

 FTA Tech Conference

Where does software come from?

@elstudio August 2019

Where does software come from?

- We think of software development as solo, pizza-fueled effort
- Where software actually comes from
 - It's a team sport
- Open-source software multiplies effort
- Implications for public sector teams:
 - Collaboration and skills
 - Policy and governance
 - Security



@elstudio

Eric Johnson

Solutions Engineer at **GitHub**

- A developer, tech lead and consultant
- I help with digital transformation for the public sector
- Also a:
 - Runner, Talker
 - Never actually a camp counselor



Software didn't used to exist at all

In 1967 software comes from

NASA hires a lab at MIT to make a computer to navigate men to the moon

- Margaret Hamilton's team made the code for the Apollo Guidance computer
- On the way they invented modern software engineering
 - As an afterthought!
 - (The contract for the AGC didn't even mention software)



**Still, software
was
enormously
expensive**

- The Apollo guidance software required **1,440 person-years** of work
 - From a **350-person engineering team** at peak
 - With MIT's university infrastructure

That's a lot of pizza!

- What if you don't have those sorts of resources?



Photo by [Fatima Akram](#) on [Unsplash](#)

In 1991 software comes from

When a developer loves a problem very much

Linus Torvalds had ideas for multitasking with Intel's 386 processors.

“I’m doing a (free) operating system (just a hobby, won’t be big and professional like gnu) for 386(486) AT clones.”

The Linux kernel.

Where does software come from?







Interconnected community

**Software is far
beyond
individual
efforts**

**That's true for
Apollo and for
Linux**

- Linus released kernel version 0.1 in 1991 with 10,230 source lines of code
- A recent Linux kernel (version 5.2) has over 28m
- That's contributions of 13,594 developers
 - and 1,340 companies
 - since 2005, says the Linux Foundation
- A 2010 estimate valued this work at \$1.6b

Where software actually comes from

In 2016 software comes from

A farmer tired of driving his tractor down the same old ruts

- Matthew Reimer's Pixhawk-powered tractor drives itself
- And gets written up in the [Wall Street Journal](#)



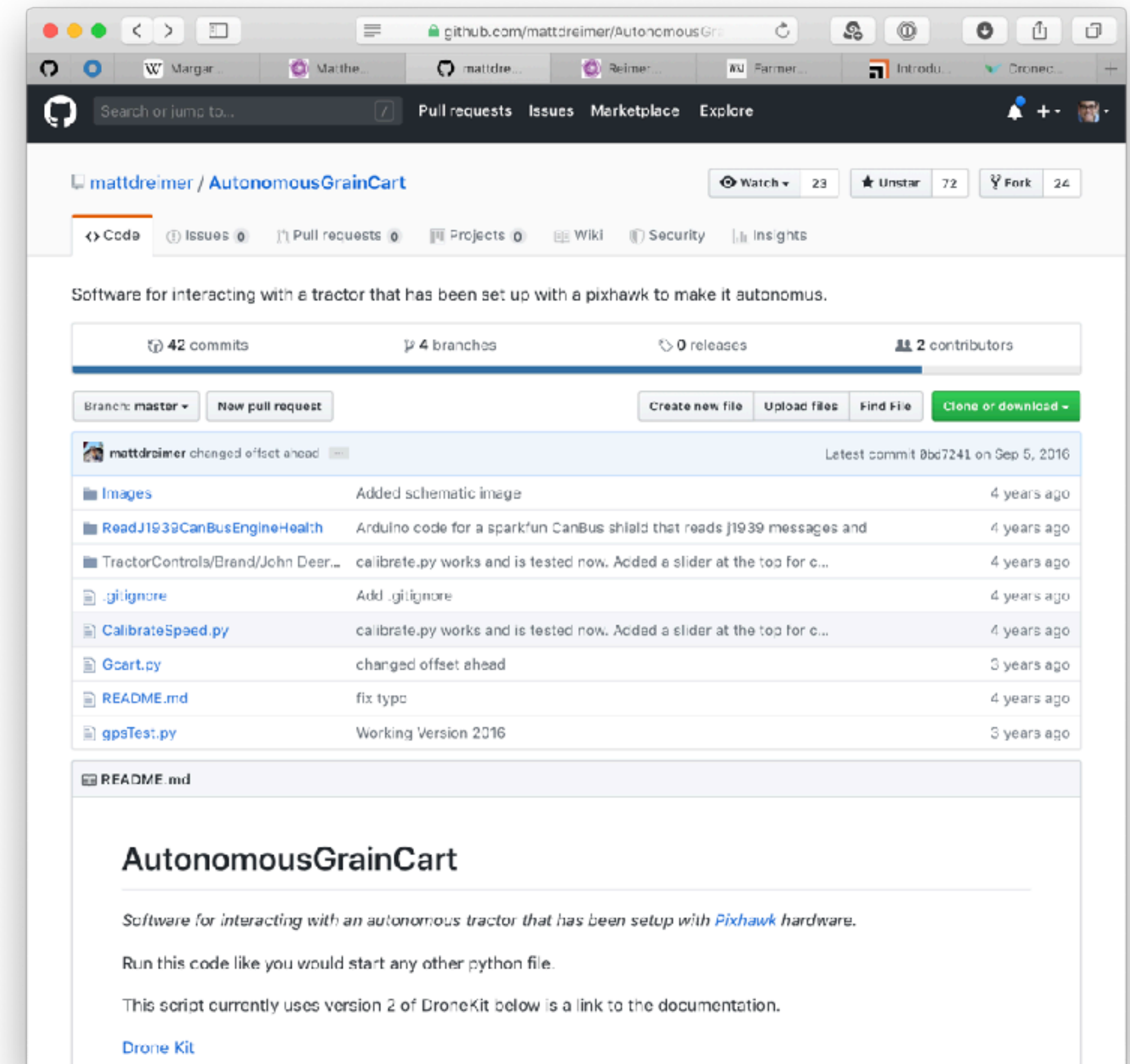
“The robot tractor isn’t a prototype or top-of-the-line showpiece. It’s an eight-year-old John Deere that the 30-year-old Mr. Reimer modified with drone parts, open-source software and a Microsoft Corp. tablet.”

– Jacob Bunge, [Farmers Reap New Tools From Their Own High-Tech Tinkering](#), WSJ, May 2, 2016

Software comes from

Matthew's code relies on other open-source tech

- Hardware:
 - A John Deere tractor
 - Pixhawk drone controller
- And software:
 - Interface libraries for Python
 - Matthew's Python code





Matthew spent \$8,000 on his DIY self-driving tractor

He's not a U-Michigan educated mathematician at MIT
Or a CS student from the University of Helsinki

Software has changed.

Pixhawk has its own story

In 2008 software comes from

A PhD student who wanted to make a drone fly by itself

Lorenz Meier found he first had to make it fly

- Flight software and hardware wasn't his expertise, so he recruited a team of students
- The Pixhawk team released their software as OSS



“By choosing to build an open source community around this, I ensured that development efforts and research results of many years and many talented people worldwide were combined with a full-scale solution which was reusable and standardized.

– Lorenz Meier, [How I accidentally created the most used standards in the drone industry](#)



“Combined, we had much more development power and skills than any of the well-resourced companies in the field.”

– Lorenz Meier, [How I accidentally created the most used standards in the drone industry](#)

Open-source communities bring challenges

Open hardware

“Small” hardware changes make failures. Users blame software

Usability issues

Tough to address w/ community contributions

Overall organization

Maintainers spend lots of time keeping the overall software organized

Reliability and UX

Users expect product-level user experience



How Pixhawk solved these is illustrative



Process to the rescue

Code changes evaluated by 2 people
Checked by automated flight testing —
about 1,000 test flights a month





Community gets more expert

Instead of academics and enthusiasts contributors now use PX4 in
some type of product

Specialists (control theory/robotics, computer vision) see their area

There's a gap at the center





Governance becomes more important

Dronecode, part of the Linux Foundation, coordinates industry and open-source developers

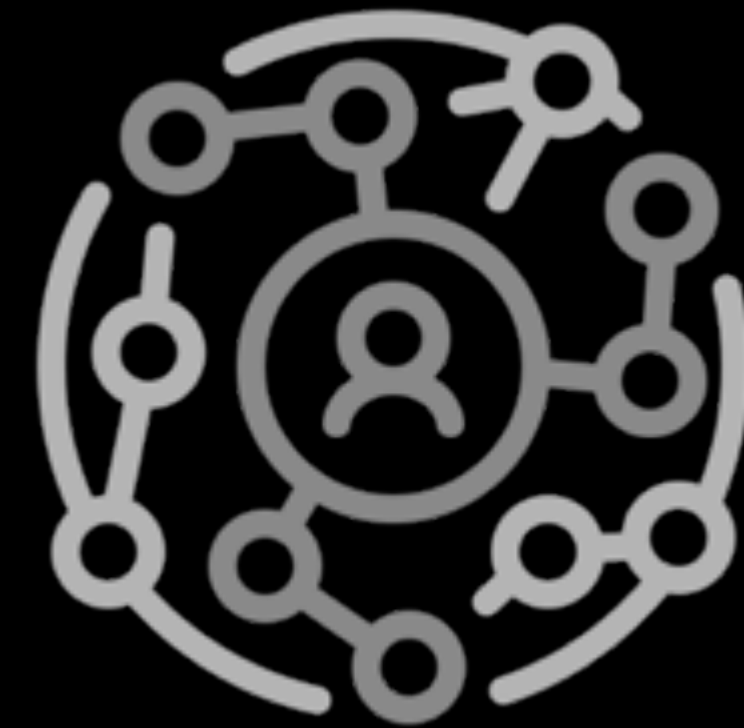
Pixhawk found it needed:



Collaboration



**Process &
Testing**



Governance

“Go find a problem worth solving for you and others and build a community around it!”

– Lorenz Meier, [How I accidentally created the most used standards in the drone industry](#)



Open-source software multiplies effort. But is it a free lunch?



207

Direct contributors

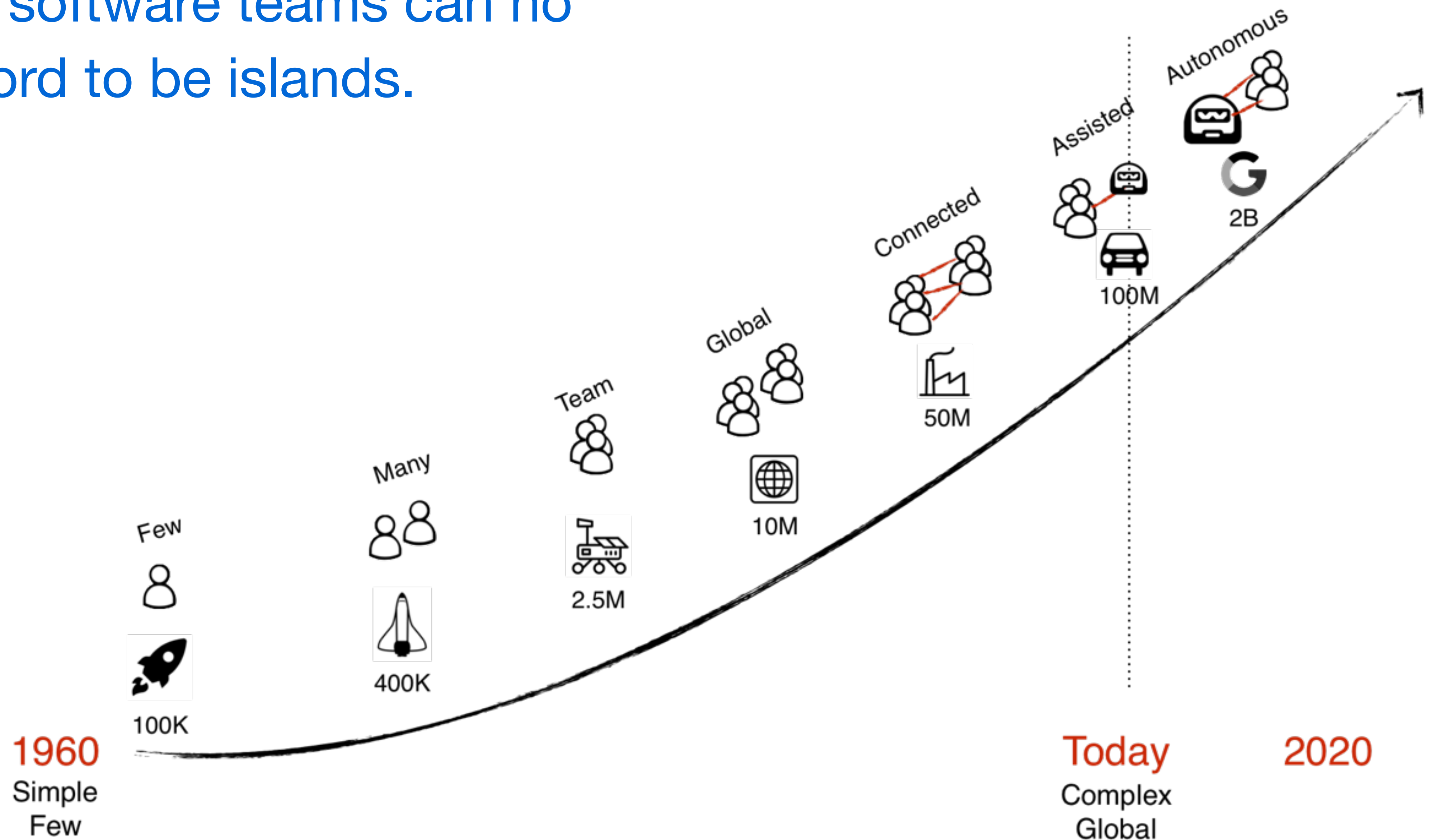
achael/eht-imaging h5py/h5py PythonCharmers/python-future networkx/networkx scipy/scipy scikit-image/scikit-image jek/blinker
brandon-rhodes/pyepphem kennethreitz/requests codecov/codecov-python urllib3/urllib3 micheles/decorator pandas-dev/pandas erikrose/
more-itertools pyamg/pyamg pytest-dev/execnet nex3/pygments pypa/readme_renderer python-attrs/attrs yaml/pyyaml boto/botocore chardet/
chardet PyCQA/flake8 aliles/funcsigs imageio/imageio ThomasWaldmann/argparse blink1073/tifffile jupyter-widgets/ipywidgets pytest-dev/py
numpy/numpy sphinx-doc/sphinx calvinchengx/python-unittest2 pytest-dev/pytest-xdist kwgoodman/bottleneck pallets/markupsafe
kevin1024/pytest-httpbin pytest-dev/apipkg python-excel/xlwt Anorov/PySocks jupyter/nbconvert pytest-dev/pluggy python/cpython pytest-dev/
pytest-runner certik/enum34 statsmodels/statsmodels Blosc/python-blosc spatialaudio/nbsphinx pytest-dev/pytest ericgazoni/
openpyxl msabramo/tox boto/boto3 pallets/werkzeug python-greenlet/greenlet zzzeek/sqlalchemy ipython/ipykernel pytest-
dev/pytest-faulthandler paxan/python-dateutil dask/fastparquet scikit-learn/scikit-learn PyCQA/pyflakes gsneadders/
python-webencodings pytest-dev/pytest-rerunfailures benjaminp/six python-excel/xlrd cython/cython theacodes/cmarkgfm youngpm/gdalmanylinux
spulec/moto matplotlib/matplotlib matplotlib/cycler pytest-dev/pytest-forked schlamar/nose-ignore-docstring numpy/numpydoc shibukawa/
snowball_py python-hyper/brotli py dask/dask pydata/xarray bitprophet/alabaster cheshire/virtualenv certifi/python-certifi eleddy/numexpr
pypa/wheel astropy/astropy ipython/ipython matthew-brett/nb2plots waylan/beautifulsoup pallets/flask chevah/python-cffi PyCQA/pycodestyle
shibukawa/imagesize_py newvem/pytz airspeed-velocity/asv cloudpipe/cloudpickle eliben/pycparser adamchainz/flake8-comprehensions
MobileDynasty/pytest-env brettcannon/importlib python-pillow/Pillow kjd/idna tornadoweb/tornado python-babel/babel pallets/itsdangerous
kataev/flake8-rst PyCQA/mccabe matthew-brett/texext joshspeagle/dynesty nose-devs/nose mdsitton/configparser-3.2.0r3 lxml/lxml
HypothesisWorks/hypothesis getsentry/raven-python calvinchengx/python-mock sphinx-gallery/sphinx-gallery spyder-ide/qtpy gitpython-developers/
GitPython nucleic/kiwi choldgraf/sphinx-copybutton rtd/sphinx_rtd_theme html5lib/html5lib-python pallets/click nedbat/coveragepy
pallets/jinja mwaskom/seaborn pydot/pydot pytest-dev/pytest-mock njsmith/colormap SimpleITK/SimpleITKPythonPackage
jazzband/contextlib2 mozilla/bleach tox-dev/detox pyparsing/pyparsing



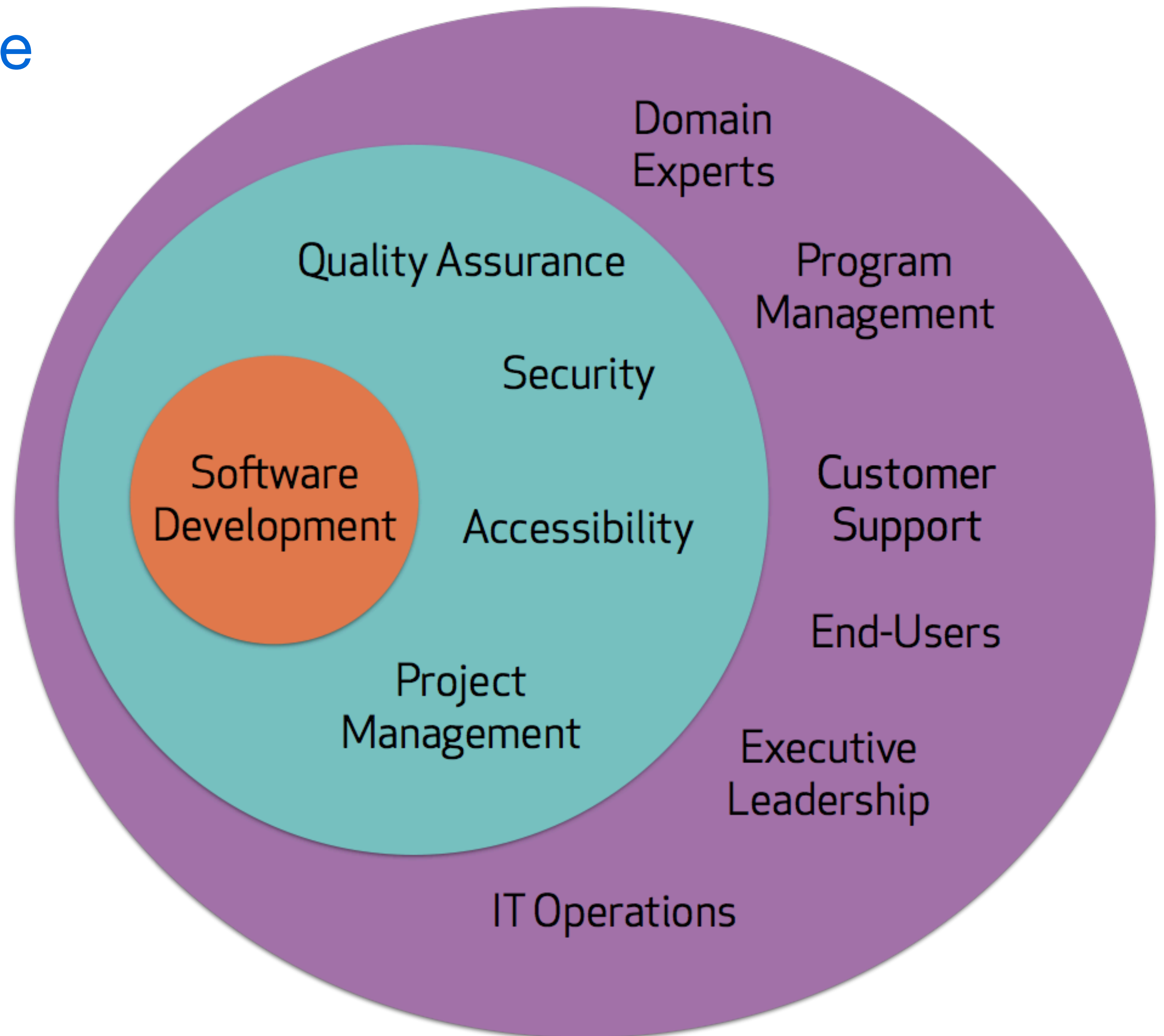
21,485
Community contributors

Implications for public sector teams

Apps and software teams can no longer afford to be islands.



Everyone Speaks Software



Expectations have increased, even for the public sector

- The supercomputer in our pockets changes delivery models
 - Constituent expectations increase
 - Cost of service can decrease
 - But also increased cost of software development
- Product focus, rather than project focus doesn't come naturally

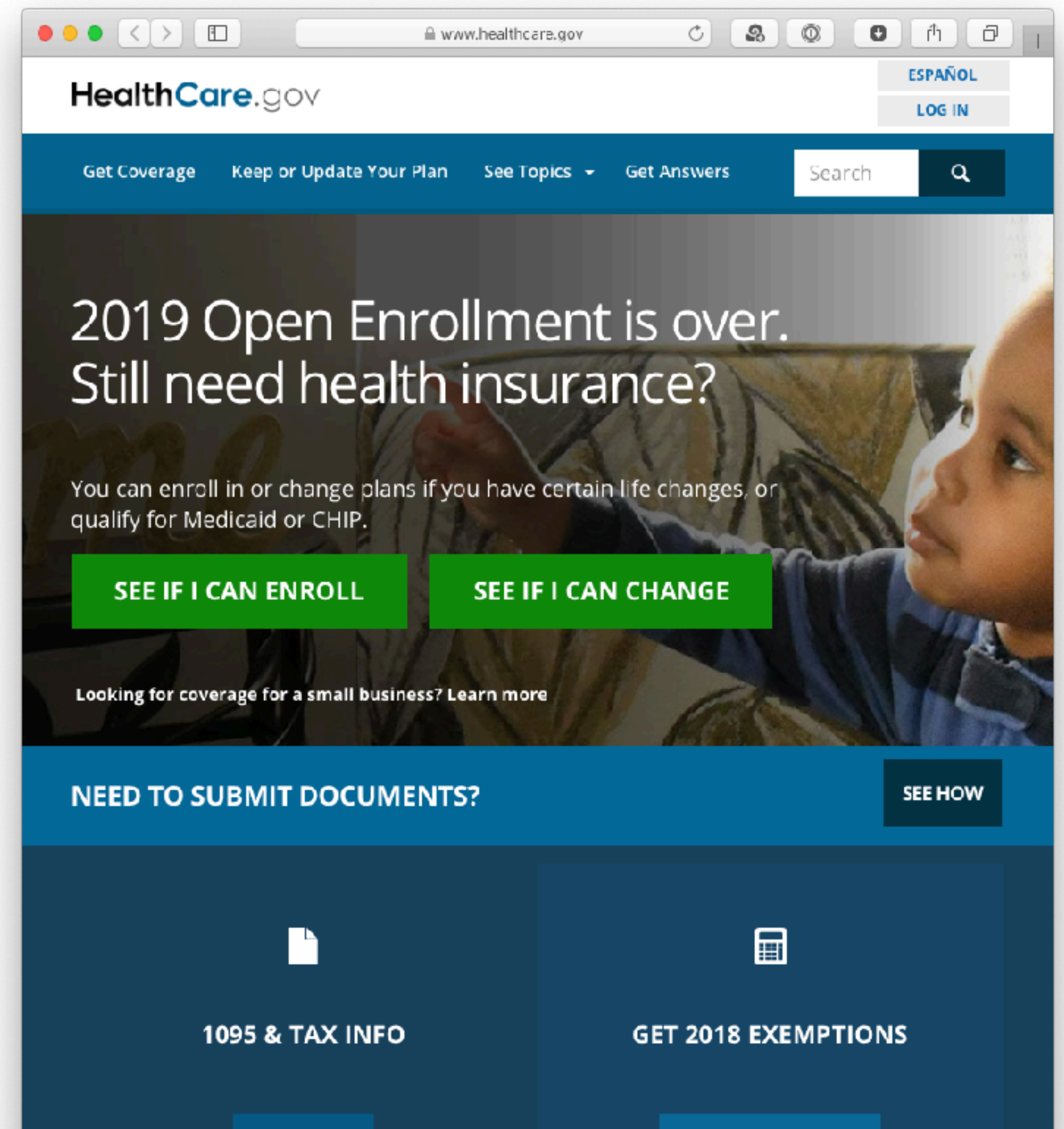
It's not only companies.

Remember the launch of Healthcare.gov?

“Reporters and the public were asking increasingly hostile questions... that struck at the heart of not only Obama’s competence, but his entire political vision. If the government could not successfully implement existing programs, citizens could plausibly ask, why let it create new ones?”

Harvard Business Review

Schlesinger and Bhayani, HealthCare.gov: the Crash and the Fix





Software transformation at VA

VA | Technology Incubator

Home Lighthouse SaaS Open API Pledge Microconsulting

SaaS at VA › SaaS Catalog › GitHub

GitHub


Product description

GitHub is a code hosting platform for version control and collaboration using Git. It lets you and others work together on projects from anywhere. It provides access control and several collaboration features such as bug tracking, feature requests, task management, project Kanban boards, and wikis.

Approved use case

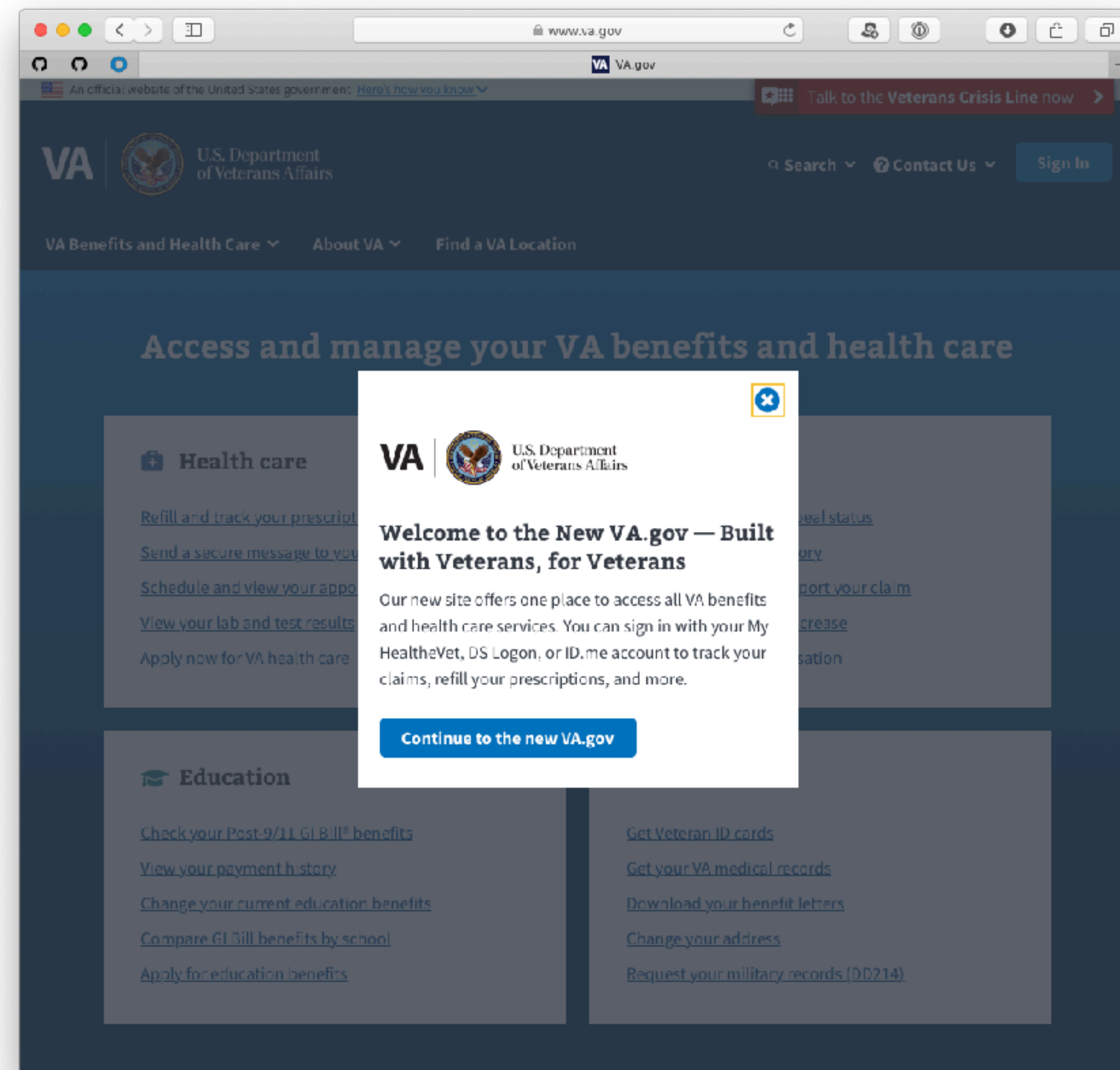
GitHub is authorized for authority to operate (ATO) low usage; it cannot include any protected health information (PHI) or personally identifiable information (PII). Users must complete mandatory VA training.

How to get started using this product at VA



GitHub Product Attributes

- FedRAMP Authorized
- Free tier available
- Enterprise - No funds required
- FISMA Low
- Two factor authentication required



The screenshot shows the new VA.gov website interface. At the top, there's a navigation bar with the VA logo, U.S. Department of Veterans Affairs, and links for Search, Contact Us, and Sign In. Below this, a large banner reads "Access and manage your VA benefits and health care". A prominent white modal box in the center says "Welcome to the New VA.gov — Built with Veterans, for Veterans" and provides information about the new site's capabilities, including signing in with My HealtheVet, DS Logon, or ID.me. A blue button at the bottom of the modal says "Continue to the new VA.gov". The background shows a grid of service categories: Health care (with links for prescriptions, messages, appointments, lab results, and health care application) and Education (with links for GI Bill benefits, payment history, education benefits, school comparison, and education application). On the right side, there are links for ID cards, medical records, benefit letters, address changes, and military records requests.



What would it take to empower other VA teams?

VA goals:

- Software quality
- Improve velocity
- Improve delivery
- Open VA data to external development via API
- Speed contractor onboarding
 - Improve transition to ops/maintenance
 - Control IP
 - Comply with Federal Open Source directives

Challenges

- Legacy tooling and processes
 - Heavily customized Rational tooling
- Unevenly-adopted DevOps practices
- Team education and onboarding

Solution

- GitHub deployed enterprise-wide
 - Enterprise Cloud and Server
- Enterprise rollout plan
 - Training and migration support
 - DevOps tooling baseline + options



Federal and state governments now encourage code sharing

Sharing code

code.gov's compliance dashboard

M-16-21

The Federal Source Code Policy requires 100% code inventory along with 20% Open Source publishing

The Pentagon's FY 2018 NDAA

Requires unclassified, non-defense article code to use open source licenses and repositories

States like **California** have followed:

<https://code.ca.gov/>



California's reasons for encouraging OSS code sharing are typical

<https://code.ca.gov/about-us/>

- Improve access to the state's custom-developed software
 - Support state government-wide reuse
 - Encourage public collaboration
- Make it easier to conduct software peer review and security testing,
 - to reuse existing solutions and
 - to share technical knowledge



Government Drivers for GitHub



Transformation

Modernization efforts like DevOps, DevSecOps, Agile, and Cloud require modern tooling and approaches.



Improved Security & Quality

Having security and compliance integrated into the SDLC fabric rather than bolted on at the end is key to working responsibly as an Open Source Enterprise.



Transparency and IP Control

Contractor led development must be collaborative and include government stakeholders in the process rather than just reported on.



M-16-21

The Federal Source Code Policy requires 100% code inventory along with 20% Open Source publishing.



Talent acquisition and Retention

The new breed developers need to be given the right tools for the job.



Does this mean we have to develop everything in public?



Nope.

But there are advantages to working collaboratively —
even behind the firewall.

In fact, there's a name for this.

Inner- Source

“Use of open source software development best practices and the establishment of an open source-like culture within organizations”



innerSource(culture, technology) = culture * technology

Best technology, mostly silo-ed mentality
 $\text{innerSource}(1,9) = \mathbf{9}$

Best technology, thriving culture
 $\text{innerSource}(9,9) = \mathbf{81}$



Thriving culture, weak technology
 $\text{innerSource}(9,1) = \mathbf{9}$

Best culture, poor technology
 $\text{innerSource}(3,10) = \mathbf{30}$

Discovering and refining what team members have already built provides non-linear return on investment.

The screenshot shows a GitHub search interface. At the top, the search bar contains the query 'language:java dependency:jackson-xml-databind'. Below the search bar, a sidebar lists various GitHub features with their respective counts: Repositories (0), Code (0), Commits (89+), Issues (0), Pull requests (0), Packages (0), Marketplace (0), Topics (0), Wikis (0), and Users (0). The 'Commits' section is highlighted. To the right of the sidebar, a message indicates 'Showing 89 available commit results'. Below this, three commit results are displayed: 1. 'faster xml jackson databind dependency updated' by anuradha151, committed to anuradha151/semester-final on Jan 6. 2. 'Update pom.xml' by rrauhhub, committed to rrauhhub/games-attempt on Oct 16, 2018. 3. 'Modify pom.xml jackson-databind Version' by alreadyna, committed to alreadyna/BookReviewComm on Feb 21. At the bottom of the sidebar, there are links for 'Advanced search' and 'Cheat sheet'.

language:java dependency:jackson-xml-databind

language:java dependency:jackson-xml-databind All GitHub

Repositories 0

Code 0

Commits 89+

Issues 0

Pull requests 0

Packages 0

Marketplace 0

Topics 0

Wikis 0

Users 0

Advanced search Cheat sheet

Showing 89 available commit results

faster *xml jackson databind dependency* updated

anuradha151 committed to anuradha151/semester-final on Jan 6

Update *pom.xml*

rrauhub committed to rrauhhub/games-attempt on Oct 16, 2018

Modify *pom.xml jackson-databind* Version

alreadyna committed to alreadyna/BookReviewComm on Feb 21

Re-use allows for
better pacing

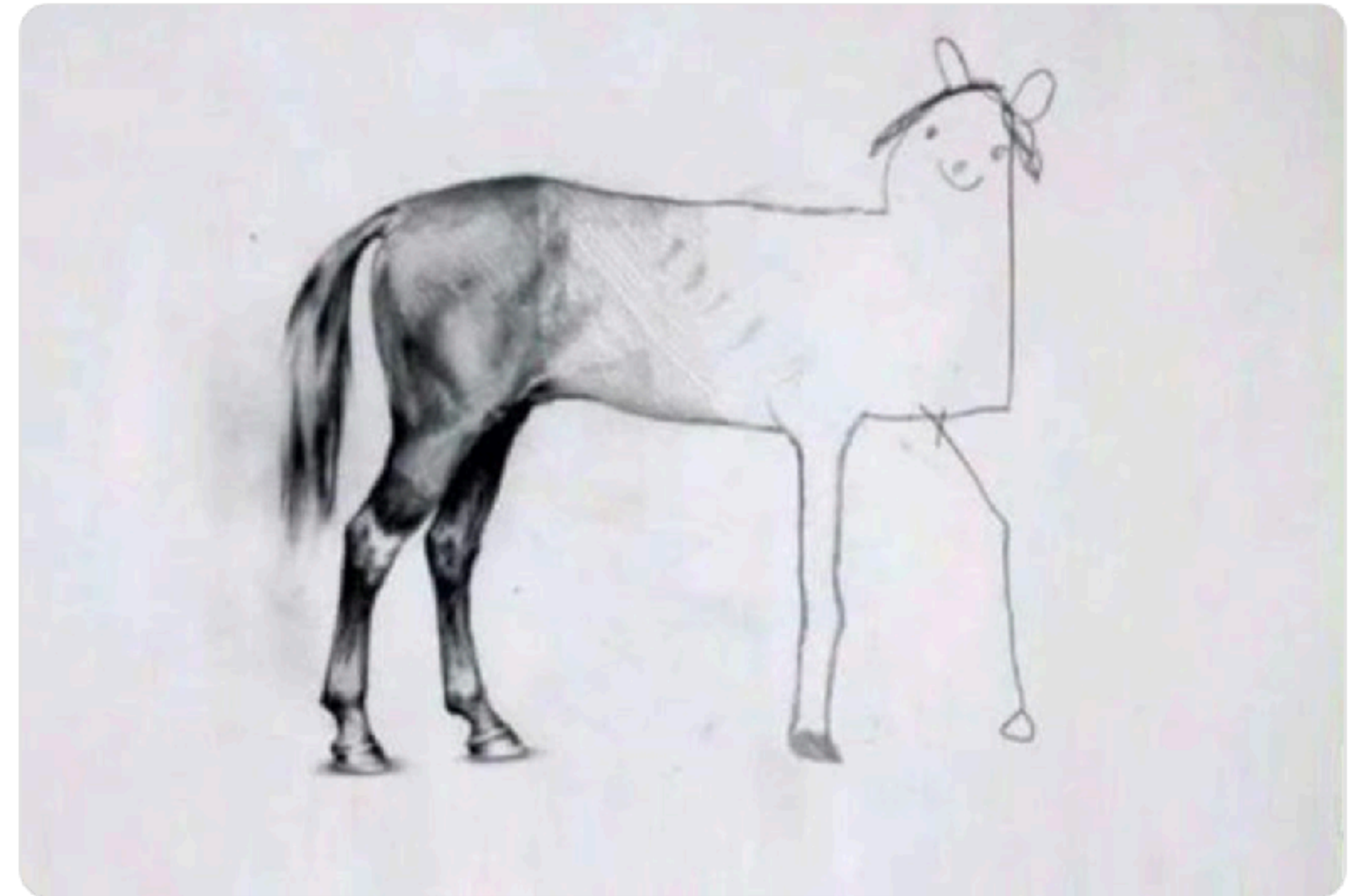


Mark Dalglish
@markdalglish

Following



When your team successfully hits the deadline.



Recommended by Tim O'Reilly and 235 others



Bill Higgins

Follow

IBM Distinguished Engineer focused on culture and workforce modernization at scale

Apr 9 · 11 min read

Tools as a catalyst for culture change

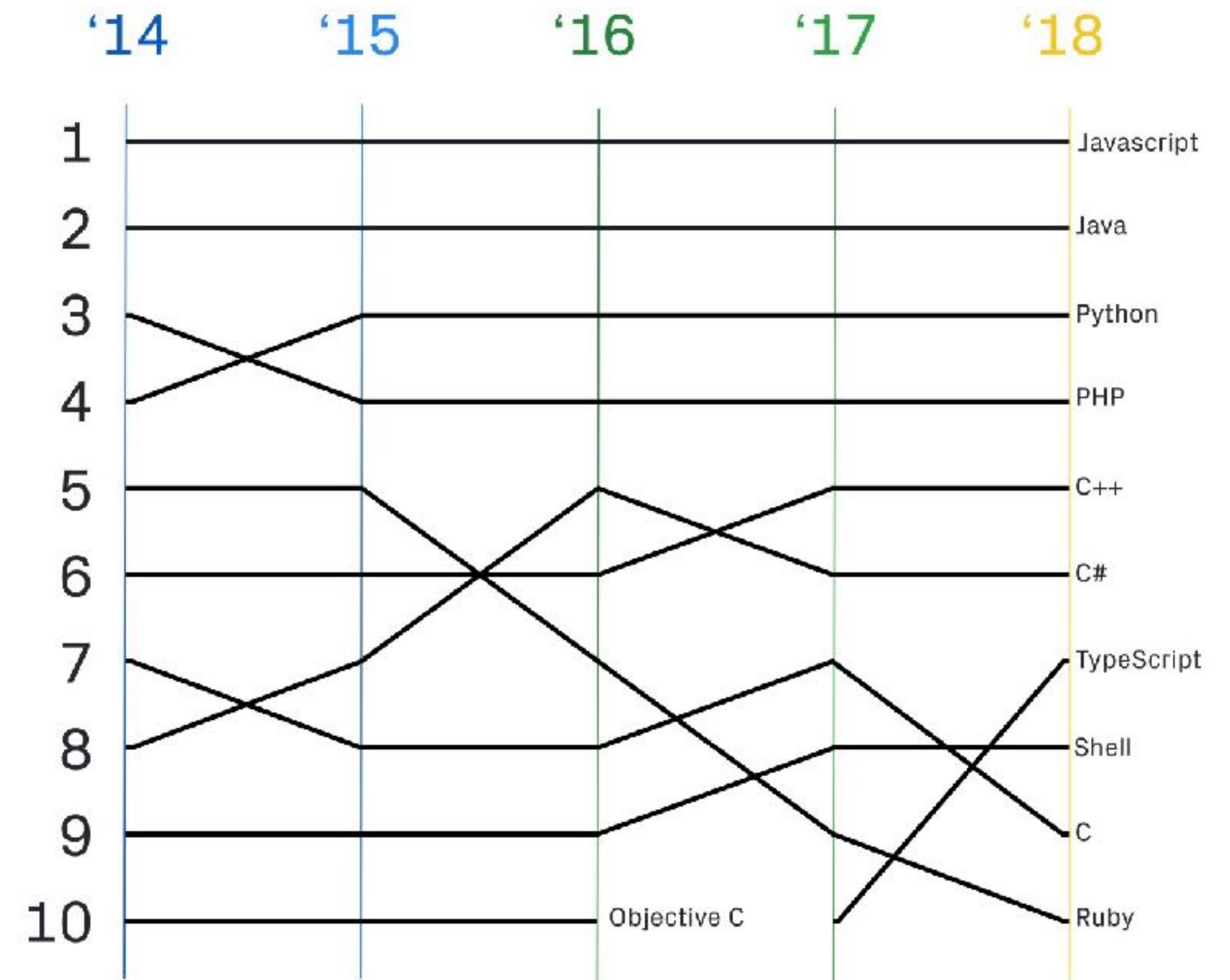
Over the past two and a half years, I've led a project at IBM that deployed a new set of tools to help improve the company's product development efforts. What is the benefit of providing better tools to employees? A first answer is that it helps increase employee productivity. While this is true and part of the answer, it is much too narrow. The broader answer is that giving employees great tools is an excellent way to concretely affect positive culture change.

What about security?

Language popularity over time

- JavaScript, Python, PHP, TypeScript and Ruby in the top 10
- You're probably already using open source
 - Python analytics and machine learning tools anyone?

— [Octoverse 2018](#) report



Where exactly does that code come from?

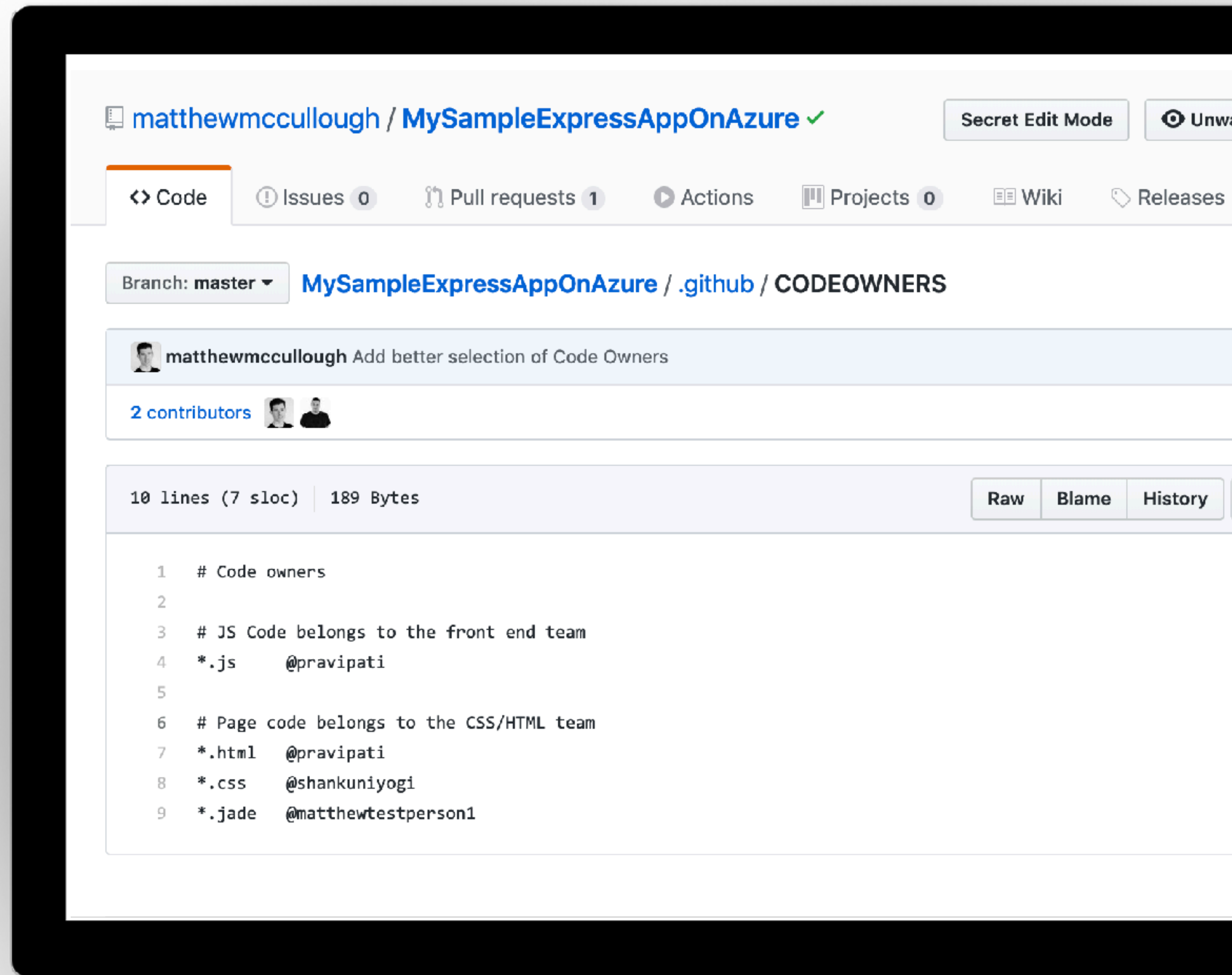
- Contemporary practice increases dependencies
- Maybe 200-odd packages for a Rails app
 - Many maintained by core committers
- JavaScript multiplies this 10-fold
 - A React “hello world” may use 2,000 packages



healthy **habits** for security

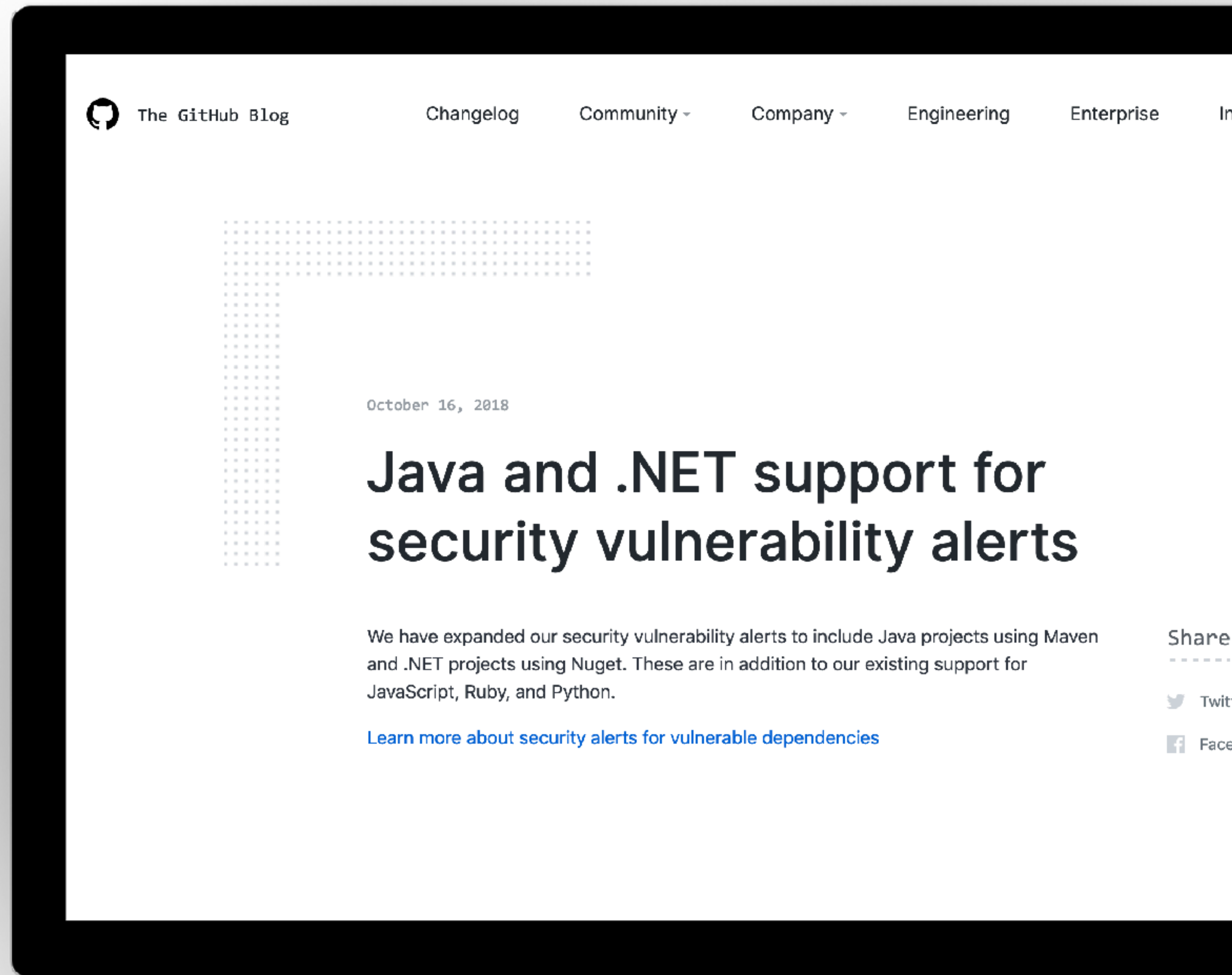
CODEOWNERS

automated workflow
for requested
reviews

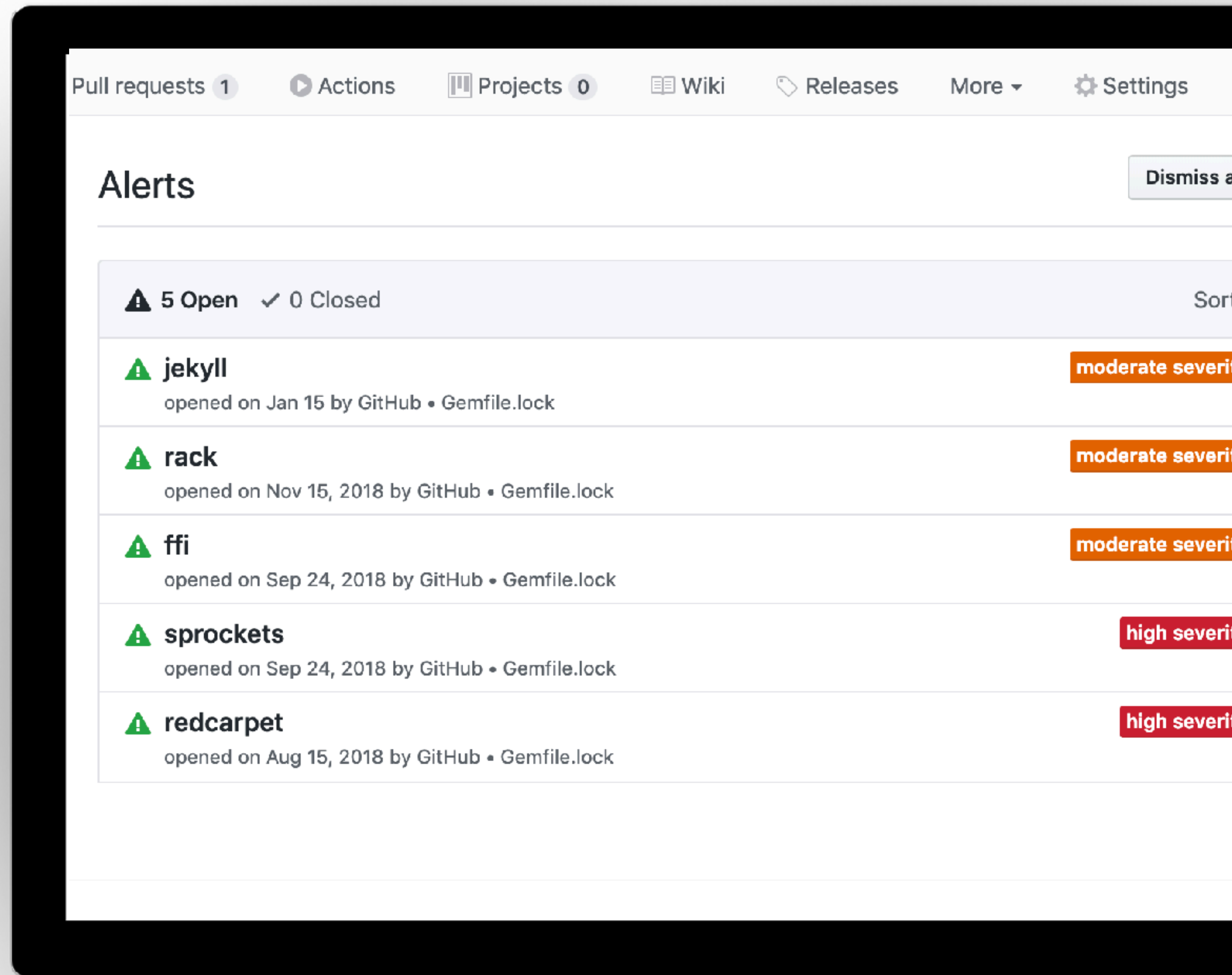


four **eyes** for reviews

Use the
community's
knowledge and the
platform's alerting



GUI Alerts have
click through details



The instructions are
a mere **copy-and-
paste** away from
implementing

jekyll

Dismiss ▾

⚠ Open

GitHub opened this alert on Jan 15

1 jekyll vulnerability found in Gemfile.lock on Jan 15

Remediation

Upgrade jekyll to version 3.6.3 or later. For example:

```
gem "jekyll", ">= 3.6.3"
```

Always verify the validity and compatibility of suggestions with your codebase.

Details

CVE-2018-17567 [↗](#)

moderate severity

Vulnerable versions: < 3.6.3

Patched version: 3.6.3

Jekyll through 3.6.2, 3.7.x through 3.7.3, and 3.8.x through 3.8.3 allows attackers to access arbitrary files by specifying a symlink in the "include" key in the "_config.yml" file.

Command line
users receive the
alerts too

```
♥ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 295 bytes | 295.00 KiB
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 delta
remote:
remote: GitHub found 22 vulnerabilities on JessRudder/text-animation-test
(2 critical, 4 high, 15 moderate, 1 low) To find out more
remote: https://github.com/JessRudder/text-animation-test/issues
[remote:
To https://github.com/JessRudder/text-animation-test
07052a9..41451c3 screenshot -> screenshot
```


You might also consider

Speed up response with automatic scanning

- Automate scanning of source code and static code analysis
 - With tools like SonarQube or maybe Snyk
- Cache known good sources
 - By running your own package registry
 - Nexus or Artifactory
- But **do not slow your response** to community security alerts

How to improve collaboration?

- Put your code in one place
- Make it searchable
- Release your code to the community whenever possible
- Communities of practice
- Tools can help
 - Whether your code must be hosted on-premises or in the cloud

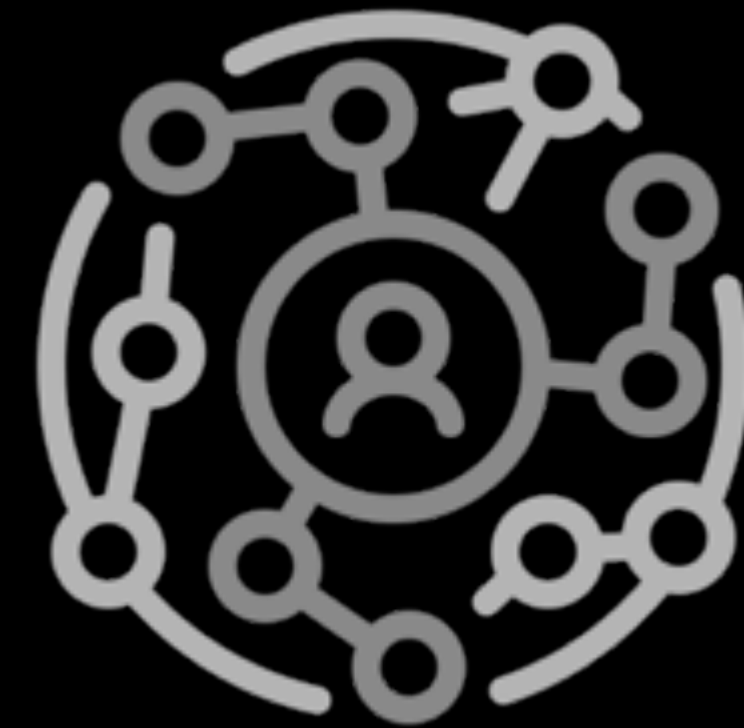
May we all learn from the best in software today



Collaboration



**Process &
Testing**



Governance

In 2020 software comes from

Your team, contributing to the work of thousands

- Let me know how it's working for you!
- Eric Johnson
 - @elstudio on GitHub & Twitter
 - elstudio@github.com



Photo by [Priscilla Du Preez](#) on [Unsplash](#)

GitHub

