Journal of Clinical and Translational Science

www.cambridge.org/cts

Research Methods and Technology Special Communications

Cite this article: He W, Sampson R, Obeid J, Hutson K, Knosp BM, LaSalle BA, Melancon B, McGhee K, Lenert LA, and Brady K (2019) Dissemination and continuous improvement of a CTSA-based software platform, SPARCRequest©, using an open source governance model. *Journal of Clinical and Translational Science* **3**: 227–233.
doi: 10.1017/cts.2019.403

Received: 26 April 2019
Revised: 2 August 2019
Accepted: 5 August 2019
First published online: 27 August 2019

Keywords:
Open source governance; research management system; modular code; SPARCRequest©; CTSA; digital ecosystems; metric tracking; budgeting; South Carolina Clinical & Translational Research Institute; clinical research

Address for correspondence:
W. He, PhD, South Carolina Clinical & Translational Institute, Medical University of South Carolina, Suite 100, Roper Medical Office Building, Charleston, SC 29425, USA.
Email: hewwe@musc.edu

†A source of funding has been added to the financial support section. An addendum detailing this change has also been published (doi:10.1017/cts.2019.438).

© The Association for Clinical and Translational Science 2019. This is an Open Access article, distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives licence (http://creativecommons.org/licenses/by-nc-nd/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is unaltered and is properly cited. The written permission of Cambridge University Press must be obtained for commercial re-use or in order to create a derivative work.

Dissemination and continuous improvement of a CTSA-based software platform, SPARCRequest©, using an open source governance model

Wenjun He[1], Royce Sampson[2], Jihad Obeid[3], Kyle Hutson[1], Boyd M. Knosp[4], Bernard A. LaSalle[5], Brian Melancon[6], Kimberly McGhee[1], Leslie A. Lenert[3] and Kathleen Brady[2,†]

[1]Department of Medicine, South Carolina Clinical & Translational Research Institute, Medical University of South Carolina, Charleston, SC, USA; [2]Department of Psychiatry and Behavioral Sciences, Medical University of South Carolina, Charleston, SC, USA; [3]Biomedical Informatics Center, Medical University of South Carolina, Charleston, SC, USA; [4]Roy J. and Lucille A. Carver College of Medicine and the Institute for Clinical and Translational Science, University of Iowa, Iowa City, IA, USA; [5]Department of Biomedical Informatics, Center for Clinical and Translational Science, University of Utah, Salt Lake City, UT, USA and [6]Pennington Biomedical Research Center, Louisiana Clinical and Translational Science Center, Baton Rouge, LA, USA

Abstract

SPARCRequest© (Services, Pricing, & Application for Research Centers) is a web-based research management system that provides a modular and adaptable "electronic storefront" for research-related services. Developed by the South Carolina Clinical & Translational Research Institute at the Medical University of South Carolina, it was released as open source (OS) code in 2014. The adoption of SPARCRequest© accelerated in 2016, when, to ensure responsiveness to the needs of partners, its governance also became open. This governance model enables OS partners to suggest and prioritize features for new releases. As a result, the software code has become more modularized and can be easily customized to meet the diverse needs of adopting hubs. This article describes innovative aspects of the OS governance model, including a multi-institutional committee structure to set strategic vision, make operational decisions, and develop technical solutions; a virtual roadmap that ensures transparency and aligns adopters with release-based goals; and a business process model that provides a robust voting mechanism for prioritizing new features while also enabling fast-paced bug fixes. OS software evolves best in open governance environments. OS governance has made SPARCRequest© more responsive to user needs, attracted more adopters, and increased the proportion of code contributed by adopters.

Introduction

SPARCRequest© (Services, Pricing, & Application for Research Centers) is a web-based research management system created by the South Carolina Clinical & Translational Research (SCTR) Institute, a Clinical and Translational Science Award (CTSA) hub at the Medical University of South Carolina (MUSC) [1,2]. Since 2012, SPARCRequest© has offered a one-stop-shop for research-related services and resources, coupled with an online ordering and fulfillment tracking system. It began as a grassroots effort at MUSC to provide investigators with a centralized portal through which they could access research support services and pricing. This portal has facilitated clinical and translational scientific proposals and budget development, streamlining research development and service center management. Features added during iterative development cycles have removed barriers to study activation; reduced the administrative burden on service centers and core facilities; and supported budgeting, billing compliance, metric tracking, evaluation, and reporting. From 2014 to 2018, the number of active SPARCRequest© users at MUSC (users who logged in at least once during that year) increased from 721 to 1152, and the total number of research studies and projects created increased from 3831 to 8246.

SCTR team leaders recognized the potential of SPARCRequest© to benefit study teams, grant administrators, clinical service providers, and evaluation teams across the MUSC enterprise and the clinical and translational science community. Sharing a common, but customizable, research management platform would enable rapid integration of new features across the CTSA and Institutional Development Award Clinical and Translational Research (IDeA-CTR) networks. It would also enable institutions to tailor the source code to meet local needs and preferences. In short, it would help them build a robust and sustainable biomedical informatics infrastructure that is crucial for translational research [3].

To realize this promise, SCTR opted in 2014 to make SPARCRequest© an open source platform (SPARC OS), sharing its code publicly through a BSD3 license (https://bit.ly/2qGjhgD), which offers almost complete freedom to use, modify, and redistribute the software, as long as the original copyright and license are acknowledged. Open source (OS) initiatives rely on a community of volunteer developers to test and improve the code [4,5], and the most successful achieve sustainability by retaining the interest and loyalty of those developers.

The SPARCRequest© team invited developers at partner institutions in the CTSA network and beyond to contribute to the code, address bug fixes, and help develop new features. SCTR remained the primary decision maker as to which new features would be included in each SPARCRequest© release, sending out documents and notifications to adopters during monthly conference calls. This type of "benevolent dictator" model worked for the initial phase of the OS project.

As the project progressed, however, adoption issues, variable leadership involvement, and funding challenges at adoption sites [5] began to put its sustainability at risk. As more institutions joined and became dependent on SPARCRequest© for their infrastructure, adopters requested a more egalitarian and flexible approach to confronting these challenges. A streamlined decision-making process was needed to enable customization of the platform while avoiding major refactoring of the code with each new release.

In response, SCTR collaborated with CTSA hubs at the University of Iowa and the University of Utah to establish a multi-institutional OS governance model that provides partner institutions a voice in the development and prioritization of a SPARCRequest© roadmap for new and updated features. They did so with funding from an administrative supplement (UL1 TR001450-03S) to MUSC's CTSA grant from the National Center for Advancing Translational Science (NCATS) at the National Institutes of Health (NIH). Under the new model, a business process plan permits extended and inclusive discussion from the community about which new features to implement, while also offering an expedited route for bug fixes.

We adopted the OS governance model in an effort to make SPARCRequest© adaptable to partner institutions' digital ecosystems. We achieved our aim by making changes to the code – specifically increasing its configurability and modularity – in response to feedback from the governance subcommittees. A robust interface now enables institutions to use applications of choice (e.g., REDCap or a clinical trial management system) for tracking and fulfillment of services or to use the SPARCRequest© fulfillment module, SPARCFulfillment. Other systems (e.g., finance, awards, Electronic Institutional Review Board) can also be interfaced to streamline study activation and reporting.

Lessons learned include the need for a readiness assessment before implementation and the importance of continuous institutional commitment and user group alignment. New SPARC OS partners are now asked to complete a readiness checklist to ensure they have the proper resources in place to benefit fully from the partnership. A virtual roadmap helps ensure alignment by making strategic goals and progress toward attainment of those goals transparent to all partners. Regular committee meetings, annual retreats, and a wiki ensure robust communication among all stakeholders.

We outline here our experience with establishing an OS governance structure to support widespread dissemination and adoption of SPARCRequest© by the clinical and translational science community. The OS journey of SPARCRequest© and its transparent, responsive governance structure can serve as a sustainable model for other CTSA hubs and IDeA-CTR institutions with promising electronic tools that they would like to share with the broader research community.

Methodology

OS Governance Structure

In April 2016, with funding from an NCATS administrative supplement, SCTR collaborated with the CTSA hubs at the University of Iowa and the University of Utah to form an OS governance structure that included three committees.

A Steering Subcommittee, led by MUSC, comprises product owners and informatics leaders from adopting SPARCRequest© hubs who meet monthly to review the product roadmap and prioritize the proposed development themes. Members of the Steering Subcommittee discuss and align institutional priorities, ensure that required resources are in place, and assign development tasks.

An Operations Subcommittee, led by the University of Iowa, is made up of research operation managers, who serve as the user representatives for their institution and specify the desired functionalities. Members of this subcommittee vote weekly to prioritize all proposed new functionalities and meet monthly to discuss institutional adoption barriers and needs. In addition, when undesirable, duplicative, or conflicting functionalities are proposed, subcommittee members can question them by commenting on the feature specifications and starting a discussion around the topic before the next meeting.

A Technical Subcommittee, led by the University of Utah, is composed of development managers, who meet weekly to discuss technical approaches for new functionality and to resolve issues. Between meetings, subcommittee members use real-time communication channels to resolve issues and avoid any development conflicts, such as code branch overlap.

Business Process Model

Most institutions adopting SPARCRequest© have their own development teams that vary in size, methodology, and time they can dedicate to the project. To ensure effective and efficient cross-institutional collaboration and code contribution, the SPARC OS initiative established standard business processes and technology standards. We adopted an agile methodology that relies on self-organizing, cross-functional teams, iterative development, and frequent releases. The agile methodology is designed to respond more nimbly to changing user needs than a waterfall-based (i.e., top-down) development process [6,7]. Specifically, we implemented Scrum [8], an agile framework for managing software development that divides work into actions that can be completed in two- to four-week sprints. Teams of three to nine members hold 15-minute "stand-up" meetings, known as daily scrums, to discuss progress and adjust planning as needed. Using the Scrum framework, we established a business process model to implement the SPARC OS governance framework, as shown in Fig. 1.

Pivotal Tracker [9], an agile project management tool, is used to specify and prioritize tasks and for quality control. Development tasks are categorized primarily as new features or bugs. Each feature or bug request is entered as a "story" that describes the desired functionality from the user's perspective, with mockup or related screenshots attached. The Pivotal Tracker storyboard has a space to support the weekly cross-institutional vote on new feature requests. Each developer has a dedicated workspace on the Slack platform [10], a collaboration hub for work.

https://doi.org/10.1017/cts.2019.403 Published online by Cambridge University Press

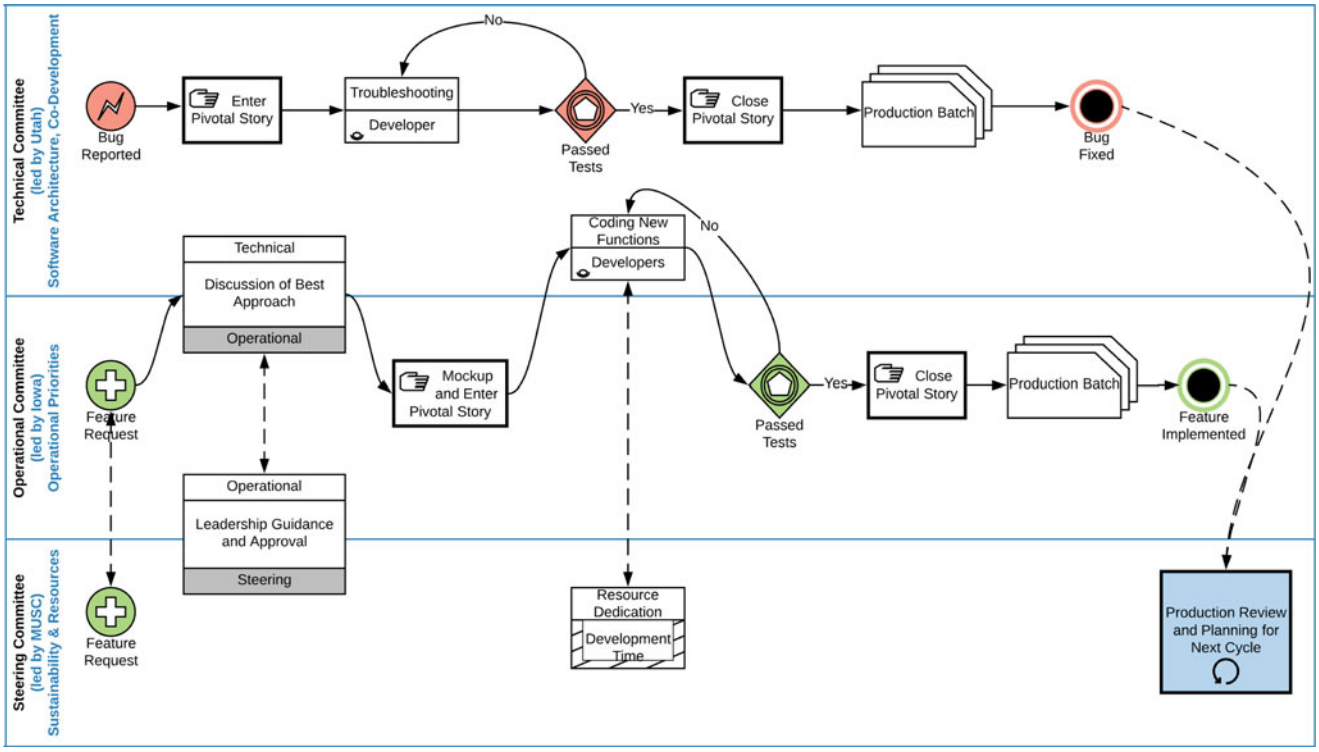


Fig. 1. The SPARC OS business process. Iowa, University of Iowa; MUSC, Medical University of South Carolina; Utah, University of Utah; SPARC OS, SPARCRequest© (Services, Pricing, & Application for Research Centers) open source.

GitHub [11], a software development platform, and Git-related tools are used for code contribution and code base management. SPARCRequest© source code is public and downloadable to any GitHub user under the BSD3 license. The public can also contribute code in the form of pull requests. A streamlined contribution guide [12] is provided for the community to follow. All Technical Subcommittee members serve as code reviewers.

Each of the subcommittees plays a defined and inter-related role in this business process. Typically, the Steering and Operations Subcommittees initiate the feature requests, the Technical and Operations Subcommittees discuss the best technical approach for implementing the feature, and the Steering and Operations Subcommittees vote to prioritize the order of feature development. The new feature request then enters the development cycle of coding, testing, and production release. Bugs found during development or between releases can be reported by the Technical Subcommittee and go directly into the development cycle for a fast fix-and-patch. This model ensures that new features are thoroughly discussed and prioritized by the SPARC OS community and that urgent fixes are addressed promptly.

Partner Alignment

We instituted the SPARC OS governance model because simply sharing source code is not sufficient to sustain an OS initiative. If the initiative was to be successful, we knew that partner institutions with diverse cultures would have to learn how to work together and that consensus would have to be built about the role of each of the subcommittees.

Early meetings of the subcommittees, conducted virtually, were not always effective, as each institution had its own way of running meetings and managing projects. Designating a consistent project manager to run those meetings brought a sense of stability and was an essential first step to improving their effectiveness. Annual retreats provided a forum for discussion about the next year's activities and an opportunity to work together face-to-face to build collaborative projects and grants, as well as trust.

Over time, the community worked together to better define the scope of the subcommittees. For example, the Steering Subcommittee became less involved in the day-to-day details of production, maintaining a focus instead on high-level goals and strategies. The Technical Subcommittee was better able to focus its efforts once a process for prioritizing features was instituted.

Also key to building alignment among the members of the SPARC OS community was a virtual roadmap (https://bit.ly/2G0Hp20), created using ProductPlan [13]. The roadmap shows progress toward agreed-upon functional goals and development themes and can be accessed and updated in real time. This transparency helps build confidence in SPARC OS community members that they know where the project is headed and the progress it is making toward its agreed-upon goals.

A number of electronic communication tools enable robust communication among SPARC OS members to continue between meetings. An instant messaging tool provided by the Slack platform [10] enables direct communication among developers. The SPARCRequest© wiki (https://bit.ly/2U5MN95) can be updated by any member of the SPARC OS community in real time.

Results

Expanded OS Partnership

Within 15 months of the implementation of OS governance, two new CTSA hubs (Case Western Reserve University and

https://doi.org/10.1017/cts.2019.403 Published online by Cambridge University Press

Table 1. *SPARC OS partners by year*

Year	No. of SPARC OS partners (total)	New partners	Partner type
2014	2	Medical University of South Carolina	CTSA hub
		Children's National Health System	Non-CTSA
2015	5	The University of Iowa	CTSA Hub
		The University of Utah	CTSA hub
		The Louisiana Clinical and Translational Science Center	IDeA CTR hub
2016	8	University of California, San Francisco	CTSA hub
		University of Missouri Health System	CTSA hub
		University of California, Los Angeles	CTSA hub
2017	9	The University of Nebraska	IDeA CTR hub
2018	12	University of California, Irvine	CTSA hub
		Case Western Reserve University (cloud instance)	CTSA hub
		Greenville Health System (cloud instance)	CTSA affiliate site

CTR, Clinical and Translational Research; CTSA, Clinical and Translational Science Awards; IDeA, Institutional Development Award; SPARC OS, SPARCRequest© (Services, Pricing, & Application for Research Centers) open source

University of California, Irvine) and one CTSA-affiliated institution (Greenville Health System) joined the SPARC OS community. In all, twelve partners (Table 1), including eight CTSA hubs, and their affiliates have adopted the software, for a total of 27 institutions. Fig. 2 shows the geographic distribution of the 12 SPARC OS partners.

Increased Configurability and OS Code Contribution

During the 2016 annual OS retreat, partners identified the biggest difficulties they encountered during SPARCRequest© implementation. These included attributes of the code that were specific to MUSC's digital microclimate, such as MUSC-specific labels and hard-coded configurations. Although partners could customize the source code at their local instance to fit their own needs, they would have to recreate those customizations with each new release.

To remedy this, SPARCRequest© developers at MUSC, Utah, Iowa, the Louisiana Clinical and Translational Science Center (LA CATs), and other partner institutions collaborated to increase the modularity and configurability of the code, so that adopting institutions could turn on or off groups of functionalities as desired. Developers moved settings, which had once been hard coded, to the database, where they would not be overwritten with each new iteration of the source code. Moving the settings from the code kernel, which updated with each release, to the database allowed the code to become substantially more configurable.

Configurable code provided users the opportunity to opt in or out of features. This increased configurability enabled MUSC to invite its OS partners to create functionalities and modules to meet their individual needs. Because these features could be "switched off," they did not oblige any other institution to implement them. This modularity motivated partners to increase their contributions to the code design, since they could develop a module that would not only meet their own needs but also potentially create value for their OS collaborators. As shown in Fig. 3, code contributions by OS partners, which were negligible before the establishment of OS governance, increased to approximately 11% in the latest two SPARCRequest releases© (v3.4.0 and v3.5.0).

Standardization of Modularity

Increased code configurability has enabled SPARC OS partners to contribute not only new features but also entire modules to meet their own and the community's needs. Even before the institution of OS governance, adopters had attempted to contribute modules. For example, the University of Iowa wrote "Additional Details" code to help service providers collect more information on individual service(s) in SPARCRequest© using AngularJS and other frameworks. However, at that time no formal structure enabled discussion of technical aspects of the module by development teams and stakeholders across the SPARC OS community. As a consequence, the module could not be adopted into the main branch of source code.

Learning lessons from the past, the SPARC OS community collaborated to develop standardized requirements and guidelines that aligned contributor efforts. Novel features of the "Additional Details" module were conserved and re-coded with expanded functionalities in 2017 to become the current **SPARCForms** module (https://bit.ly/2G3MjuR), which is continuously being improved.

In 2018, LA CATs, an NIH-funded IDeA-CTR hub, developed a **SPARCFunding** module (https://bit.ly/2FZaNFB) to support CTSA and IDEA-CTR pilot project program administrators. This module enables interested institutions to keep all center-related services, funding opportunities, and metric tracking within SPARCRequest©. The landing page of the module lists all existing funding opportunities from defined funding organizations, displayed by letter of intent and full application, and provides a link to a "Download Documents" page, where administrators can view and/or download application or pre-application documents submitted by researchers.

By encouraging SPARC OS partner engagement and code contribution, the SPARC OS governance structure improved the quality and robustness of the software, benefitting all SPARCRequest© users. At MUSC, the overall general user satisfaction rate increased from 87.8% in 2016 to 91.1% in 2018, according to data obtained from a system satisfaction survey that is triggered when a request is submitted to SPARC.

Readiness Checklist

Another key to the success of the SPARC OS initiative has been ensuring that potential partners have resources in place to fully benefit from participation. Change management literature [14] suggests that innovative projects are far more likely to succeed and less likely to lead to conflict or dissatisfaction when organizational readiness for change has been assessed before the project begins. Accordingly, SPARCRequest© developed a streamlined readiness checklist (https://bit.ly/2Uo2LjM) based on an existing framework [15] and a template Toolkit for Electronic Health Record adoption [16]. Potential partners are asked to complete the checklist to gauge whether they are financially and operationally prepared to participate in an OS initiative or whether they should instead focus on building the foundation for future participation. For example, the

https://doi.org/10.1017/cts.2019.403 Published online by Cambridge University Press

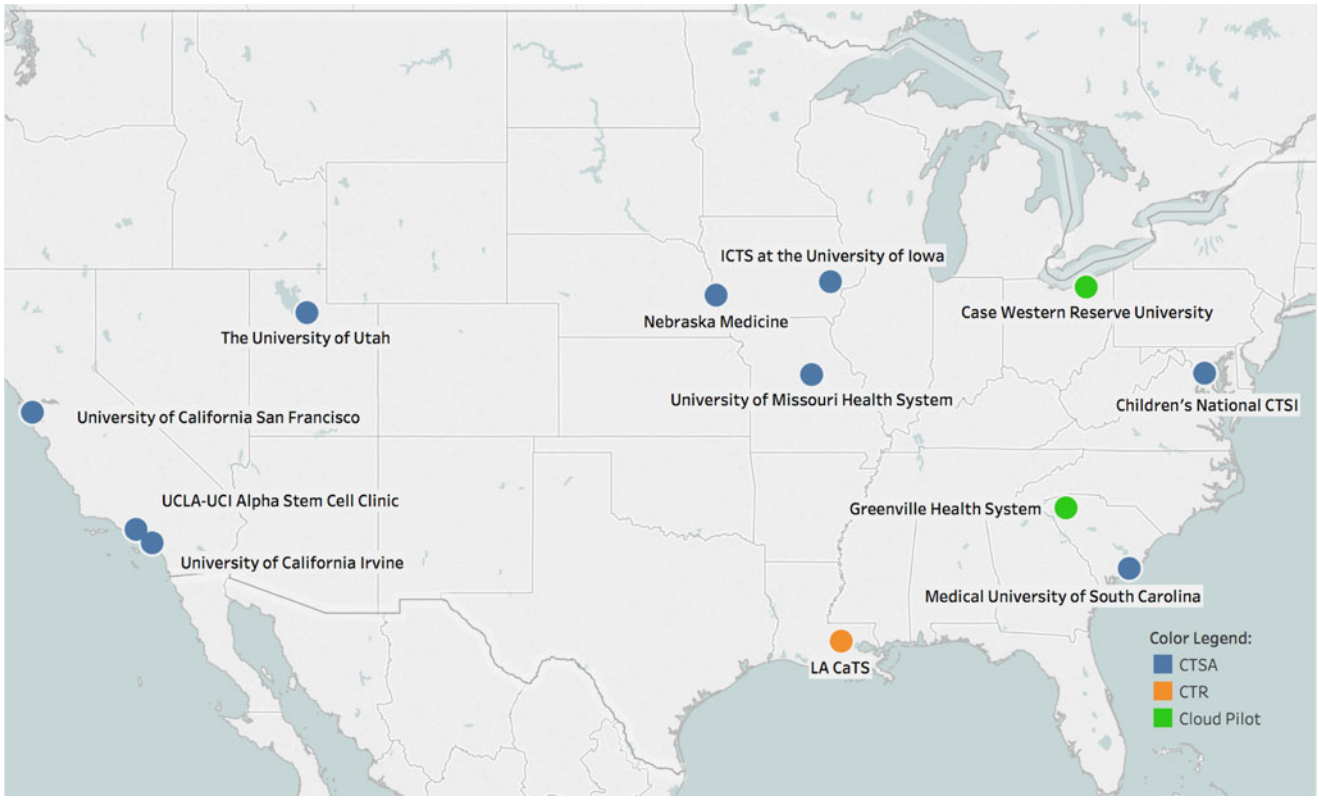


Fig. 2. Geographical distribution of current SPARC open source partners. CTSI, Clinical and Translational Science Institute; ICTS, Institute for Clinical & Translational Science; LA CATS, Louisiana Clinical and Translational Science Center; MUSC, Medical University of South Carolina; UCIrvine, University of California, Irvine; UCLA, University of California, Los Angeles; UCSF, University of California, San Francisco.



Fig. 3. Code contributions by open source partners to SPARCRequest© as of April 2019, by percentage of Pivotal Tracker stories and GitHub pull requests. SPARCRequest©, Services, Pricing, & Application for Research Centers.

checklist asks if potential partners have performed the cost-benefit and return-on-investment analyses needed to secure institutional and leadership buy-in and if they have the project management tools, workflows, and training in place to ensure that they can integrate seamlessly into the release cycle.

Discussion

By adopting a responsive governance model, the SPARC OS community has positioned itself for long-term sustainability by increasing engagement from community partners, which is a hallmark of success for OS initiatives [17]. Unlike more bureaucratic, top-down software development, where developers are assigned tasks, developers on OS projects are instead allowed to choose the tasks that suit their talents, needs, and interests. Any partner can enter requests for new features or bug fixes, and these requests are prioritized by members of the SPARC OS community within the production tool. Developers work on the highest-priority items, as determined by the community.

Also contributing to the long-term sustainability of SPARCRequest© are the modularity and configurability of its code,

https://doi.org/10.1017/cts.2019.403 Published online by Cambridge University Press

both of which increased dramatically after adoption of OS governance. Indeed, the organized but decentralized governance and specified workflow associated with OS initiatives tend to lead to more robust, modular, and configurable software platforms [18-21]. The increased configurability and modularity seen in SPARCRequest© code after institution of OS governance confirm a tenet advanced by Conway [22] and others [23-25]that a product mirrors the (governance) structure of the organization that produced it. In essence, more loosely coupled organizations [26] create more "loosely coupled" or modular code. Because it is only loosely coupled to the central code kernel, modular code enables developers to customize a tool or make relevant changes to a needed function without learning the code in its entirety or affecting the product as a whole and without concern that their modifications will be lost with each new release. Modular code is also thought to be more flexible and adaptable to change than monolithic code, which is harder to kill, maintain, or augment as design needs change [27]. Baldwin and Clark have argued that modularity and configurability attract and help maintain a community of developers by enabling them to exchange valuable work, an opportunity not available in a more monolithic system [19,28]. The ability to make meaningful contributions to the code to meet their own and others' needs has further increased engagement by the SPARC OS community [9,29,30].

SPARCRequest© has also positioned itself for sustainability by ensuring that it can adapt nimbly as user needs change. This can be difficult for milestone-driven software development, which assumes planned progress toward a defined goal, such as addressing a given user need. If that need should change, it can wreak havoc with such a development project's production timelines. By adopting agile methodologies [6,7], which emphasize continuous (re)design, iterative modifications, and frequent releases ("release early, release often"), SPARCRequest© ensures continuous improvement of the code while maintaining responsiveness to change. If the project's goals shift, it is easier to back out from bite-size additions than to overhaul a monolithic system.

In summary, adoption of an OS governance model has enabled SPARCRequest© to become more sustainable by increasing the engagement of its OS community, increasing the configurability and modularity of its code, and making the platform more responsive to change. These modifications help ensure that SPARCRequest© retains an engaged community of developers and stakeholders long after funding from the administrative supplement ends. Because each institution is managing its own code while participating in the overall governance structure, programming resources and alignment are sufficient to ensure the product's sustainability.

Although participation in the SPARCRequest OS consortium has been stable, potential partners have sometimes dropped out due to lack of technical bandwidth, management changes, funding difficulties, or a combination of these reasons. To address these challenges, the SPARCRequest© team not only created a checklist to assess partner readiness but also developed a cloud instance, which is currently being piloted at Case Western University and Greenville Health System. Partners must still meet all the operational and financial requirements laid out by the readiness checklist, but they are relieved of some of the technical burden, since MUSC manages the cloud instance. Initial experiences with the cloud instance of SPARCRequest© have been positive. Therefore, at the recommendation of adopting hubs, we are developing a cloud hosting business plan that will provide a financially sustainable model for the application as well as support SPARCRequest© adoption by interested hubs with limited informatics resources.

Conclusion

The OS journey of SPARCRequest© and its transparent, responsive governance structure can serve as a model for other institutions with promising electronic tools that they would like to share with the broader research community. Its multi-institutional OS governance and subcommittee model provides partner institutions a voice in setting priorities for feature updates in new releases. That model has encouraged input from the SPARC OS community for each release and optimized the application release cycle, with increased user satisfaction. While SPARCRequest© had always been designed to support dissemination, it was developed in a single institution's digital "microclimate," which initially limited adoption by other institutions. As a result of adopting OS governance, SPARCRequest© is now much more adaptable to partner institutions' digital ecosystems. Its code has become at once more modular and more configurable, encouraging engagement and code contribution by the SPARC OS community and helping to secure the long-term sustainability of the initiative. OS partners can now easily customize the platform to the needs and preferences of their own institutions while also making contributions that benefit the larger community. Having achieved strong partner alignment, the SPARC OS community is continuing to evolve SPARCRequest© functionalities to better meet the needs of CTSAs and IDeA-CTRs nationwide.

Acknowledgements. The authors on this manuscript would like to express their appreciation to Leila Forney, Andrew Cates, and all members of the development teams for their contribution to this project and to Tammy Loucks for her editorial feedback on this manuscript.

Financial Support. This publication was supported by the National Center for Advancing Translational Sciences of the National Institutes of Health, under Grant Number UL1 TR001450 (K.B.) and UL1 TR001450-03S1 (K.B.). This publication was also supported in part by U54 GM104940 from the National Institute of General Medical Sciences of the National Institutes of Health, which funds the Louisiana Clinical and Translational Science Center.

Disclosures. The authors have no conflicts of interest to disclose.

References

1. **Sampson RR**, ***et al.*** SPARC: a multi-institutional integrated web based research management system. AMIA Joint Summits on Translational Science proceedings 2013; 2013: 230. eCollection 2013.
2. **Glenn JL, Sampson RR.** Developing an institution-wide web-based research request and preliminary budget development system. *Research Management Review* 2011; **18**(2): 39–57.
3. **Obeid JS**, ***et al***. Sustainability considerations for clinical and translational research informatics infrastructure. *Journal of Clinical and Translational Science* 2018; **2**(5): 267–275. doi:10.1017/cts.2018.332
4. **Ibrahim H, Warner B.** Understanding the open source development model. Linux Foundation; 2011. Retrieved from http://www.ibrahimatlinux.com/uploads/6/3/9/7/6397792/00.pdf. Accessed April 23, 2019.
5. **Raymond ES.** The cathedral and the bazaar. *First Monday* 1998; **3**(2). doi:10.5210/fm.v3i2.578
6. **Dingsøyr T**, ***et al.*** A decade of agile methodologies: towards explaining agile software development. *Journal of Systems and Software* 2012; **85**(6): 1213–1221.
7. **Nerur S, *et al.*** Towards an understanding of the conceptual underpinnings of agile development methodologies. In: Dingsøyr T, Dyba T, Moe NB, eds. *Agile Software Development: Current Research and Future Directions*. Berlin, Heidelberg, Germany: Springer-Verlag; 2010. 15–29.
8. **Schwaber K, Sutherland J.** *The scrum guide*; 2018. Retrieved from https://www.scrumguides.org/scrum-guide.html. Accessed April 21, 2019.

https://doi.org/10.1017/cts.2019.403 Published online by Cambridge University Press

9. **PivotalTracker Enterprise**. *Pivotal tracker*; 2019. Retrieved from https://www.pivotaltracker.com. Accessed April 21, 2019.
10. **Slack Technologies**. *Slack*; 2019. Retrieved from https://slack.com/. Accessed April 21, 2019.
11. **GitHub**. *GitHub*; 2019. Retrieved from https://github.com. Accessed January 7, 2019.
12. **SPARCRequest Team**. *Contributing to SPARCRequest*; 2017. Retrieved from https://github.com/sparc-request/sparc-request/blob/v3.6.0/CONTRIBUTION.md. Accessed April 21, 2019.
13. **ProductPlan**. *About ProductPlan*; 2019. Retrieved from https://www.productplan.com/about/. Accessed January 29, 2019.
14. **Paré G, *et al.*** Clinicians' perceptions of organizational readiness for change in the context of clinical information system projects: insights from two cross-sectional surveys. *Implementation Science* 2011; **6**(15).
15. **Schulte M**, DBA, FACHE, CPHIMS, eds. *Go-Live: Smart Strategies from Davies Award-Winning EHR Implementations*. Chicago, IL: HIMSS; 2011.
16. **Healthcare Information and Management Systems Society (HIMSS)**. *FAFP EHR readiness assessment tool*; 2011. Retrieved from https://www.himss.org/fafp-ehr-readiness-assessment-tool. Accessed July 9, 2019.
17. **Ehls D.** Open source project collapse - sources and patterns of failure. In: Proceedings of the 50th Hawaii International Conference on System Sciences. Hilton Waikoloa Village, Hawaii: IEEE; 2017. 5327–5336.
18. **O'Reilly T.** Lessons from open-source software development. *Communications of the ACM* 1999; **42**(4): 32–37.
19. **Sullivan KJ, *et al.*** The structure and value of modularity in software design. *ACM SIGSOFT Software Engineering Notes* 2001; **26**(5): 99.
20. **Haff G.** *How Open Source Ate Software: Understand the Open Source Movement and So Much More*. New York, NY: Apress, 2018.
21. **Sorkun MF, Furlan A.** Product and organizational modularity: a contingent view of the mirroring hypothesis. *European Management Review* 2016; **14**(2): 205–224.
22. **Conway ME.** How do committees invent? *Datamation* 1968; **14**(4): 28–31.
23. **MacCormack A, Baldwin C, Rusnak J.** Exploring the duality between product and organizational architectures: a test of the "mirroring" hypothesis. *Research Policy* 2012; **41**(8): 1309–1324.
24. **Henderson RM, Clark KB.** Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. *Administrative Science Quarterly* 1990; **35**(1): 9.
25. **Peng G, Mu J.** Do modular products lead to modular organisations? Evidence from open source software development. *International Journal of Production Research* 2018; **56**(20): 6719–6733.
26. **Weick KE.** Educational organizations as loosely coupled systems. *Administrative Science Quarterly* 1976; **21**(1): 1.
27. **Bonaccorsi A, Rossi C.** Why open source software can succeed. *Research Policy* 2003; **32**(7): 1243–1258.
28. **Baldwin CY, Clark KB.** The architecture of participation: does code architecture mitigate free riding in the open source development model? *Management Science* 2006; **52**(7): 1116–1127.
29. **Lakhani K, Wolf RG.** Why hackers do what they do: understanding motivation and effort in free/open source software projects. *SSRN Electronic Journal* 2003. doi:10.2139/ssrn.443040
30. **Hars A, Ou S.** Working for free? Motivations for participating in open-source projects. *International Journal of Electronic Commerce* 2002; **6**(3): 25–39.

https://doi.org/10.1017/cts.2019.403 Published online by Cambridge University Press