ARCHIVEMATICA AS A CASE STUDY FOR SUSTAINED DIGITAL PRESERVATION

Ashley Blewer
Artefactual Systems
Canada
ablewer@artefactual.com

Sarah Romkey
Artefactual Systems
Canada
sromkey@artefactual.com
https://orcid.org/0000-0003-3833-7648

Ross Spencer
Artefactual Systems
Canada
rspencer@artefactual.com

Abstract – Archivematica is an open source software platform that incorporates the community's experiences and skills to create a robust digital preservation processing system. This paper describes Archivematica as a case study for sustained digital-preservation software. This paper briefly covers the software's history before exploring in more detail the components that make Archivematica what it is today, taking in some of the benefits and some of the current limitations of the approach. Looking to the future these limitations are also discussed, as well as other considerations about how the system, and how the software can move forward alongside the entire digital preservation community.

Keywords – Digital preservation; open source software; maintenance

Conference Topics – Designing and Delivering Sustainable Digital Preservation; Building Capacity; Capability and Community

I. HISTORY

In 2008, Peter Van Garderen had an idea for a "simple" filesystem-based digital preservation system. Archivematica has since grown into a comprehensive digital preservation system.

Archivematica's goal was introduced at the iPres 2010 conference [1], which was stated as to "reduce the cost and technical complexity of deploying a comprehensive, interoperable digital curation solution that is compliant with standards and best practices." In a few years, the software grew to encompass "system scalability, customization, digital repository interfaces, format policy implementation, and a business plan that stays true to the ideals of the free software community" [2].

Archivematica uses a microservices design pattern to provide an integrated suite of software tools that allows users to process digital objects from ingest to access in compliance with the ISO-OAIS functional model. Users monitor and control the digital preservation pipeline via a web-based dashboard. Archivematica uses METS, PREMIS, Dublin Core, the Library of Congress BagIt specification and other best practice standards and practices to provide trustworthy and interoperable archival information packages (AIPs) for storage in your preferred repository. As a product, Archivematica seeks to "not re-invent the wheel" but rather leverage existing utilities by wrapping them in a fluid digital preservation system. Archivematica is made up of services that any institution could seek to run on their

16th International Conference on Digital Preservation
iPRES 2019, Amsterdam, The Netherlands.

DOI: 10.1145/nnnnnnn.nnnnnnn

own, but one of the goals of the product is to lower the barrier to entry for digital preservation, and create a system where all of these services work in congruence with each other.

Archivematica's sustainability model was addressed in the 2012 iPRES proceedings by Peter Van Garderen and Courtney Mumma, in their paper "The Community driven Evolution of the Archivematica Project": "The free and open-source, community-driven model provides the best avenue for institutions to pool their technology budgets and to attract external funding to continue to develop core application features as requirements evolve. This means the community pays only once to have features developed, either by in-house technical staff or by third-party contractors such as Artefactual Systems" [3].

II. Dependencies

Archivematica as a software has been a work continually in progress of development for over a decade, and has grown from a small grant-funded "backend for AtoM (Access to Memory)"[1] into a robust, standards-based digital preservation system. This paper seeks to disclose the current practices for development and long-term support for Archivematica by its primary caretaker, Artefactual Systems, as a transparent case study for sustainable open source digital preservation systems.

Supporting software requires much and many different kinds of work. There are dependencies both in and around software systems. These dependencies in software form a functional system much like cogs in a machine or groups of people -- everything must function together and thusly depend on each other for success.

Software systems are written in one or more specific programming languages and sometimes a framework for those languages. The core structures of Archivematica is written in Django, which is a framework for writing web applications in Python. Just like Archivematica as a software requires regular maintenance and support, the languages and frameworks of which software is composed of also require a certain level of maintenance. Updating to the latest versions of programming languages and associated frameworks within a software system are essential maintenance tasks.

Archivematica typically runs by itself on a dedicated server or virtual machine, and between Archivematica and hardware is the Linux operating system. This operating system also requires regular system updates for maintaining the security and integrity of the platform on which Archivematica and all of its dependencies run. Falling behind on these updates can affect how the software performs in the greater computing environment, whether on the web or in a locally installed application. Software, programming languages, and operating systems all often have an "end of life" or "end of support" date for given versions, and it is important to heed those dates and perform appropriate updates. The farther along a software, language, or system develops, the harder it can be to perform an update from a software's current version to its latest version.

Archivematica as a software is particularly dependency-heavy due to its primary design philosophy. Archivematica's intention is not to reinvent any wheels, so it utilizes existing tools to perform most of its core preservation tasks (such as file identification, virus scanning, characterization, normalization, et al). Archivematica as a software intends to provide a wrapper around these preservation services and operates as a workflow engine that allows users to run preservation microservices in a logical, standards-based, and recommended sequence. Each of these core preservation tools follow their own development cycles, and in addition to the maintenance needed to

1An open source software for archival discovery.

upkeep itself, Archivematica must consider the upkeep of the tools that run within it.

These technical considerations are all part of what can be described as "maintenance," which has gained traction as a worthwhile topic of discussion and reflection in technical communities[2] All of the non-technical components that go into enhancing software (writing and documenting requirements, training, writing documentation) are also a crucial part of the maintenance and long-term sustainability of software projects and initiatives.

III. Artefactual's Role as Software Steward

As stewards of the software, Artefactual tries to guide sponsoring institutions in software development projects towards more generalizable workflows for both ease of maintenance and community benefit. Because it is not feasible to support all features forever, sometimes tough decisions have to be made. The project team may decide to depreciate a feature if it has fallen into disuse and removing it is easier than maintaining it. For example, the project team is currently depreciating the integration with Archivists' Toolkit.

One major aspect of this is Artefactual's core business model, the traditional "bounty model." This bounty model means that institutions can sponsor development of new features or fixes that they require for their workflows, and these features are then rolled into the software in a future public release. The bounty model allows institutions to asynchronously share resources, and allows smaller, less funded organisations the ability to benefit from features requested by larger, better funded organisations. The bounty model funds what features and functionality people want in Archivematica, so it leaves feature prioritization development in the client's hands. The tension of innovation versus maintenance is a prominent part of the management of funding the software and keeping it healthy and maintained.

Artefactual incorporates a fee of 10% as a "Community Support fee", added on to most development contracts specifically to support the ongoing code maintenance: activities such as resolving merge conflicts in the code, documentation and regression testing. The fee also emphasizes to development sponsors that they are part of the commitment for the maintenance of the feature Other than this maintenance fee, fundraising to pay for code maintenance or technical debt derived from the bounty model is a non-trivial task. It is difficult not just for Artefactual, but also for clients to convince their administrators to approve funding maintenance work in this context.

Artefactual serves as the stewards of the Archivematica software but the community stretches far beyond the reaches of Artefactual. Many Archivematica users have never interacted with Artefactual. This is one of the nice things about open source software, but it can be difficult to gauge community interest broadly. This creates a tricky dynamic that is difficult to balance for Artefactual -- wanting to be contributing members of the community but not be the sole drivers of the future of the software.

Recently, Artefactual has begun developing a more open and organized road map for Archivematica. The hope is to achieve better management of maintenance tasks through better planning. A roadmap can take many forms and Archivematica's remains a work in progress. A Trello board[3] is used to track enhancements and fixes which either

2See, the Maintainers Research Network: https://perma.cc/68WW-M9L8

3See, the Archivematica Roadmap: https://perma.cc/6KKT-BREV

have a milestone for an upcoming release of the software, or remain on the "wishlist". Roadmap items are intended to be a broader description than an individual issue; in agile methodology they would be referred to as "epics" [4]. The purpose of describing epics on the roadmap is to allow the user community a window into what they can expect from the software in the future at a high level, and also for them to see, and possibly contribute to gaps in functionality.

IV. Funding Software

A. The Traditional Bounty Model

As mentioned, Artefactual's primary method of developing the Archivematica software has been the bounty model -- one institution (or several in collaboration) pays for a feature which is then incorporated into the project for general use. Taking a look at a typical estimate is broken down for a very small feature or "bugfix" in Archivematica[4], the complexity of any change to the system is apparent:

TABLE I
Sample Estimate for Archivematica Development Project

	Step	Who	Estimate (Hours)
1	Analyze the requirements of the feature or the fix.	Analyst	6
2	Create a pull-request to satisfy those requirements.	Developer	12
3	Create appropriate unit tests.	Developer	6
4	Code review	Development team	2
5	Seek approval on product milestone	Product Owner	0
6	Rebase and merge with QA branch	Developer	0.5
7	Update or create new documentation	Analyst, developer and development team	6
8	Update sample data for testing	Developer	2
9	Release-candidate made.	Development team	Part of product release
10	Verification testing	Analyst team	Part of product release
11	Regression testing	Analyst team	Part of product release
12	Additional feature documentation, e.g. delivery of a screencast to users.	Analyst or project team member	4
		Total	38.5 hours

As demonstrated above, there is rarely any part of an estimate that might be

4Numbers are only representative of a small-feature or fix. As an example of what might be out of scope in these estimates might be the development of a new microservice. Numbers can also vary from case-to-case. As an example, documentation might already be largely complete for a feature for which a bug-fix is being submitted, so might not need adding to or enhancing.

characterised as a quick fix. The number of people involved in the creation of a software change is also quite high. Not represented in the tasks above include the work done by Artefactual and a client required to get to the stage of creating an estimate, internal and external project meetings along the way, and other administrative overhead associated with work. The tasks that fall under "Part of product release" are, in theory, funded by the 10% Community Support fee that Artefactual adds to each development contract.

B. An increasingly agile bounty model

A number of projects in recent years have adopted more agile approaches for Archivematica development, such as buying hours to be used toward a final goal, but without that goal being 'fixed', as in the "waterfall model" of project management alluded to in the above example. Bentley Historical Library sponsored a number of features to support their inhouse workflows and took an iterative approach to the development, allowing developers and users to collaborate on requirements as the project progressed. In another model, Wellcome Collection has improved testing in Archivematica by sponsoring development work around its automated test suite. Wellcome and Artefactual have worked together to define goals in roughly bi-weekly sprints based on what was felt was needed and could be achieved in that time with the resources available -- in contrast to setting a strict goal up-front. In the future it is hoped that organisations will continue to sponsor development in iterative ways that may contribute to sustainability of the project through better testing, dependency upgrades and the like.

C. Alternative approaches?

With the bounty approach Artefactual runs the risk of developing features most pertinent to larger, well-funded organisations and neglecting the input of smaller organisations. With the agile approach there are risks too. One such risk is that the final output might not be 'all that was envisioned at first'. That being said, one of the benefits of an agile approach is that it is better able to manage the uncertainty involved in any development process. An example of this might be having to react to develop a new programming library to deal with a type of data that was previously unanticipated. Perhaps, the implication for an organisation working with Artefactual this way is that they are just happy to push Archivematica in a forward-direction. Benefits of agile for the clients that Artefactual work with are:

- The creation of burn-down data in support of decision-making to create a greater trust around estimation in future.
- Greater freedom to affect the small changes along the way, bug-fixes, patches, documentation, release-packaging, etc.
- An incremental, but systematic approach to feature development, that without a fixed end-point, allows the feature to evolve as its use is further understood. Something that can be lost when a feature has to travel a fixed path from point inception to implementation.

D. Contributions

Receiving contributions of code, documentation and other community participation is a marker of health for any open source project. In this respect Archivematica has been less than healthy in the sense that the vast majority of code for the project has been written by employees of Artefactual Systems. Artefactual has begun to address this problem through collaborations with partners; namely, an Memorandum of Understanding (MOU) and a project sponsored by Wellcome Collection has opened the doors to code contributions through collaborative code review and increased automated test coverage for the code (thereby making it easier to accept code from an external contributor).

A common question faced by Artefactual relates to the chosen open source license for Archivematica. All Archivematica code is released under a GNU Affero General Public

License (A-GPL 3.0) and associated documentation is also released under a Creative Commons Share-alike license. The decision-making around this choice is asserted in the Archivematica Contributor Agreement, which all contributors are required to sign:

"One of the key challenges for open source software is to support a collaborative development environment while protecting the rights of contributors and users over the long-term. Unifying Archivematica copyrights through contributor agreements is the best way to protect the availability and sustainability of Archivematica over the long-term as free and open-source software. In all cases, contributors who sign the Contributor's Agreement retain full rights to use their original contributions for any other purpose outside of Archivematica, while enabling Artefactual Systems, any successor Foundation which may eventually take over responsibility for Archivematica, and the wider Archivematica community to benefit from their collaboration and contributions in this open source project."

V. MAINTENANCE

Beyond new feature development, software projects needs to be patched, fixed, upgraded, debugged and monitored. In addition to that, processes and regulations for taking these actions need to be addressed, maintained, and supported [5].

From the Maintainers conference in 2016, Nathan Ensmenger presented on the "unexpected durability of digital technologies," and found that in studies very little of maintenance goes into what we would think of as bug fixing (e.g., making sure the software works the way we expect it to). He wrote that, "The majority of software maintenance involve what are vaguely referred to in the literature as "enhancements"... This included the introduction of new functionality, as dictated by market, organisational, or legislative developments" [6].

A. Addressing technical debt

Increasingly estimates for Archivematica projects look to manage technical debt upfront -- making it as clear as possible to clients that a feature is only a small percentage of what is involved. Alluded to also is the maintenance cost built into development contracts on-top of what is already estimated.

Contributors engaging with Archivematica via the GitHub organisation are presented with a list of acceptance criteria designed to tackle technical debt. These criteria need to be satisfied to see a contribution accepted by Artefactual. The practice is followed inside Artefactual as well.

The Archivematica Contributing guidelines [7] describe coding standards that should be adopted when developing a fix or a feature which in part, aims to avoid technical debt. Artefactual could go further such as in these examples:

- Maintaining calendars of dependency end-of-life dates and building the time and financial dedication required for updates into the software release cycle
- Allowing space in the development schedule to remove "dead" code and simplify the codebase
- Paying attention when a developer says "I could make our lives so much easier if I had time to do X." Often the most vocalized needs are user facing but if an adjustment to the code makes the developers' work easier it ultimately results in a better, more maintainable product.
- Balancing new feature development with known maintenance cycles. For example, by leveraging agile rituals, a project could devote a sprint (or more) to focus on upgrades and maintenance work.

B. Accept some depreciation

Decision-making around the maintenance of software means accepting a level of depreciation over time. As mentioned previously, depreciating features is possible

and does happen. Questions around deprecation are around how to perform this thoughtfully and with appropriate consultation of the community. As an example, Artefactual has put forth to the Archivematica community the decision to depreciate Archivists Toolkit support in Archivematica, based on the perception that most Archivists Toolkit users have moved along to using ArchivesSpace or other tools.

VI. Recent developments

In 2018, Artefactual opened a single Issues repository[5] to track all known issues or bugs, whereas previously they had been spread amongst many code repositories. Artefactual invited members of the digital preservation community to join the GitHub organisation and the Issues team with the goal of allowing the larger digital preservation community to take part in the process changes too, opening up the conversation. Labels are used to provide consistency and guide movement through Artefactual's public kanban. Labels help determine the milestone (release timeline) of a project. For the community, organisations are invited to add their own label so that they may signify interest in an issue and see how that work progresses.

Within the Issues tracker, Artefactual asks all issues be described as 'problems'. This started as an internal regulation but stems from parts of the broader open-source community, to help focus the attention of the writer on the problem that needs to be solved and not just adding what they believe to be the solution. The hope is that this increases engagement amongst everyone involved or interested in a particular issue.

VII. Keeping focused

It is fair to say that the interest in improving Archivematica comes from a few different disciplines. The ability to deploy with ease might come from a systems administrator's perspective. Asking to use the most up-to-date programming libraries might make a developers life easier; but the original use-case shouldn't be forgotten while many different streams of work might happen parallel to each other.

As a demonstration, Artefactual is continuing to seek out ways to increase Archivematica's ability to scale horizontally and vertically [8]. The end result of any software optimization would mean very little to a community of digital preservationists if the outputs (system-agnostic archival information objects [AIPs]), are somehow rendered unhelpful, or worse, incorrect. The hypothesis of digital-preservation is that the AIP is the object that will be taken into the future. As such, when all the optimizations that make Archivematica as fast as possible, or as efficient as possible, are complete, then the details of the process by which the material was processed and the AIP was made still need to be reflected in the preservation metadata.. Users must still be able to trace the original-order, and order-of-activities that happened on transfer objects so that their chain of custody remains unbroken.

Tackling this in software, a collaborative, multi-company community project such as the Preservation Action Registries[6] (PAR), alongside OPF, Preservica, and JISC, represents an important mechanism for interfacing with external tools. As more preservation, or preservation adjacent systems interact with a common interface such as PAR, then there are more eyes on a consistent digital preservation community's ecosystem of tools and utilities. Through the PAR more organisations can take responsibility for maintaining the integrity of the output of digital preservation tools, their performance and their reliability, thus improving them for the benefit of a whole.

VII. Looking forward

As Archivematica's primary maintainers, Artefactual believes that the sustained way forward for the Archivematica project is through community involvement. Artefactual

5See, Archivematica's collected issues repository: https://perma.cc/TC24-PHT8

6See, the RDSS-PAR Project: https://perma.cc/A2AM-ZRTD

wants to see as many people preserving digital heritage in the most sustainable way possible; in order to achieve this end Artefactual needs to know its team is focused on the features and workflows needed the most. Meaningful community involvement also means lifting the veil on software development practices and challenges, asking for help when needed, and being transparent about what they can and cannot do.

Some ways in which Artefactual have been working to improve the practices around community development in Archivematica not included in this paper include:

- Developing better practices around release cadences e.g. by being more deliberate in managing our release scope, and looking back on what was achieved, will make releases more predictable and build trust with the user community.
- Understanding more about how users want to use Archivematica and what they want to use it for, challenging assumptions encoded in the system and understand how to rewire it to improve on those.
- Identifying and filling missing community roles. Identifying who is doing the work, how this work is being organized, and who is the advocate for maintenance -- Is it Artefactual, other companies, clients, non-clients, or end users?
- Figuring out how to talk about community and maintenance, such as what kinds of language and in what situations (public or private, at conferences or online, et al).
- Negotiating points of conflict, such as in a pull request to the codebase or architectural decisions.
- Developing better automated testing practices. Automated integration looks beyond what an individual script might do in the context of a workflow to the output of the system as a whole.
- Building 'maintainer' capabilities across a broader number of community members and companies so that aspects of a release such as code-review are not automatically a bottleneck for community submissions.
- To foster greater developer contributions Artefactual must improve the consistency of its software development practices which, in-part, will come out of testing, and developer-documentation; but it will also rely on standardizing interfaces to key parts of the system, where currently there are the many idiosyncrasies of Archivematica's many past developers to be seen.

These are themes that are appearing when Artefactual talks to clients or reviews the output of user groups and forums, such as the 'Archivematica User Forum'[7]. The notes taken in this type of forum are invaluable to identifying where opaqueness exists and where Artefactual can improve, in communicating intent, in engaging with contributions, and in directing the roadmap.

IX. Conclusion

Like the field of digital preservation, Archivematica is still young. And like digital preservation practices, Archivematica's are still evolving. As many assertions as this paper makes, it is hoped that the spirit that comes across is one of 'community'. In the presentation created to support Mumma and van Garderen (2012) [4] one of the more humorous slides might be considered as the one of 'Mr-T' making it very clear that 'WE NEED YOU'. And this hasn't changed. In 2012, Mumma and van Garderen list ways to contribute:

- Discussion: ask questions and provide opinions/suggestions on project list
- Support: answer questions on the discussion list
- Dissemination: blog, tweet, present, train
- Documentation: manual updates, wiki

7See, Archivematica User Forum, Call for Participation: https://perma.cc/GNB7-WD5Q

gardening
- Testing: report bugs, request features
- Development: fix bugs, contribute patches, write Plugins
- Maintenance: provide technical services

And seven years later, the entire digital preservation community and these methods are still very much at the core of what makes Archivematica a sustainable digital preservation project.

Archivematica has and will continue to be a resource driven by the digital preservation community. The aforementioned projects in 2018 and 2019, featuring collaborations with the organisations of JISC, Wellcome, and PIQL, will allow Archivematica to move beyond the current development bounty-model and support contract models performed primarily by Artefactual Systems to something larger, healthier, and more robust.

ACKNOWLEDGEMENTS

Thanks to everyone who currently or previously has worked at Artefactual. Hillel Arnold and Erin O'Meara for their thoughts organized during their DigiPres presentation "Acts of Maintenance". To the various organisations that have sponsored features development in Archivematica. To those who provide technical services. Finally, to those who blog, or present, or train, about using Archivematica; supporting the ecosystem through whatever resources are available to them.

REFERENCES

[1] P. Van Garderen, "Archivematica: Using Micro-services and Open-source Software to Deliver a Comprehensive Digital Curation Solution," iPRES 2010. https://perma.cc/2Z8V-RBYX
[2] C. Mumma, P. van Garderen, "Realizing the Archivematica vision: delivering a comprehensive and free OAIS implementation," iPRES 2013.
[3] C. Mumma, P. van Garderen, "The Community-Driven Evolution of the Archivematica Project," iPRES 2012. https://perma.cc/RWN9-QVU8
[4] Agile Alliance, "Epic," https://perma.cc/LM5X-S437
[5] H. Arnold, E. O'Meara, S. Romkey, "Acts of Maintenance," DigiPres 2018. https://www.slideshare.net/Archivematica/acts-of-maintenance
[6] N. Engsmenger, "When Good Software Goes Bad, The Surprising Durability of an Ephemeral Technology," The Maintainers 2016. https://perma.cc/H3R2-8FEL
[7] Artefactaul Systems, "Contributing." https://perma.cc/2TCP-6JPK
[8] N. Shalom, "What is Horizontal and Veritcal Scaling?" https://perma.cc/979P-674B