DOECode: Reinventing the OSTI Energy Science and Technology Software Center

Jay Jay Billings

20160715

OSTI's role is a unique one: We are charged with advancing science and sustaining technological development by making R&D results available to both the Department of Energy (DOE) Complex and the general public. As part of this mission, we have always maintained that a critical component of the Scientific and Technical Information (STI) that we track is software. This is even more evident today as software continues to invade every aspect of modern scientific research, whether it be as part of the largest scientific instruments in the world, through high-performance modeling and simulation, driving big data analytics, or just supporting scientists in their day to day tasks. In light of this, we are very happy to announce our new initiative to reinvent OSTI's Energy Science and Technology Software Center.

What is the Energy Science and Technology Software Center?

The Energy Science and Technology Software Center (ESTSC) is the centralized software management facility for the DOE. It licenses and distributes scientific and technical software packages developed by DOE contractors, national laboratories and other facilities. The collection also contains selected software from the Nuclear Energy Agency of the Organization for Economic Cooperation and Development. In fact, it contains over 3500 distinct pieces of software. The Center was officially started at OSTI on October 1st 1991, but before that it was housed at Argonne National Laboratory where it was known as the National Energy Software Center. The National Energy Software Center, originally called the Argonne Code Center, was started by Margaret Butler in 1960, in part to support the American Nuclear Society and the development of new reactor physics codes, [1].

The original logo for the ESTSC.

The Center has grown an changed in many ways over the past fifty six years, all to better meet the needs of scientists. In keeping with that grand tradition, we have identified a few things that we feel we can do better, and we have also identified a lot of new types of software-related information that we can collect and preserve in addition to executable code and documentation.

Introducing *DOE Code*, the new ESTSC

We asked ourselves "What would a modern ESTSC look like?" That is, if we were rebuilding, from scratch, a centralized software management facility for the DOE, what would be the big things that we would take into account? There are many things to consider when answering that question, but perhaps the biggest thing that has happened in the software world since the original development of the ESTSC is the rise of Open Source Software, (OSS).

For those who are not familiar with it, OSS is is software that can be freely accessed, used, changed, and shared (in modified or unmodified form) by anyone, according to the Open Source Initiative, [2]. There are other definitions of open source software too, such as “Free Software,” but essentially the idea is that anyone can access the source code and modify under some set of terms. The truly amazing thing is that OSS has taken the world by storm. The 2016 Future of Open Source Survey, [3], examined the roles of open source software in 1313 companies from sixty four countries. The survey revealed that nearly 65% of companies use OSS in their development and nearly 55% use it in production infrastructure! Likewise, nearly 65% of companies contribute to, encourage developers to engage in, and provide features for OSS.

Software development is also a very social activity in the 21st century. Not only do communities share development activities in collaborative sites like GitHub, [4], or through software foundations like the Eclipse Foundation, [5], but they organize large gatherings, called Hackathons, where possibly hundreds of developers will gather in the same room to add functionality by “hacking on the code.” A modern ESTSC would embrace these new social aspects in every element of the site, connecting people with projects and linking information about software across various platforms in novel ways. In addition to just linking information, it would also offer enable certain social media aspects that users have come to expect from almost any website.

Another major thing that we want to introduce in the reinvention of the ESTSC is its branding. While the name Energy Science and Technology Software Center is quite descriptive and understandable, the acronym ESTSC does little more than roll off the tongue. It also does not acknowledge the Department, speak to software developers, or capture that some DOE code is neither software nor related to energy science. We decided to rename the Center to simply *DOE Code* and to develop a new set of logos to go with the new name.

The new logo for DOECode.

Software developers will immediately recognize the significance of this logo. For those who do not speak code, this logo acknowledges and celebrates the fact that the DOE has long been a leader in the development of scientific software. It also acknowledges that the DOE depends on code and could not run without it by depicting the an object-oriented programming concept called a constructor, which shows that the DOE requires Code to operate.

A New Focus

We know anecdotally that there are many software projects in the DOE Complex that are not tracked by OSTI. Some of these packages have not yet been “official released” but are nonetheless available on open source sites like GitHub, Bitbucket, etc. Other packages that contain significant contributions by DOE scientists might be publicly available, but were never reported because the copyright is held by a university, company or some other organization. There are many such reasons why OSTI might not know about a particular piece of software, but whatever the reason may be OSTI still has the job of archiving and disseminating all DOE STI.

So it is clear that DOE Code needs to have a major focus on source code. There are two options for how to do this: Mandate that everyone put their repositories at OSTI or *aggregate project metadata and archives* of source code repositories. The same choices apply for other types of software-related STI such as documentation and downloads. The latter is process of collecting information about STI, archiving a copy, and linking to the original from a central site. Each item collected this way forms a record and the collection of records are put, in aggregate, on a website for customers. While a mandate is simple, it is generally inflexible. Aggregating metadata and archiving STI is the more sustainable approach because it allows developers to use the tools that they like while reducing competitive forces that might otherwise drive them to, say, ignore a mandate. That is, DOE Code will integrate with open communities instead of trying to replace them.

Software tools that aggregate metadata but provide a limited amount of original content are called *aggregators.* Aggregators are very common for news media and you may be familiar with sites like Google News, Pulse or Huffington Post which aggregate news from hundreds of sites. As an aggregator, DOE Code will behave the same way, but pull data from many sites on the internet using *Connectors*. Using connectors will make it possible for DOE Code to provide a large amount of metadata about software projects, but directly link to the services where the code is actually stored. This would include any tool that the developers are using (GitHub or Bitbucket), services provided through National Laboratories (like ORNL's installation of GitLab), or individual websites. Each connector will be treated the same, but have an implementation that handles the peculiar details of each site.

Many people who spoke at OSTI's series of 2015-2016 workshops asked for OSTI to host repositories as alternatives to public services like GitHub. Their chief concern was a valid one: DOE owned repositories would, in theory, not run afoul of DOE's own regulations, whereas licenses for online hosting services might. So, in addition to connecting to repositories hosted by online providers, OSTI plans to offer repository hosting services in two different ways, both of which will directly connect back into DOE Code. The first way will be through repositories hosted directly by OSTI on OSTI's resources, which could be either open to the public or completely private.

The second way is through creating *organizations* or *groups* on common online providers. You can checkout the second case by going to OSTI's GitHub organization where the source code for the new DOE Code is already hosted, [7], (more information below). The idea with this case is that OSTI should have a presence in the open communities. If we make organizations or groups, then it is far easier to both offer “OSTI-hosted” repos on that community and to track and archive all DOE software repositories on that service.

Another important consideration that came up repeatedly in the workshops was social media integration. This goes beyond simply adding buttons to “Share” content. It includes things such as a notification system to watch for news and updates about sets of software and links to author profiles. OSTI also needs to maintain the existing workflow with STI managers at the various laboratories and it may be possible to streamline this through messaging and sharing services hooked into DOE Code.

Researchers are quick to point out that if DOE Code is an aggregation service and a modern web application, then it needs an open, web-accessible Application Programming Interface, or “web API.” Web APIs are used to access a service or website using codes, or programmatically. Programmatic access makes it possible to use the service without a human in the loop. There are lots of applications for this, such as software engineering research to look at how DOE software has changed over time, other types of data analytics, and move advanced cases like continuous integration, performance analysis, and cloud-based code retrieval and execution.

All of these things mark great changes in the transformation of the ESTSC into DOE Code, but there will undoubtedly be many more feature requests and amazing ideas from the community. Which leads us to perhaps the most important change of all for DOE Code: Community Engagement.

Community Engagement

To be honest, something that we could do much better with DOE Code is engage the community in the whole process of redesigning it, developing it, and – if they are willing – using it. It should not only serve as the archival service for DOE software packages, but it should act as a kind of social hub for all things related to DOE software, which is a natural evolution inline with how other products have grown.

This starts by engaging the community in the development of requirements. We have created many requirements teams to advise us on all the features that we need for DOE Code and we are meeting with these teams as quick as we can. These teams are broken down between DOE, National Laboratory and External Teams. The DOE Teams include:

- Core – The core group of people tapped by OSTI to provide and analyze requirements
- OSTI-Technical – The technical team at OSTI responsible for implementing DOE Code
- OSTI-Policy - Policy team at OSTI responsible for DOE Code
- DOE-GC - DOE General Counsel
- DOE-ASCR - DOE Office of Advanced Scientific Computing Research
- DOE-AT - DOE Applied Technologies Offices

The National Laboratories have many requirements of their own, so we are creating the following Lab teams:

- Labs-Software - Software experts from multiple labs
- Labs-Data - Data scientists and consumers from multiple labs
- Labs-Operations - Operational experts from multiple labs
- Labs-HPC - High-Performance Computing researchers from multiple labs
- Labs-STI Managers - The current STI managers/Software Points of Contact at each lab
- ECP - Representatives of the DOE's Exascale Computing Program

We have designated several “external” teams because the DOE works with many institutions outside of the laboratories and in many cases these institutions are required to submit STI to OSTI. Our external teams include:

- University Partners - Representatives of OSTI's University Partners
- Industry Partners - Representatives of OSTI's Industry Partners
- Students - The best and brightest software engineering, computer science and related students in the complex
- Postgraduates - Postdocs, post-masters and post-bachelors appointees who work with software

We have also engaged a software architect from Oak Ridge National Laboratory to work with the requirements teams and lead the development of DOE Code. Jay Jay Billings - who is actually an astrophysicist by training - is on-site at OSTI working on this project every week. Jay's research career still includes a small amount of astrophysics, but he spends most of his time working on scientific software projects at ORNL and leading their new Scientific Software Initiative, which hopes to improve the way ORNL approaches software development. In addition to his research career, Jay is active in the open source community and particularly so in the Eclipse Foundation, where he has the distinct honor of serving on the Architecture Council that guides the development of the platform and provides mentorship within the community.

In addition to our requirements teams, we are also hitting the road to get the word out. We have identified several conferences and workshops where OSTI personnel will be on hand or speaking to tell attendees about DOE Code as well as other OSTI products. We are hoping to do a few conferences a year, so feel free to send us suggestions! The goal with this sort of direct outreach is to make it clear that DOE Code is here for the community.

One of our great hopes with DOE Code is that it can become a community project for the whole DOE Complex and beyond. To that end, we have made it open source and available on GitHub, [6]. (We have even created a whole DOE Code organization on GitHub where interested developers can request repositories, [7].) We hope to engage software developers across the Complex who want to test, contribute to, or maybe even deploy the system at their site. This also makes it possible to accept feature requests and bug reports directly from users without requiring a single point of contact. Documentation, development records and guidelines, requirements, and all sorts of information will be available on GitHub. Of course, this means that DOE Code will actually have a record on itself!

Timeline

A firm “release date” for DOE Code has not been established. Since the goal is to have a modern project, we plan to release the new capability incrementally. This will allow us to release early, often and iteratively so that we can quickly respond to changes in requirements. Each iteration will include requirements analysis, code development, community engagement, increased support for providing repositories, and improvements in our archival services. This should result in a fully featured product in the time frame of eighteen to twenty four months.

If you would like to know more about the project, feel free to reach out on Twitter (@OSTIgov or @jayjaybillings) or email Jay directly at billingsjj <at> ornl <dot> gov. We would love to hear from you!

[1] – In memoriam: The remarkable career of Margaret Butler, http://www.anl.gov/articles/memoriam-remarkable-career-margaret-butler
[2] – OSI FAQ, https://opensource.org/faq#osd
[3] – 2016 Future of Open Source Survey, https://www.blackducksoftware.com/2016-future-of-open-source
[4] – GitHub, https://github.com
[5] – The Eclipse Foundation , https://eclipse.org/org/
[6] – DOE Code Repository, https://github.com/doecode/doecode
[7] – DOE Code GitHub Community, https://github.com/doecode