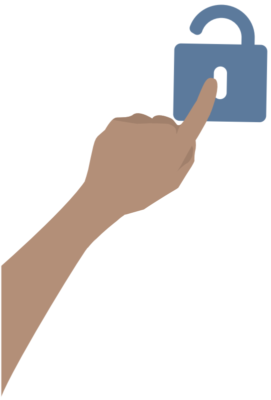
Use the framework below to evaluate each tool individually. For support in answering the questions in this evaluation, contacting the tool author directly is a good place to start. Other places to look for information include the tool author's website (if one exists) and the tool's GitHub repository and documentation.

The key questions of this section are summed up in the evaluation matrix that accompanies this document.

4.1 Tools and Ecosystem

4.1.1 Is the tool relevant to the needs of the new context?

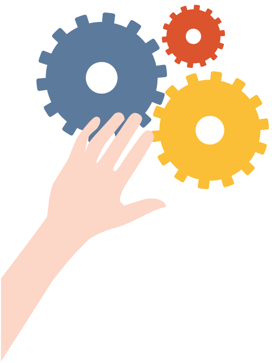
- What purpose, context (including regulatory environment and technological infrastructure) and user needs were the tool originally built for?



Examples

- Tower Builder was originally designed for citizens in Mexico with the goal of rooting out corruption in procurement contracts for a large infrastructure project.
- Budeshi was designed to be used by a government department in Nigeria to tackle corruption as well as to allow citizens to provide feedback to the government about public services.

- How closely do these map onto the new purpose, context and user needs? Are there any big differences or gaps?



Examples

- One organisation **struggled to transfer their tool** to a new sub-national legal environment because their tool was built around federal regulations which differed drastically between locations (e.g. some didn't have freedom of information laws).
- Another organisation was faced with **user needs they had not anticipated** – in particular, the need to communicate via SMS to give feedback on the platform due to a lower degree of internet access in the new context.

"Just because software can be plugged and played [in a new context], doesn't mean the service is the same."

4.1.2 Can the tool be adapted, and how much adaptation might be needed?

It is unlikely that an existing tool will fit a new context exactly and a degree of customisation is to be expected. However, if the tool needs so much customisation that the potential advantages of re-use (vs building from scratch) are lost – for example saving time and money, taking advantage of updates to the original tool, and so on – then it's worth considering whether the tool is a good fit.

In trying to evaluate whether it would be feasible to adapt a specific tool, consider the following:

- How much adaptability does the tool itself allow for? Are all the tool's features hard-coded in a way that makes them difficult to separate or adapt, or was the tool built in a modular way to allow for easy adaptation and updates?
- How possible will it be to adapt the tool but still be able to take advantage of updates to the base code?
- How much adaptation might be needed?

Alternatively, if the tool is big and complex, are there any less complex tools available that could be used together instead to address the project's needs? The OCP's tool directory includes a number of smaller tools, such as OCDS Kit, a suite of command-line tools for working with OCDS data.

Examples

Some adaptations that surfaced from interviews included:

- Changing the UX design interface of an open contracting portal to suit the preferences of a new audience in a new country.
- Adding different levels of user permissions within the tool to meet local government requirements.
- Building feedback mechanisms into the tool so that users could communicate directly with those managing the tool.

4.1.3 How is the tool licensed?

If code is licensed under a free and open source licence, it is legally available to be re-used free of charge. Ideally the entire tool ecosystem – all of the code – in the existing tool will be licensed as free and open source. Also look at dependencies. Are there any parts of the ecosystem that are proprietary, and will need to be paid for?

4.1.4 What support is available, and how can it be accessed?

When re-using an existing tool, adequate support (for both implementation and maintenance) can dramatically increase the chances of success.

Look for support from the following sources:

- **The tool author.** Can the tool author be contacted directly? The resources tool authors have to provide support vary, but many are able to provide support to some degree. Some might even have funding and a mandate to provide this.

  The types of support needed from a tool author could include:

  - installation help and troubleshooting
  - fixing bugs
  - updating the base code.

Engagement with tool authors can come in many forms, including in-person trainings or one-on-one meetings, or remote support via email, GitHub or calls.

- **Trainings.** Trainings can sometimes be offered offered by members of the open contracting community on specific tools or approaches.

- **Conferences.** Though conferences are more likely to be organised around more general topics, they might be good spaces to discuss and share learnings about specific tools and approaches, and to make contact with others working on similar problems.
- **The OCDS Helpdesk.** This is the place to go for support on tools authored by the Open Contracting Partnership.

Examples

- Open contracting support providers we interviewed described varying needs for support: "If they are a technical user, not much support is needed. But if they are a non-technical user, you have to provide step-by-step support to both install the tool and use the tool."
- Successful implementers we spoke to almost always had opportunities to engage directly with tool authors, implementation partners, or others within the tool ecosystem, both in the initial implementation as well as over the longer term. Most cited this support as a key factor in the success of the project.

4.1.5 Is there a re-use community centred around the tool?

Some tools have more of a community around them than others. Places where communities are formed, or where engagement and sharing take place, can include GitHub, feedback forms embedded in the tool itself, Google Groups centred around the tool, Slack channels and mailing lists.

Look at what opportunities exist to engage with others who have either re-used or tried to re-use the tool being evaluated – for example, how have the GitHub issues in a repository been responded to?

"[What works well is] an environment where there are tools, people using them, people contributing, making use of things that exist, a feedback loop."

4.1.6 Is there high-quality documentation of the tool?

If a tool's documentation is poor or has gaps, it's likely to be much more difficult to implement (particularly if there is a lack of in-person support).

When assessing the documentation of a tool, look for, at minimum, clear statements on what the tool does and how it does it, and a complete step-by-step outline of the setup process. Reference materials, use cases and examples can also be very helpful.

4.1.7 Will the tool be usable by its intended end-users?

Tools aimed at users who might not have a technical background should be easy and attractive for them to use. In open contracting projects, these types of users might include government actors, journalists, and the general public.

Look again at the project's defined target user groups, needs and skill sets. How do these match up with the tool?

"Really understanding your target audience's skillset...we're thinking, what is the target skill set for the tool and how do you build it for those needs?"

4.2 Implementation team

4.2.1 What skills and knowledge are required to re-use the tool?

When considering re-using a tool, it's important to look at whether there is sufficient capacity, knowledge and skill to carry out the project.

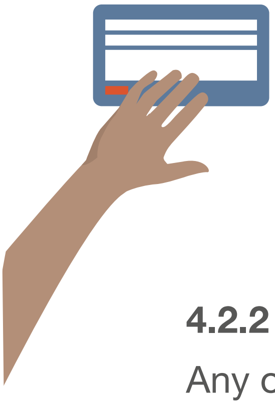
Depending on the tool, the skills and understanding required to successfully re-use a tool in a new context are likely to include some of the following:

- general familiarity with common data or programming terms like scraping, processing, or validating
- programming (in the relevant language)
- OCDS formats
- data conversion and validation
- command line interface
- GitHub and publishing code
- some specific concepts around open contracting, such as red flags
- other areas such as systems administration, web development, statistics, and/or marketing.

In evaluating a tool as a possible candidate for re-use, it can help to make a list of requirements. How does this list match up to the implementation team's current capacities?

Also consider how the team is structured. Specific roles that could be needed include:

- Project manager
- Programmer
- Systems administrator
- Web developer
- Data analyst
- Outreach manager.



Examples

- Some implementers we spoke to noted that outsourcing external software developers proved more difficult than having a software developer on staff to build and maintain the tool.

4.2.2 Are there sufficient resources to carry out the project?

Any open contracting project requires resources, and re-using an existing tool is no exception. In re-using a tool, costs can come from a number of areas:

- **Technical costs.** Are there any parts of the tool's ecosystem that will need to be paid for?

- **Human resources.** As is common in technology projects, this is likely to make up a large part of any budget dedicated to re-using a tool.

  "There is an assumption that open source is free – but all along the process, people's time spent should not be assumed to be cheap. It's much cheaper to install an open source thing than commercial, but it's far from free."

- **Consider different funding models**
  - **Joint funding.** This type of funding finances both the original tool author and the team re-using the tool (ideally over the long term, so that maintenance and updates to the tool are included).

Examples

MySociety's parliamentary monitoring platform TheyWorkForYou was most successfully replicated in Kenya and South Africa due in part to joint funding provided to both countries as well as to MySociety.

- **Long-term funding for maintenance and infrastructure.** For long-term sustainability of the tool, this is crucial.

4.2.3 Can a formal plan for tool maintenance be put in place?

It's increasingly recognised that forecasting and planning for tool maintenance is critical to a tool's long-term sustainability.

Languages and libraries tend to change over time – in short, as one interviewee put it, "the codebase [of many open source projects] gets a little old in the tooth." Both the authors of a tool and those who have re-used it might need to work to keep their tools up to date.

Additionally, if the foundations of a tool have been edited and added to by multiple users with different styles and approaches, without appropriate documentation it can be particularly hard for newcomers to the code to follow the logic and structure.

With these factors in mind, consider the following:

- Who will carry out software updates and at what pace?
- Who will fix bugs?
- How do users report issues with the tool and who is responsible for responding?

4.3 Context

4.3.1 Is the technological infrastructure adequate?

Tools are generally built with a certain infrastructure in mind. When weighing up whether to re-use a particular tool, it's important to make sure the necessary infrastructure is there to support it. This might include:

- regular access to computers in relevant places of use (e.g. government departments)
- reliable internet coverage
- widespread smartphone use.

Examples

In our research, technological infrastructure was cited surprisingly frequently as a big impediment.

"There are different challenges. Apart from capacity, another thing is having computers and internet. Some of [our partners in government] say that in their offices they don't have computers, they don't have internet. And all of this is online. The current mode of operation is files and physical copies."

4.3.2 What data is available?

In any open contracting project, availability and quality of data is crucial to the success of the project. Projects that re-use tools are no exception here.
In evaluating the data available, consider the following:

- What format is the data in originally? (e.g. PDF, Excel, CSV)
- Can this data be easily converted into a format that the tool can work with? How? If a tool requires people to convert the original data into a new format that the tool can read, are these people regularly available to continue to input data into the tool, and do they have the necessary skills to do so?
- If working with OCDS, how will the data be validated (i.e. checked for accuracy and quality before use)? There are tools available to ensure a data set complies with OCDS requirements; for example, the Data Review Tool.

Examples

Through our interviews, two key difficulties came up repeatedly: lack of data availability and lack of quality input data. Issues included:

- missing fields in the original data
- data input in different file formats
- translating data into OCDS
- data not being validated
- a lack of skills necessary to publish the data correctly.

"Where a lot of projects fail is that they have the idea all worked out but the data is not available. Really be clear about whether you have access to data you need."

4.3.3 What political buy-in exists for the project?

Where relevant (this will not be a consideration for all projects), maintaining high-level political buy-in from leadership, as well as buy-in across government departments, is crucial for success.

Examples

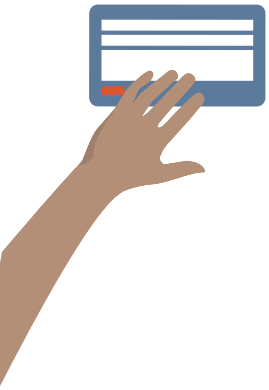
- In interviews, government buy-in was discussed as being a fundamental first step before entering into a discussion about tools. Having existing tools to demonstrate can be helpful here. Some interviewees described having success with in-person tool demonstrations, through workshops or one-to-one meetings with high-level politicians.
- One person we interviewed noted that they used lower-tech tools to explain what open contracting data could do, beginning with spreadsheets to illustrate examples before showing JSON files. Others cited the use of infographics and comics.

"If you need individuals from different departments to regularly enter their data, you need buy-in."

4.4 Sustainability

4.4.1 What feedback mechanisms exist?

Once a tool has been re-used, how will feedback be received from users? And how can implementers provide feedback to the authors of the original tool? This increases the chances that the project will be successful in the longer term, and contributes to the success of new projects using the same tool.



Examples

Some examples of feedback mechanisms discussed during interviews included:

- feedback forms built into an online platform
- interactions over GitHub
- emailing directly with the tool author
- mobile feedback mechanisms.

Section 5

Example work-through

Example work-through

The work-through below offers a simplified example of how an organisation might start conducting their evaluation

Example Scenario: A journalism organisation has collected data from an investigation in OCDS format. They would like to find a way to communicate their findings.

What are the project's core objectives?

To share the results of the investigation in an interactive, storytelling way.

Who are the tool's potential users?
What are their needs and what skills do they have?

Primary user group	Civil society, including other journalists.
Need to be able to	Easily navigate and get meaning from data collected around government contracts.
General level of technical skill	Users are able to navigate the web and understand graphs but are not necessarily able to understand code or work with databases.

What technical functions does the tool need?

Essential	Desirable
• can present data in the form of visualisations and graphs • web-based • uses OCDS data.	• attractive website layout • displays effectively on mobile.

Which tools have the features the project needs?

Tower Builder	A system that generates websites with data visualisations that mix open contracting and beneficial ownership data.

Are the right conditions in place for re-use to be successful?

Relevance	The tool was originally designed to follow the money in large contracting processes. This is relevant to the current investigation. The tool has already been proven to work in contexts that are similar to this project.
Adaptation	The tool is designed to perform a set of key functions. The main design elements of the website are customisable (e.g. colours, background image, logo).
Licence	Free and open source.
Support	Though the tool author does not have a formal structure in place to provide direct support, they can be contacted. The original tool is currently actively maintained and updated.
Community	The tool has a GitHub repository.
Documentation	Extensive documentation in English and Spanish.
Usability	Well-designed web-based tool – usability is high.
Skills and understanding required	The tool was originally designed to work only with OCDS data (requiring understanding of OCDS, JSON, and the command line interface). However, the possibility now exists to add data in CSV format as well as from an API. The tool also requires knowledge of Markdown format.

Moving forward

Moving forward

As the field of open contracting continues to grow and tools continue to be developed, organisations re-using tools are a critical part of the open contracting ecosystem. We hope this primer extends an invitation to new tool re-users, while also providing insights for existing practitioners.

More support and information can be found from the sources below.

- **OCDS Tools Directory.** As already mentioned, this is a living directory of tools that are currently available for publishing and analysing open contracting data. The directory is managed by the Open Contracting Partnership.
- **Inspiration Gallery.** Also managed by the Open Contracting Partnership, this gallery features open contracting tools that have not necessarily been designed for re-use, but that could nonetheless be useful as inspiration for new tools.
- **The Open Contracting Partnership's website.** This website includes:
    - Details on how to contact the OCP for support and advice.
    - The Open Contracting Journey. An overview of what an open contracting project might look like.
    - Information and tutorials on how to implement the Open Contracting Data Standard.
- **OC-Hub.** An online course created by Transparency International that provides an overview of open contracting concepts.
- **Red Flags For Integrity**. A methodology guide developed by the Open Contracting Partnership and Development Gateway that shows how to use open data in public procurement to identify corruption risks.
- **Budeshi Guideline**. A user guide developed by the Nigeria-based Public and Private Development Centre (PPDC). Alongside guidance on how to use the PPDC's open contracting portal Budeshi, the guide also gives a valuable overview of open contracting and OCDS.