From Development to Adoption: Lessons from Three Open Standards

A Report by OpenNorth for the Open Data Institute (UK)

Executive Summary

Through extensive documentation research of public and internal resources, and interview data from two lead standard developers at OpenNorth – Stéphane Guidoin (Open511) and James McKinney (Popolo, Represent) – we detail challenges faced in the standards development process and outline some recommendations for those in the open data community wishing to undertake their own data standard development.

Main Observations:

- Collaboration does not guarantee accurate use case (or persona) generation, particularly if potential adopters are not engaged
- Who collaborates matters - domain experts and IT experts have different skill and knowledge sets
- Software tools can help support adoption
- Support for standards is needed from software platforms used by data suppliers
- Standards can come after a product is created, and arise out of data needs

Recommendations:

- Create standards based on demand for data to ensure there will be adoption
- Create simple, lightweight standards
- Involve official collaborators as early as possible in the development process and maintain their engagement with the project
- Involve both domain experts (who have knowledge of content) and IT professionals (who will implement the standard)
- Create software tools to ease the process of adoption (such as data validators, conversion tools, and mockups)
- Seek software publisher support for the standard (such as enterprise platforms governments may rely upon)
- Factor in costs of outreach, facilitation, and maintenance

We conclude with a proposed model for standards adoption, whereby the rationale for adoption is supported through a software product, and adoption of a standard increases the quality of the same product, thereby facilitating a positive feedback loop.

Introduction

OpenNorth's three standards, Open511, Popolo, and Represent, are case studies of different approaches to standards development, and indicative of variability and uncertainty of collaborative approaches to standards development. At one end is Open511, a standard initiated with an explicit motive of collaboration and co-creation. At the other is Popolo, which was developed through much a less formal, but still consensus driven model. Finally, Represent's CSV schema itself represents a standard that arose out of existing data and development needs; not initially to be a collaborative effort (but remains an open standard). Represent, although the least collaborative, is arguably the most successful of the three standards. Meanwhile, Open511, the most structured collaboration of the three, has experienced the least adoption. **This brings us to question the notion of openness in data standards - does collaboration guarantee adoption?** We describe each standard based on existing documentation and present the main findings from interviews with the lead developers. From this data we outline a potential positive feedback model for standards adoption and present seven key recommendations for future standards developers.

What is an open data standard?

Ironically, the term 'standard' can be used in a number of ambiguous ways, or not at all. Traditionally, data standards can be placed at multiple levels of a hierarchy; a 'standards-stack' (Bloom & Sieber, n.d.), although Bloom & Sieber contend this hierarchical view implies assumptions of quality and thus is not accurate. Standards can range from atomic level standardization of individual terms (such as an ISO standard for date formats or geospatial coordinates), to the arrangement of terms (dictionaries and schemas), to compliance (quality or compliance certification), and even the processes governing data use and exchange. The term 'standard' has been used to refer to all of these levels, or can sometimes be omitted, which can create confusion. For example, the United States' National Information Exchange Model (NIEM) is referred to as a common vocabulary for information exchange, but in fact consists of both a dictionary and schema. Meanwhile, the United States' Green Button initiative is referred to as a standard but is, in fact, a certification programme for compliance with a data standard (the North American Energy Standards Board's Energy Services Provider Interface standard).

'Open' in a standard can also be applied at multiple levels. An open standard can involve interoperability with other standards and technologies, or it can refer to the governance of the standard's development itself. A common example is the General Transit Feed Specification (GTFS), which originated as the Google Transit Feed Specification but achieved such widespread adoption that it became a standard. The standard is developed in collaboration with the community, with any stakeholders able to create requests and vote (provided there are two adopters of the proposed change), but is still governed by Google.

Finally, the difference between a data standard and a commonly used and accepted data format (a de facto standard) can also be ambiguous. The Comma Separated Values (CSV) format, a file format defined by the Internet Engineering Task Force (IETF) is an open format, but is not standardised.

We hope the results in the rest of this report shed light on the varying levels of openness a standard can take.

Method

OpenNorth's three standards (Open511, Popolo, and Represent) were treated as separate case studies. These case studies were informed by documentation research and semi-structured interviews. Documentation research included data collected from publicly accessible documentation, Google Groups discussion fora, and online videos and presentations. Internal documentation, such as slide decks, grant proposals, and reports, were also analysed. Documentation was screened to develop the narrative of each case study, as well as analyse for themes such as barriers to standards adoption. For semi-structured interviews, two lead standard developers were interviewed: Stéphane Guidoin (Open511) and James McKinney (Popolo and Represent). A supplementary interview was conducted with Michael Mulley, a developer who created the technical implementation of Open511, to triangulate results. For these interviews, an interview guide was developed to collect developer perspectives on the development process, and draw out their views on the ideal standards development environment. The interview guide can be seen in Appendix A.

Results

This section describes each standard and is informed through documentation research of publicly available information, such as discussion groups and API documentation, and results of interviews. First, an overview of each standard is given. Then, major themes from interview results are presented.

Open511

http://www.open511.org/

Open511 is a data standard for road events, such as various types of roadworks, cultural events, and road closures. It does not cover traffic conditions, but does support traffic camera data. It was designed for real-time implementation as a web or mobile app, via Application Programming Interface (API), in either XML or JSON. The project was funded by Natural Resources Canada's Geoconnections program.

Current standards for road event data are highly focused and technical, such as the Traffic Management Data Dictionary (TMDD) using in North America, and Datex II in the European Union. These standards are designed for traffic management, specifically communication between traffic centres. Therefore, there is a gap in the 'standards market' for government-to-government coordination and government-to-citizen communication of road events. Open511 can be viewed as a much simpler, and more focused implementation of the TMDD standard.

In addition, numerous small municipalities either do not have the resources or need for a large traffic control centre, and therefore do not have a need for a complex standard to structure their road event data. As such, Open511 can cater to governments both small and large, civil society organizations, the civic tech community, and the general public.

The two major institutional collaborators for the project were: Ministry of Transportation and Infrastructure (Province of British Columbia [BC]) and the Metropolitan Transportation Commission (San Francisco Bay Area). These collaborators were secured with the intention to co-develop the standard and place them as first adopters. Official collaboration was done through a series of meetings, email exchanges, and conference calls. Outside of these two collaborators, members of the community can also provide feedback and change requests through an open Google Group. Current adopters of the standard include the Province of BC's Transportation and Infrastructure Ministry, through its real-time DriveBC dataset, and the Town of Repentigny (Québec).

OpenNorth led the development of the standard at all levels. For each iteration, a proposal was put out and a call for feedback was sent to collaborators. For a change to be considered and included in the standard, two potential adopters were required to acknowledge their support of the change.

The standard was created in three levels, or tiers, of development:

1. Definition of terms - select which terms to use and their relationships to each other
2. Attributes for terms - select the characteristics for each term
3. Values of attributes - determine the possible value range for each term

A high-level term was defined first. Existing namespaces were checked. Then, attributes the given term were defined. Finally, the potential range of values these attributes could possess were defined. This iterative process was repeated for all terms in the standard, with requests for feedback sent out regularly to collaborators.

```
"jurisdictions": [
    {
      "id": "my.city.gov",
      "name": "My City",
      "url": "/api/jurisdictions/my.city.gov/",
      "description": "Official traffic data (construction) from My City",
      "description_url": "http://my.city.gov/open511/presentation.html",
      "geography_url": "/jurisdictions/my.city.gov/geography/",
      "languages": [
        "fr",
        "en"
```

```
          ],
          "phone": "+1 555-555-0000",
          "license_url": "http://my.city.gov/opendatalicense/",
          "timezone": "America/New_York",
          "email": "mayor@my.city.gov"
       }
   ],
```

In the above example, ‘jurisdictions’ would be determined in the first phase. In the second phase, attributes such as ID, name, url, and timezone, would be assigned. Finally, the potential range of values would be assigned for each attribute. The possible range of values was determined based on existing resources if possible, to be interoperable with other standards and software. Existing standards, such as the ISO 8601 standard for date formats, were utilized to define the potential range of attribute values. This extends to place names, with the use of the GeoNames database (a free, crowdsourced database of geographic place names), and the geographic coordinate system (WGS84) was chosen due to popularity and its compatibility with Google Maps.

In addition, a conversion tool was built to convert from TMDD to Open511 data, to increase interoperability with traffic management systems. However, since TMDD data is vastly more complex to Open511, this tool was not validated to work for all use cases. Finally, a data validator was also built to aid developers in adoption.

Popolo

http://www.popoloproject.com/

Popolo is a standard information on legislative activities. It helps civil society developers (including non-profits) structure legislative data within their apps (such as websites monitoring parliaments) and share data among each other. Unlike Open511, it is aimed squarely at civil society, not government. The standard structures information on individuals (e.g. contact information for Members of Parliament) and events (e.g. motions, bills, votes, speeches). Through Popolo, developers can structure information on events such as a vote and its outcomes. Popolo defines the format and data model so that civil society organizations and civic tech developers do not have to model or transform their own data, increasing efficiency in their data operations.

Popolo is part of the Poplus community of civic tech components, and was developed with the aim to provide interoperability between different civic tech projects and allow developers to reuse code (such as a website monitoring legislature) across different countries. It has been adopted by a number of organizations outside of Canada, including mySociety and OpenPolis.

The development process for Popolo is explicitly stated on its documentation website as a three-step process:

1. Identify use cases and requirements for an activity system

2. Research prior work relating to the use cases and requirements
3. Draft a specification and improve it through rough consensus

In drafting a specification, as with Open511, high level terms (or classes) were defined first, then attributes and attribute values. For example, a 'motion' in the standard was first defined in plain language, with existing namespaces and data standards checked for potential overlap. Then it was examined for potential use-cases and requirements. A motion was defined as requiring a creator (someone who proposed the motion). This is then reflected in the schema. At this level, the developers linked attribute fields to existing vocabularies where possible to increase compatibility. Finally, this was serialized to JSON variants and RDF to become: "creator_id": "john-doe",

Standards interoperability was also pursued, with the use of GeoNames, ISO 8601, Dublin Core, Schema.org, ISA Programme Location Core Vocabulary (a W3C standard), and more.

Notably, Popolo was developed through a process of 'rough consensus', inspired by the IETF. This allowed Popolo's development to be flexible, with decisions propelled through a github issues tracker and mailing lists. This included a basic 'rule of two', requiring a minimum of two individuals to express support for a requested change before it would be considered.

Represent

https://represent.opennorth.ca/government/

There is no single repository listing every single elected official in Canada. Different jurisdictions list their elected officials on their own repositories, without standardisation across Canada. Because of this, Represent was started as a project to scrape this data and provide it in a standard format through an API. Through the Represent API anyone can query an elected official's basic job and contact information. The Represent project consists of an API, a server and database, a number of scripts to download and parse data, and finally a simple CSV schema. This schema represents the standards component of Represent.

Through an initial outreach and adoption drive instigated by the lead developer, Represent's CSV schema has been adopted by over 30% of Canada's open data-providing municipalities (with adoption continuing on a proactive basis in the proceeding years), including the City of Vancouver and City of Ottawa. Instructions are provided to governments on how to adopt the standard. These are a simple list of steps that outline the schema, and the procedure to notify OpenNorth (via an email address). The CSV schema itself is simply a set of fields, or column headers, with no hierarchy. Because of this, any data record can be represented on a single line in a CSV file.

Unlike Open511 and Popolo, Represent's CSV schema came about after a product (the Represent API) had already been developed and gained significant usage. The lead developer therefore based the schema on observed patterns, and their own accumulated knowledge, in existing data.

Represent's schema appears as the following flat structure. There are 21 supported terms, with 7 bolded terms stated as the most important. Adopters are encouraged to link back to the website with a statement of compliance with the schema. Adoption is counted when adopters submit a URL for their schema compliant CSV (which is then incorporated into the API), but otherwise no formal tracking or verification system exists to track other adopters.

Below is a sample of the first 9 main column headers, including the 7 recommended fields available in the schema. Since the schema is not strictly enforced, there are still varying levels of compliance among Canadian adopters, with some omitting district information or photos of representatives.

- **District name** (if your municipality is divided into wards, districts or divisions – Ward 2 for example)
- **District ID** (if your wards have identifiers as well as names – 2 for example)
- **Primary role** (Mayor or Councillor for most municipalities)
- **First name**
- **Last name**
- Gender (M or F for example)
- Party name
- **Email**
- **Photo URL** (a link to a photo of the elected official)

The full schema can be found at: https://represent.opennorth.ca/government/

Interview Results

In this section, we detail results from semi-structured interviews of our lead developers, organised into themes. We frame our results around the theme of adoption, and use this to inform a proposed positive feedback model for standards adoption.

What is an open standard?

Determining the operational paradigm of each standards developer was important, as assumptions on 'open' could dictate decisions and approaches to development. The first question asked of our lead developers was, "what is your definition of an open data standard, or an open standard for data?"

Some nuance was given to the definition of an 'open standard for data'. Our interviews with Stéphane Guidoin (Open511) and James McKinney (Popolo, Represent) revealed their views on ideal openness and collaboration in standards development. While both leads referenced the Open Definition (opendefinition.org) when defining 'open', they acknowledged that a 'standard' could be implemented in a number of ways.

Open511's lead (Stéphane Guidoin), focused on the democratic aspect of open standards, *"Ideally, it's the fact that the process of building [the standard] is open, people have a word in on* *how it is made. This brings in an idea of open governance"*. He also noted that standards could also come about as a result of market saturation, with reference to de facto standards such as GTFS.

The definition of a standard was also called into question. Some standards can be aspirational, *"something people put out there and they hope gets adopted"* (James McKinney, Popolo & Represent) while others can be authoritative (originating from a recognized and respected standards organization such as the ISO), and that a critical mass of adopters was needed for an aspirational specification to be accepted as a standard. Standards may also be labelled under alternative names. It was noted that, while W3C is an authoritative institution, their approach involves labelling specifications as 'recommendations' instead of 'standards', possibly in recognition of historical differences in status between formal standards organizations and other groups. These documents are still recognised as web standards.

Adoption

Adoption was recognised as a main goal of each project. However, the developer of Open511 added the category of software publishers as a key target for adoption. The respondent recognised that end users, including governments, often rely on large enterprise software to handle data collection and production. These platforms, such as a city's intelligent transportation systems, computer aided dispatch, and automated vehicle location systems, dictate the data structure that a city inputs and outputs. The developer stated that diverging from existing software that a government depends on is highly costly. Thus adoption would benefit from software publisher support for the standard, as it would negate the need for governments to develop and test conversion tools.

Popolo's lead stated that programmers do not necessarily think about data standards on a daily basis, "Many programmers are happy to adopt an existing data model that's good enough rather than design their own, because they prefer to focus on building the features that users will see. In this case, the standard solves their data modelling problem." In fact, "the programmer may not even care about the benefits of standardization, such as interoperability." He noted that data sources and libraries that implement a particular data model will usually be used without much thought. However, this activity still results in adoption of the standard, "[programmers] may not even know that the API or library uses a standard. But, by having been implemented by these 'upstream' services, the standard achieves 'downstream implementations'."

No standardized approaches were taken to track adoption. Interviewees used existing methods of communication (e.g. email, Github issues tracker). In the case of Open511 and Represent, a sales approach has been taken whereby a lead is developed and followed up. This is particularly the case with Represent, as OpenNorth hosts the Represent API as a free tool for public usage.

Social and political relations may also be important in standards adoption. Speaking on Represent's CSV schema, the lead developer noted that potential adopters are influenced by existing adopters.

“At first, there were 2 months where it was me advocating to these cities. Since then, there has been proactive adoption of the standard, by people whom we never contacted. I think the reason is that cities look at other cities, other comparable data portals, to see what datasets they are publishing and how – and existing publishers were linking to the CSV schema.” (James McKinney, Represent)

Finally, Represent’s adoption strategy was supported by grant funding from the Canadian Internet Registration Authority (CIRA), and required a certain level of convening on the part of OpenNorth. In an internal, 2015 progress report to CIRA (a funder of Represent), the lead developer noted the need for leadership in open data initiatives to promote standardization. Since individual government departments decide on their own data publication, it was observed that open data leads lacked the authority to mandate the publication of information on elected officials, let alone in a specific format. Because of this, their strategy in promoting adoption was to provide arguments supporting the use of Represent’s CSV schema as well as build up a critical mass of implementers. This involved displaying logos of municipal implementers directly on the Represent website as proof of uptake. Another vector to promote adoption were meetings and conference calls with working groups, particularly in the Ontario region.

Collaboration and Facilitation

Consensus featured in our interviews as a important factor influencing adoption. In particular, satisfaction of official development partners would determine their likelihood of adoption once the standard was complete.

Achieving consensus was a difficult process in Open511, relatively simple in Popolo, and even simpler with Represent’s CSV schema. In Open511, the role of the lead developer was more than simply to answer requests. Their role was to facilitate discussion among different stakeholders and create commonly accepted solutions. Because of this, they viewed a standard co-developed with many stakeholders to have more potential and relevance than one developed with just a single partner.

“we had two organisations developing at the same time, and it was good because if it had only been BC, maybe we would have gone live with something that doesn’t fit with other needs.” (Stéphane Guidoin, Open511)

This was a very different approach from Popolo and Represent, where a more opt-in approach was taken. The lead developer of Popolo and Represent placed less emphasis on ensuring co-creation than on achieving consensus itself.

“For example, someone has a concern. An important first step is to acknowledge their concern – to make them feel heard. Then, if you don’t agree, you need to carefully describe the history of how things came to be the way they are. Once they understand the constraints or thought that was put into it, they may understand. Don’t expect people to have as much knowledge of the

history; bring them up to speed. Be polite and understanding throughout." (James McKinney, Popolo)

However, they still felt their role was to facilitate discussion. Collaborators who were educated on the background behind a particular issue would also be more understanding of problems and constraints, thus increasing the likelihood of consensus.

"Another aspect of facilitating is asking questions – ask them how it will accommodate x scenario. It then forces them to think about it; they might come up with a solution, or they may acknowledge their proposal needs some work." (James McKinney, Popolo)

The question of who should lead standards development was also raised during interviews. The lead developer of Open511, Stéphane Guidoin, felt that it was important for conflicts of interest not to be present in the governance or development of a standard. If the lead developer of a standard was the same entity as the the main consumer, such as Google and its relationship with GTFS (Google is the primary end user of GTFS data due to its Google Maps platform), it would bias development and result in less collaboration,

"I felt that having a consumer as a lead of the standard was not a good idea, as they only promote their own use of the data, not other consumers, and doesn't fit publishers' ability to publish" (Stéphane Guidoin, Open 511)

Consensus was deemed to be important to eventual adoption, particularly when official development partners are involved.

For all three standards, funding limitations were acknowledged as the main factor preventing them from continuing standards development beyond the initial phase. Funding limitations, and the fact that these standards were highly personalised (they were led by only 1-2 individuals) meant that a management or governance team for long-term sustainability was not explored.

Tools to Aid Adoption

While standards help address the supply-side of data, both lead developers saw the need to support demand through technical implementations that would demonstrate the value of the data standard.

In addition, developing a technical implementation of a standard, such as Open511's demo and validation tool, gave developers more perspective on their standards development, "*It's the same for any software dev, but more difficult for standards. You need to check if the standard is working, but if you develop in a good old waterfall process, it doesn't work."*

Conversion tools, such as the TMDD-Open511 converter tool, were also viewed as important to reducing the barriers to adoption. Open511's developer stated that governments would feel more secure using a new standard if there was a tool that could ease transition.

Open511's development process was paralleled by a process of self-learning. Developing a technical implementation (the demo site and the Town of Repentigny implementation) of Open511 in parallel with the standard itself helped ground the standard development within a real programming environment, "*It was giving us a more concrete grasp on what we were doing."* (Stéphane Guidoin, Open511).

Origins - standards or software first?

Standards can originate from very personal developer needs, not just a need to structure data for the sake of it. Open511 came about as a result of the lead developer, Stéphane Guidoin, realising that his own web app visualizing road closures in Montreal was receiving increased requests for data. He realised that many requests were cross-jurisdictional, such as requests to extend road closure coverage to a bridge between the island of Montreal and the South Shore, and that a data standard would make the incorporation of additional data sources much easier.

Standards can also appear in order to fill gaps or complete voids in data availability, not just to improve upon existing standards. According to the respondent, Open511 was conceived at a time when road closure data was not being released publicly. The respondent mentioned that Google had started to incorporate road closures into its products, but data quality was extremely limited, with no details on the level of severity or activity and with no temporal accuracy (data published at the time did not reflect whether road closures such as roadworks were only occurring at certain times of the day).

However, Represent's CSV schema only came about *after* the API had already been developed and had gained significant following. The reasoning for standardizing data on elected officials was the same as in the case of Open511 - to increase efficiency in app development and reduce the costs of continually updating scripts that download, parse, and structure data from each jurisdiction. The difference in this case was the ordering of activity.

"The fact that there's a product out there (Represent API) that gets millions of requests each year and is used by political campaigns and regular citizens – knowing that there's a non-profit product they could help support for a variety of democratic and interesting uses." (James McKinney, Represent)

User Centred Design

Open511 attempted to incorporate stakeholder collaboration, while Popolo developed use cases in its design process. Popolo's developer found the development of use cases to be fairly straightforward. A focus on use cases was deemed sufficient to determine the requirements the standard would have to satisfy. Personas were not the focus of development, as a use case might be relevant to many different personas. Because of this, defining specific personas was not a priority, as long as use cases were enabled by the standard. Use cases were refined during the standard's development process, but were a result of differences in factors such as parliamentary procedures, rather than due to personas themselves. While personas were still deemed to be useful, Popolo's development happened to be lead by civic tech developers themselves. Because of this, and remembering that Popolo itself was designed for developers and civil society (not governments) to use, the standards developers were confident in their ability to predict use cases for other developers.

However, in the case of Open511, pre-determined use cases were found to be inaccurate as additional collaborators became involved later on in the process. While one version of Open511 was acceptable for the Province of British Columbia, it was deemed insufficient for the MTC's specific context. One example of incompatibility was the issue of road events crossing between days (such as roadworks at midnight). Some collaborators' systems could not support road events crossing over midnight (as they were time-bound to discrete days). Unfortunately, splitting up road events by days was also a sub-optimal solution. *"In the end it was almost impossible to make scheduled events that match everyone's needs."* (Stéphane Guidoin, Open511)

The choice of collaborators can also impact the development of a standard. For Open511, it was discovered that transportation experts, who were the main points of contact within each government collaborator, did not have the same views or understanding of technical implementation of APIs as their IT departments. While domain experts may be familiar with the uses of particular datasets, they do may not have the experience of database management. Since Open511's development was specifically developed to be implemented as an API, the choice of collaborator should also have included representatives from IT departments, who had knowledge of their own capacity to deploy internal and public-facing APIs.

Appropriate technology

Open511's lead developer explicitly stated that the project represents a misapplication of technology. The standard itself was developed specifically to be implemented as an API for real-time data. However, as development and collaboration ensued, it was discovered that the majority of road closures in municipalities are actually planned in advance. Because of this, the need for dynamic, real-time data was much less significant than initially thought. The lead developer noted that public-facing APIs require significantly more resources to implement than internal government APIs, and were thus a barrier to adoption. While releasing a static file, such as a CSV, on an open data portal simply requires a request to be made and some internal coordination, a public-facing API would require a full-fledged development project, project management, server testing, and significantly more labour resources.

Recommendations

In this section, we present a number of recommendations based on the results of the documentation research and interviews, where the two lead developers were asked, *"what would you do differently?"*

Create standards based on demand for data

Simply creating a specification does not guarantee adoption. A developer's role should also involve stimulating the demand for their standard. An existing product with a large following demonstrates an immediate and recognizable need for standardization. Since potential adopters will also look at other actors' behaviour in their own decision-making, having a critical mass of support is crucial for adoption.

Create simple, lightweight standards

"Do it simple in terms of the medium, avoid APIs, use CSVs, limit number of fields as much as possible to just what's really needed." (Stéphane Guidoin, Open511)

Represent's success in adoption is likely due to its simplicity. Compliance only requires release of data in a static, flat file with a limited number of required fields. On the other hand, Open511 was deemed too complex, as structuration of data in a public-facing API for real-time access was not appropriate (as most road closure data can be communicated in a static format) and requires significantly more investment from governments in order to be adopted. If a set of data should be output via an API, it is also important to think of client-server communication issues as part of the use case.

"For anyone who wants to do something serious, including civic tech – usually you will get as much data as you can on your own infrastructure and process it in a way to get a quick response time" (Stéphane Guidoin, Open511)

Users will have much quicker response times (which the respondent defined as the time required to receive data and process it to return a result) when accessing data on their local storage rather than via an API, particularly a generic open data portal API. Providing API access to data is not cheap, especially for government. With latency, bandwidth and API call limitations factored in, APIs may not have sufficient performance for those with high data and processing needs (i.e. those who need to access a lot of data at high frequency). As such, if the use case for the data involves downloading large datasets for analysis on a client machine, the data may be more suited to a static implementation, which would be reflected in its specification. These types of use cases will dictate how a data standard is written and what type of format it will support.

Those wishing to create a standard should ask themselves: what medium or format is appropriate for transmitting this data? Does the data need to be delivered in real-time, or can it be static data?

Involve the appropriate collaborators early on and sustain engagement

Institutional stakeholders, particularly governments, should not be viewed as monolithic entities. Within large institutions, there can exist specific domain experts, such as urban planners and data analysts, and IT experts, who may be responsible for interoperability of internal systems and databases. One may have domain knowledge, whereas another may have IT implementation expertise. Involving both domain experts and IT experts are therefore crucial to getting a standard adopted within an institution.

If collaborators are not engaged from the very beginning of the development process, problems discovered further down the line will be very costly and time consuming to rectify. This can result in withdrawal of support if problems are irreversible. Involving collaborators from the beginning can ensure pre-determined use cases and personas are as accurate as possible. In the case of Open 511, the limited number of collaborators restricted their ability to understand the types of use cases or personas that were needed, especially as the standards developer was not a transportation expert. Other personas or use cases that should have been developed were the perspectives of transportation departments in other cities in North America. However, resource and time limitations restricted the developer's ability to be more inclusive. Facilitators should be wary of collaborator fatigue, particularly once a standard has passed it's main development phase ('version 1.0').

Create software tools to ease the process of adoption

Data validators, conversion tools, and proofs of concept are essential for programmers who need to convert from one specification to another. Validation and conversion tools can help a data provider become compliant overnight, while proofs of concept (such as Open511's demo site) help decision-makers visualize the potential outcomes of adoption.

Seek software publisher support for open data standards

"In the end, what made GTFS work, it was the software dealers and producers that had to implement GTFS...What we really needed was a publisher, an editor of data, and the consumer." (Stéphane Guidoin, Open511)

Institutional support needs to account for more than just potential users of a standard. Governments who are reliant on large enterprise software solutions will have little incentive to adopt a standard if their existing tools do not already.

Factor in costs of outreach, facilitation, and maintenance

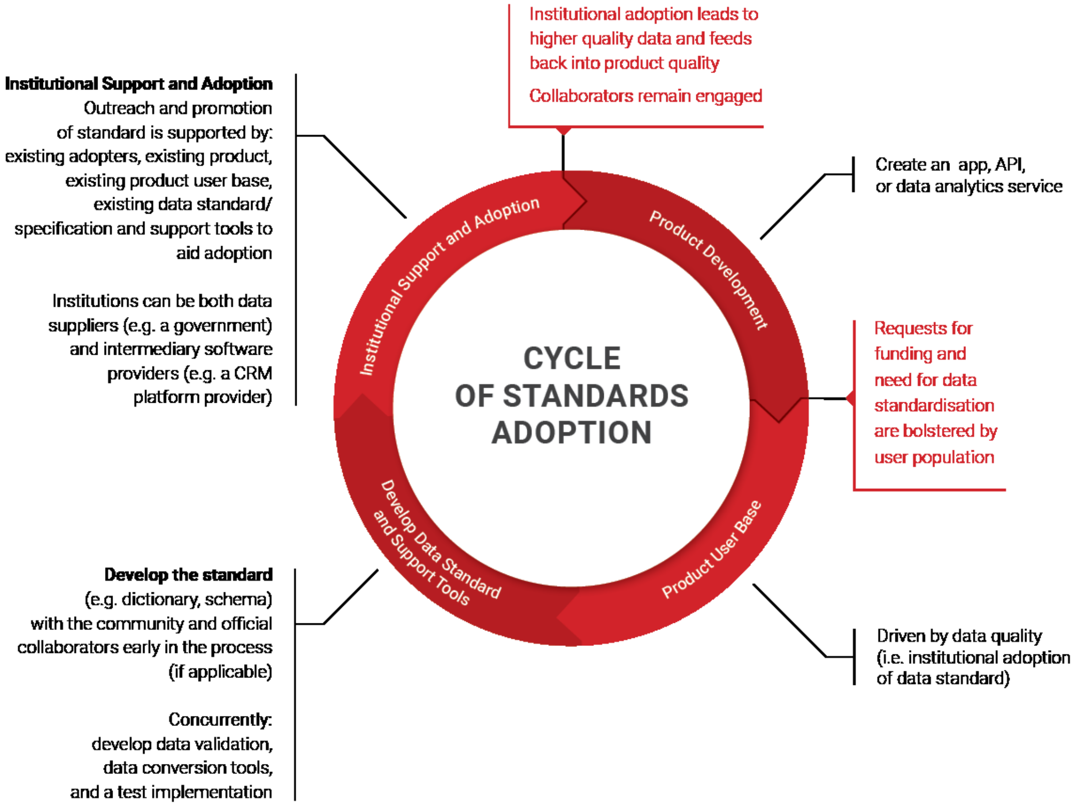
Significant time and labour required to facilitate collaboration of two major institutions in the case of Open511. This created delays in development that threatened its release schedule. Meanwhile, Popolo's development and documentation has been hindered by a lack of funding, while Represent only receives funding for maintenance of its scripts, not outreach. Because of this, we recommend attempting to factor in delays in collaboration into funding proposals.

Standards developers may need to factor in extra outreach costs to educate collaborators who are new to standards development.

A Cycle of Standards Adoption

What has emerged from the data collection is a potential positive feedback loop to adoption, categorised into four broad stages, as shown in Figure 1. Note that the ordering of stages presented here is just one approach - the development of a standard could still appear first in other scenarios.

Creating a product that generates a large user base can demonstrate to institutions (data providers) the value in complying with a standard. From there, if the right arguments are provided to institutions (e.g. existing institutional adoption rates, availability of conversion tools), adoption becomes more likely. This suggests a positive feedback loop, as compliance and support by institutions can result in higher quality, standardized data, which feeds back into the quality of the original product. Compliance itself feeds into the cycle of adoption, as it provides legitimacy (institutional support) to a standards adopter. Even very simple certification (as in the case of Represent) can help a standard gain momentum as potential adopters are able to see which other institutions have adopted before them.



Institutional adoption leads to higher quality data and feeds back into product quality

Collaborators remain engaged

Institutional Support and Adoption
Outreach and promotion of standard is supported by: existing adopters, existing product, existing product user base, existing data standard/ specification and support tools to aid adoption

Institutions can be both data suppliers (e.g. a government) and intermediary software providers (e.g. a CRM platform provider)

Create an app, API, or data analytics service

Requests for funding and need for data standardisation are bolstered by user population

Driven by data quality (i.e. institutional adoption of data standard)

Develop the standard
(e.g. dictionary, schema) with the community and official collaborators early in the process (if applicable)

Concurrently: develop data validation, data conversion tools, and a test implementation

Figure 1. A Cycle of Standards Adoption

Conclusion

In this report, we presented three standards: Open511, Popolo, and Represent. Each underwent significantly differing development processes and experienced different types and levels of collaboration and adoption. From our interviews with the key developers, we suggest a cyclical approach to standards adoption, whereby the demand for a standard is driven by product demand and existing institutional support, and the quality of the product is (at least in part) driven by adoption of the standard. We present a number of recommendations, noting that adoption is not just an activity for data providers and data users. Because data providers (governments) may use large enterprise software platforms to create or output data, support from intermediaries such as large software publishers is also crucial to spreading adoption of a given data standard.

Our three standards were developed with varying levels of collaboration, but also achieved varying levels of adoption. This calls into question the efficacy of ‘open’ in open standards. Represent is a case where collaboration was not a factor determining adoption. Ideals of openness, including collaborative governance, may be in conflict with the need for standards adoption. Standards developed strictly to a collaborative, co-creation approach may be more democratic (and may even result in a better standard), but does not guarantee a crucial requirement for adoption - consensus.

Appendix A - Interview Guide

Interview Questions

Introduction

- How do you personally define an open standard for data [open data standard]?

Initiation

Questions in this section are specifically about the actions you undertook *prior* to developing the standard.

- What was the reason behind developing the standard?
- Who was the standard developed for?
- How did you ensure this standard was sufficiently unique?
- What was the original vision for the standard, and did it change over the course of development or after adoption?
- Who and what did you need to initiate the development of the standard? What were your first steps, prior to developing the standard?
- Did you approach standard development from an: ontological perspective, user perspective?
- Were there any conflicts amongst collaborators during development? Were these anticipated or unanticipated?
- What was your model for collaboration? How did you resolve these conflicts?
- What are the key decisions that must be in place before you begin development?
- When you were designing (or first thinking of) the standard, did you also take into account: adoption, value creation by adopters, redundancies with other?

Development

Questions in this section are specifically about the actions you undertook *during* the main development phase of the standard until its initial public release.

- What were the very first actions you took when development began?
- Please describe the process of standards development
- How and when did you create dictionaries and schema?
- What resources and tools were required?
- How did you define the scope of the standard?
- Is the standard interoperable with other standards?

Maintenance and Adoption

Questions in this section are specifically about the actions you undertook once the standard was considered public

- How was the standard maintained?
- What resources are required to maintain the standard?
- How important is the user/stakeholder community in maintaining a standard?
- What actions did you take to promote adoption?
- Did you monitor adoption and usage?

- Do you consider the standard to be complete?
- Do you consider the standard a success?

Conclusion

- In hindsight, what would you have done differently in: initiation, planning, development, project management, resources?
- What should happen next for the standard to ensure sustainability and increased adoption?