Controlling your DevSecOps Journey through Open Source

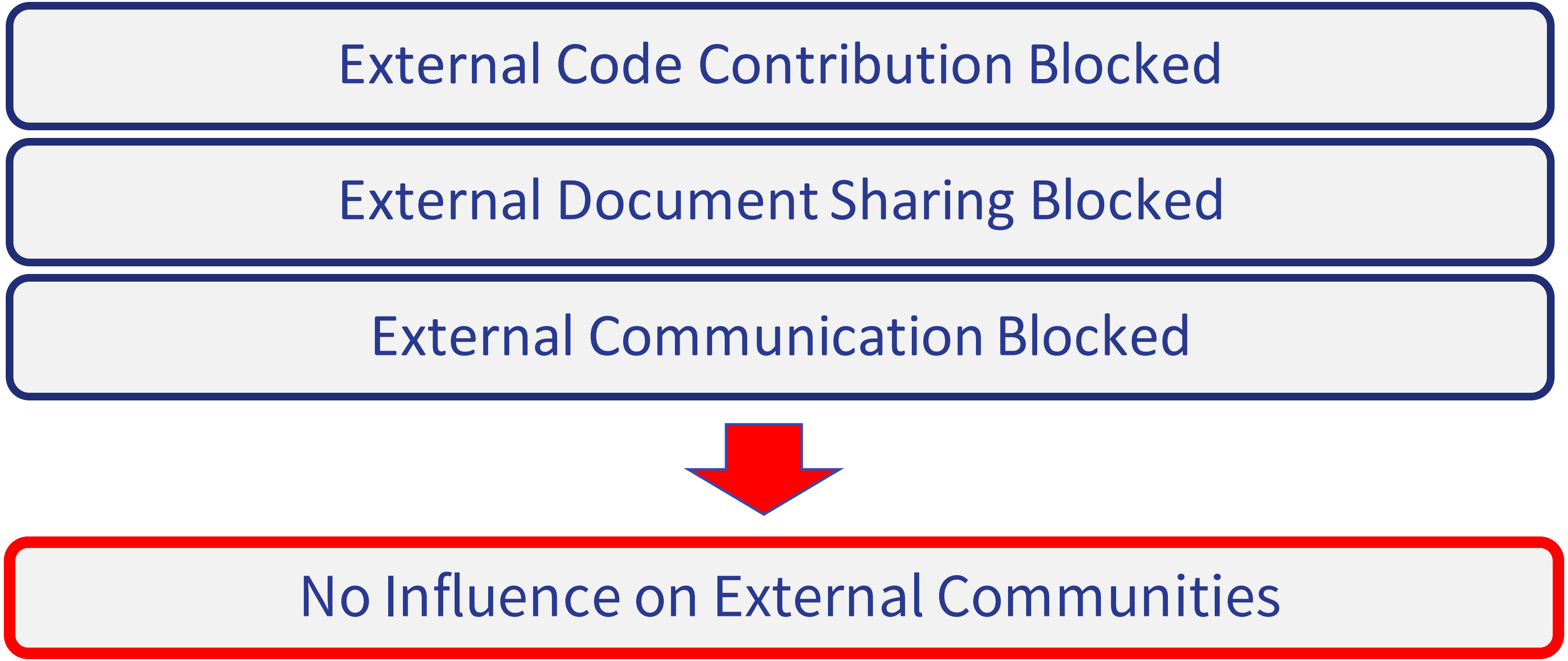
Ragha Vema
Software Engineering Principal | Fannie Mae

Devops Enterprise Summit, Las Vegas NV - Fall 2023

Agenda

- Introduction
- Importance of Open Source Software
- Open Source Program Office @ Fannie Mae
- DevSecOps using Open Source Methodology – Use cases
- Cultivating Growth: The Art of Learning and Adaptation
- Path Forward
- Questions

Why Companies Don’t Participate in Open Source Communities



External Code Contribution Blocked

External Document Sharing Blocked

External Communication Blocked

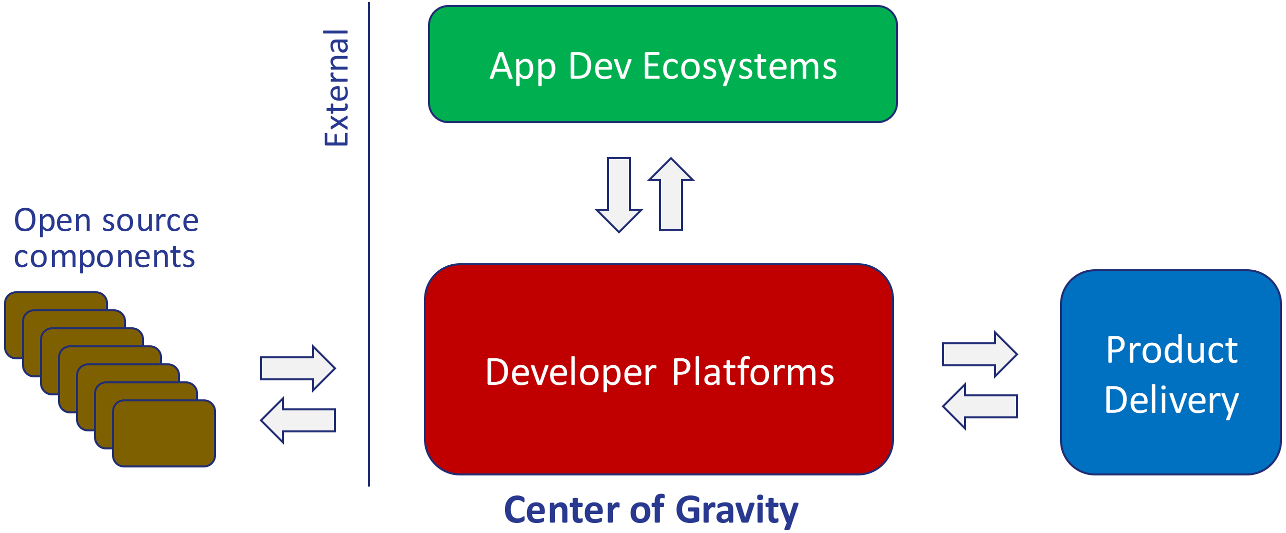
No Influence on External Communities

Is Open Source Scary?

- Scanning tools show thousands of vulnerabilities
- Lots of talk about SBOMs and supply chain security
- Everyone still remembers Equifax
- Log4J, SpringShell, Heartbleed.....

Gartner: Open source transparency makes it more secure than proprietary software

Ideal Open Source Software Lifecycle



Proactive, automated, and streamlined governance with easier ingress, upstream contributions, continuous collaboration and integration, rapid innovation, and high-frequency releases

Where can we start?

OSPO at Fannie Mae

Formed in 2021

North Star for Open-Source consumption and contributions

Build a culture of Innersource and enhance Developer Experience

Open Source vulnerabilities with no clear solution

- Developers use open source libraries to build their applications
- 90% of Modern Apps are built on top of Open Source libraries (per Sonatype 2023)
- CVEs on Open Source libraries with no patch available.

What Can We Learn from Open Source Communities?

Transparency

- Nothing is hidden
- Visible leadership, bug reports, roadmaps

Open Participation

- No strict division between users and creators
- All participants in “the room where it happens”

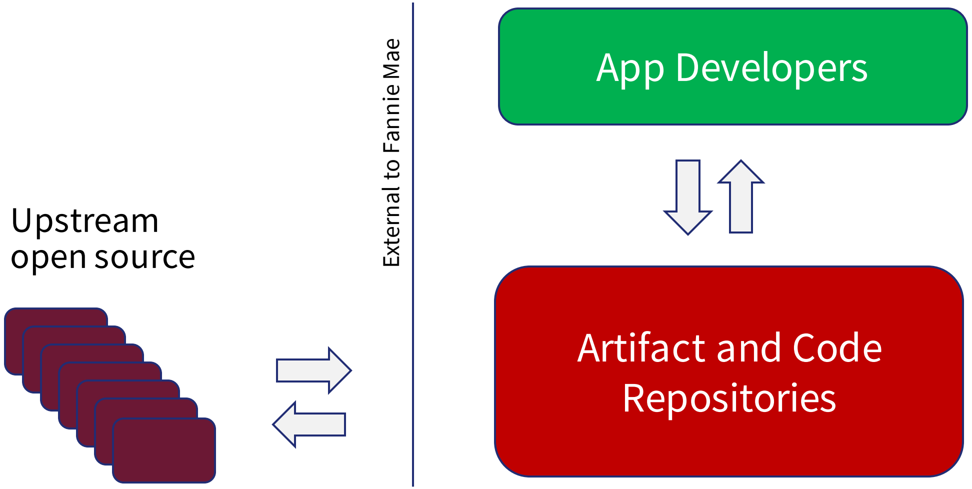
Governance

- Hierarchies of participants
- Rules for interaction, growth, succession

Use the Source – InnerSource – to distribute “golden patches”

Streamline Patching and Distribution

- Build Team to Patch
- Centralized Distribution Point
- Train and enable internal open source maintainers



Upstream open source

External to Fannie Mae

App Developers

Artifact and Code Repositories

Empower Engineers to Streamline Remediation Processes

Top 3 Vulnerabilities

CVE-2016-1000027 Deserialization of Untrusted Data

CVE-2020-13091 Deserialization of Untrusted Data

CVE-2022-1471 Deserialization of Untrusted Data / Improper Input Validation

Spring framework 5.3.x

CVE-2016-1000027

Severity : 9.8 Critical

NVD Published Date: 01/02/2020

Pivotal Spring Framework through 5.3.x suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required

Discovery

Exception

Global Exceptions

Effective vs Ineffective

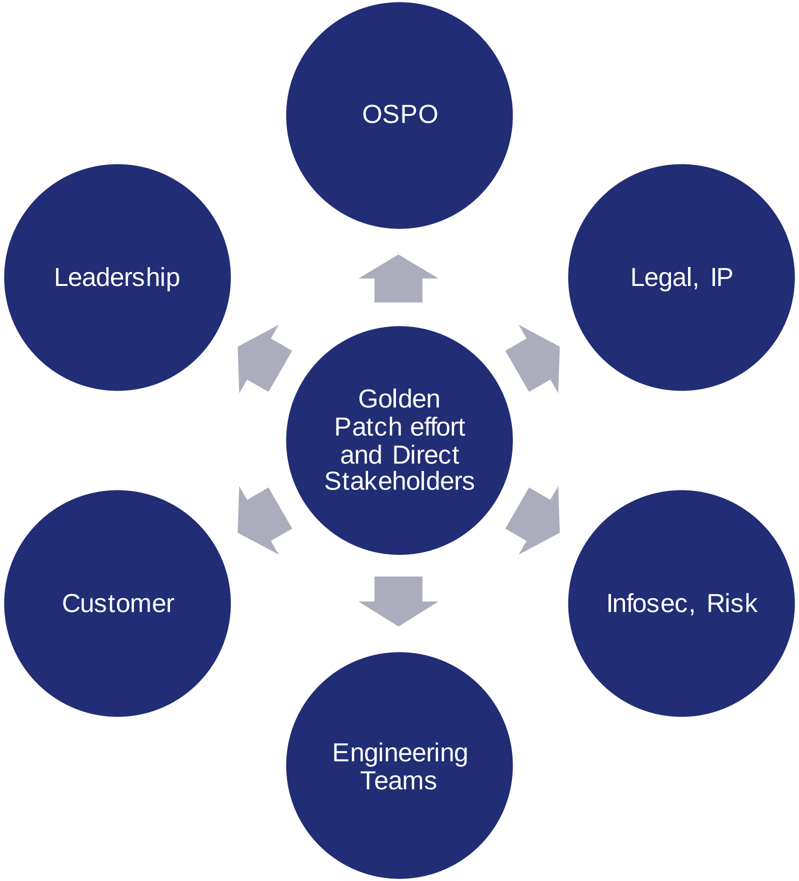
Effective Vulnerability
If the proprietary code is making calls to the vulnerable functionality

Ineffective Vulnerability
If the proprietary code is NOT making calls to the vulnerable functionality

Quest for Solution..

- Github Discussions and Issues
- No clear path unless upgraded to Spring 6.x requiring JDK 17+
- Finally receive a patch from Pivotal – WAF changes and feature flag

Rollout…



Golden Patch effort and Direct Stakeholders

OSPO

Legal, IP

Infosec, Risk

Engineering Teams

Customer

Leadership

Here comes the next one..

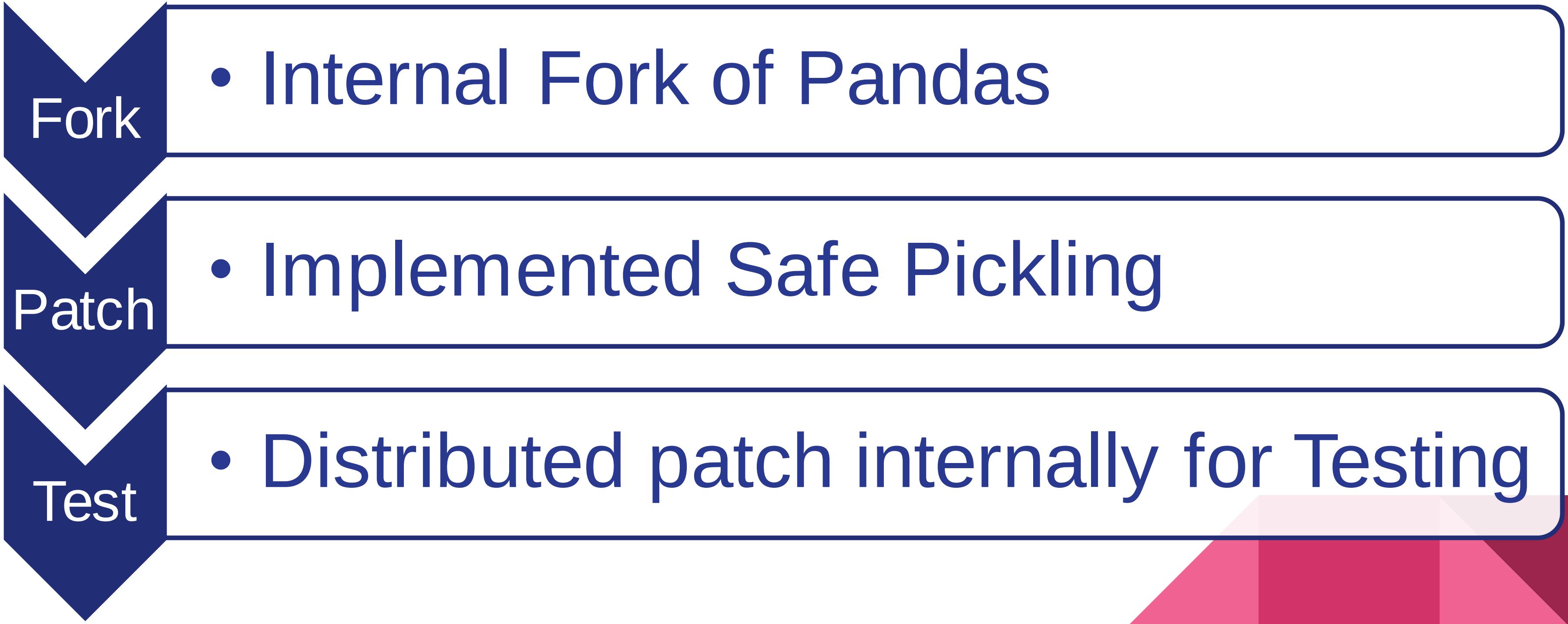
(Numpy) Pandas 1.5.2

CVE-2020-13091

Severity : 9.8 Critical

NVD Published Date: 05/15/2020

pandas through 1.0.3 can unserialize and execute commands from an untrusted file that is passed to the read_pickle() function, if __reduce__ makes an os.system call. NOTE: third parties dispute this issue because the read_pickle() function is documented as unsafe and it is the user's responsibility to use the function in a secure manner.



Fork

- Internal Fork of Pandas

Patch

- Implemented Safe Pickling

Test

- Distributed patch internally for Testing

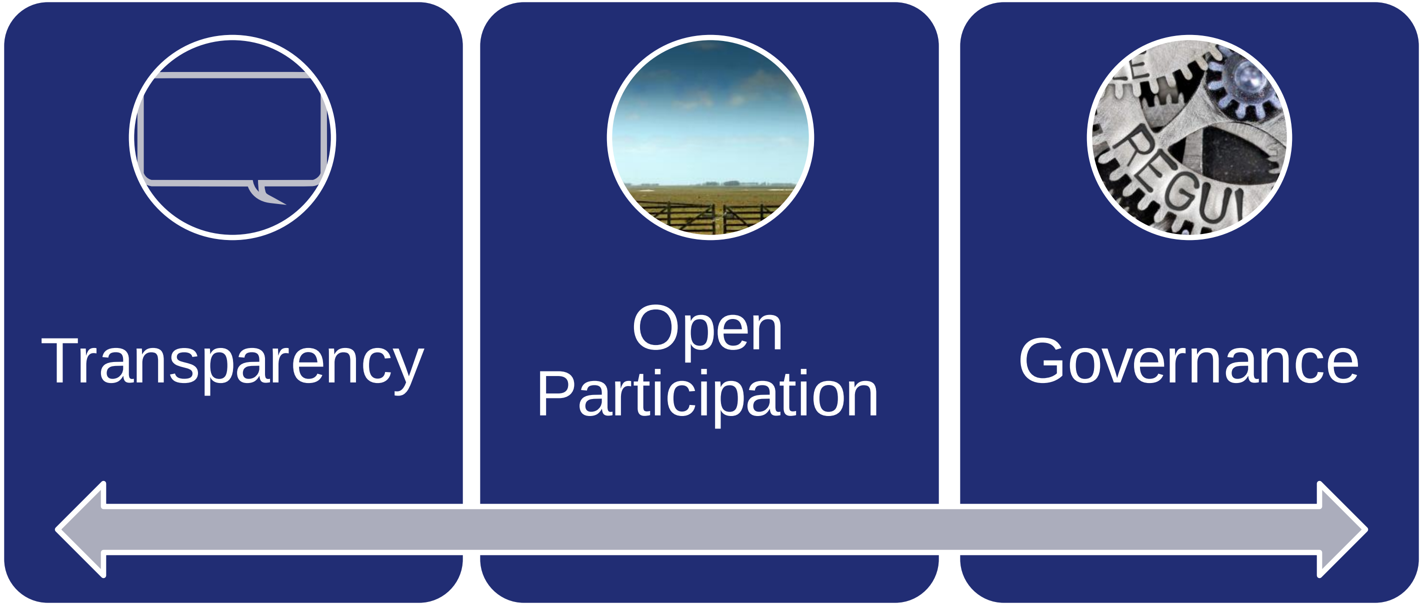
Eureka Moment - Let’s go upstream for a solution!

- The Clean Dependency Project:

- Take the concept up upstream problem-solving and apply it to Fannie Mae’s dependency management

- Proactively identify and modify dependency sources, clean them, and make available for Fannie Mae and external developers

Implement learnings from Open Source Communities



Transparency

Open Participation

Governance

Building a proactive solution

- Can your developers push fixes upstream?
- Does the originating community want your fixes?
- Do you have open source-savvy developers?

Even if you have the technical capability, you will need to partner with your internal compliance, legal, and security teams to build an approved solution

Building Partnerships and Getting Approvals

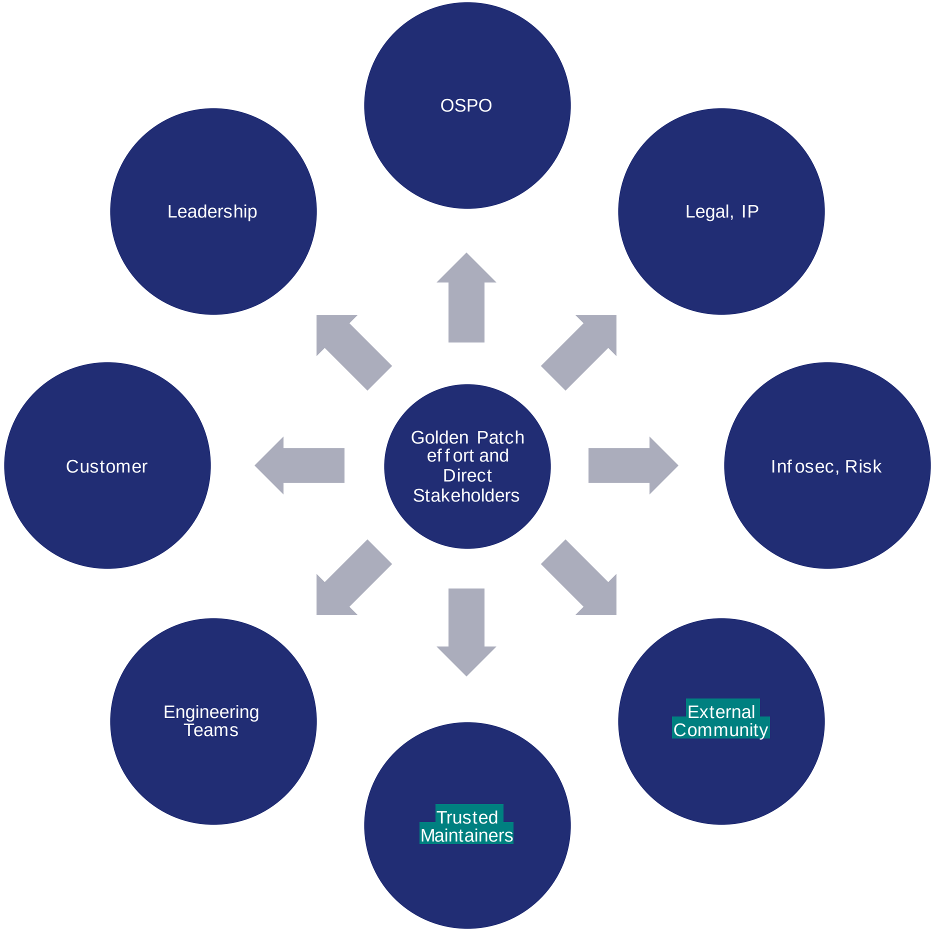
1. Unlock access to external tools
    a. GitHub, Slack, et al.
2. Select group of open source SMEs for critical technologies
    a. They can modify and fix code, as well as educate others
3. Provide access to upstream communities
    a. Can they use regular devices to access upstream tools?
    b. Do you need to build compliance pathways?

Building community

Now that you have access, how do you set up a community for success?

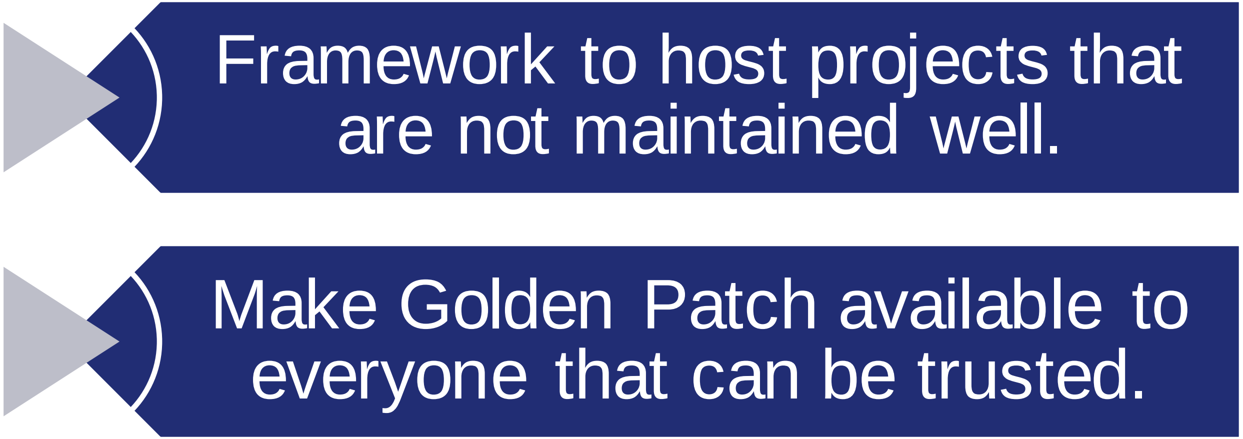
- Establish relationships with upstream partners
- Keep your internal developers engaged
  - Building communities is a new skill to learn
- Establish clear governance rules
  - How to contribute
  - How to fill leadership roles
  - Provide easy ways to share feedback and new ideas

Stakeholder Engagement for External Golden Patches



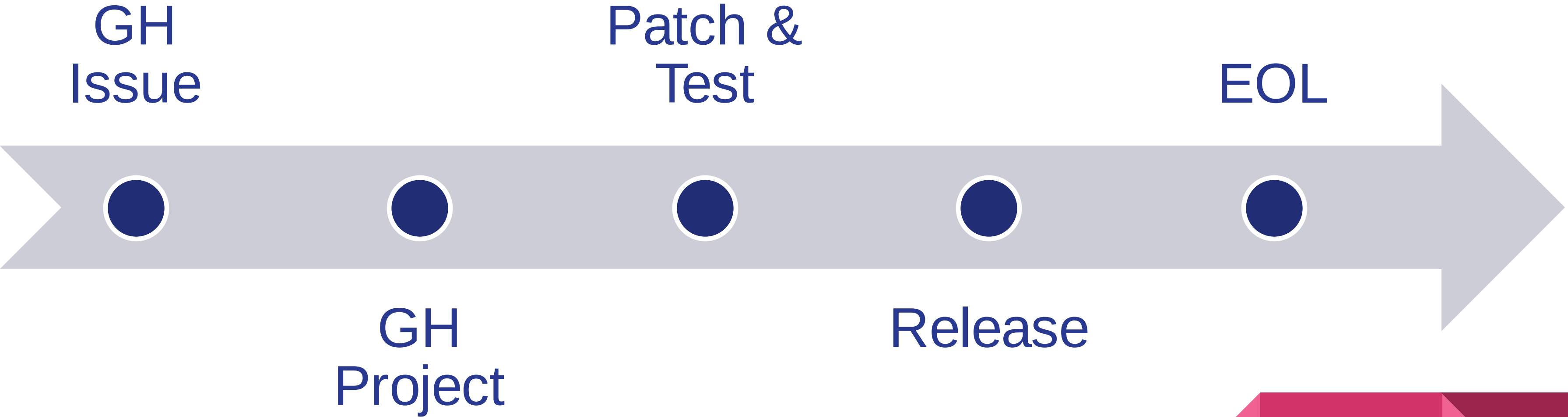
Clean Dependency Project

Objectives of CDP



- Framework to host projects that are not maintained well.
- Make Golden Patch available to everyone that can be trusted.

Clean Dependency Project Lifecycle



GH Issue

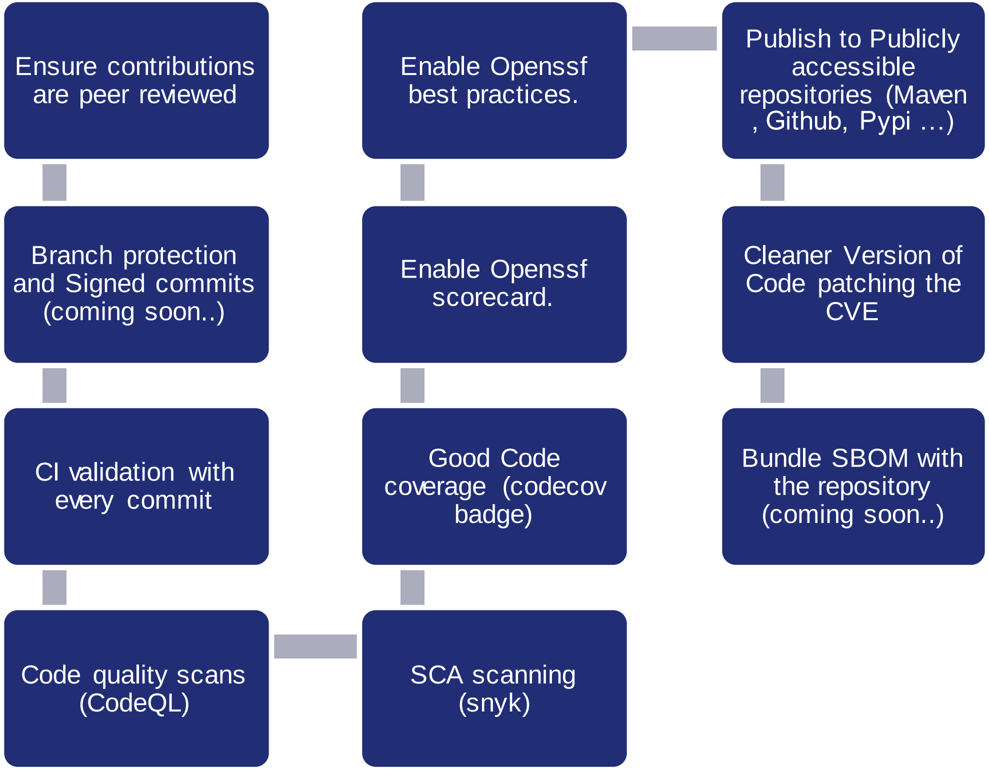
GH Project

Patch & Test

Release

EOL

Project Quality Governance



Ensure contributions are peer reviewed

Branch protection and Signed commits (coming soon..)

CI validation with every commit

Code quality scans (CodeQL)

SCA scanning (snyk)

Good Code coverage (codecov badge)

Enable Openssf scorecard.

Enable Openssf best practices.

Publish to Publicly accessible repositories (Maven , Github, Pypi …)

Cleaner Version of Code patching the CVE

Bundle SBOM with the repository (coming soon..)

continuation

Snakeyaml 1.33

CVE-2022-1471

Severity : 9.8 Critical

NVD Published Date: 05/15/2020

SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization. Deserializing yaml content provided by an attacker can lead to remote code execution. We recommend using SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization. We recommend upgrading to version 2.0 and beyond.

Progression

Fork

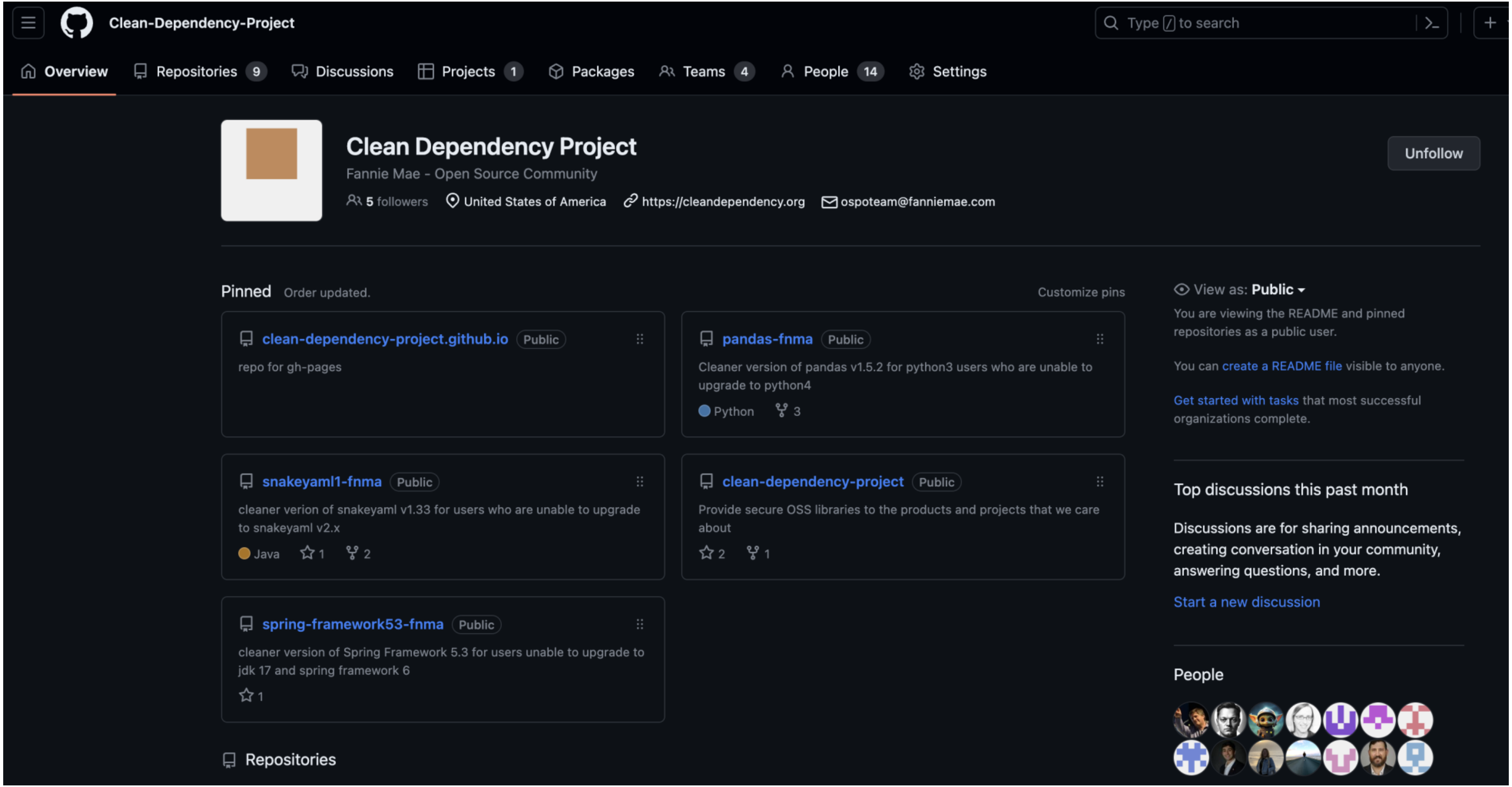
- Fork project under CDP

Patch

- Forced usage of SafeConstructor() in GH

Test

- Distributed patch for Testing



Clean Dependency Project

Fannie Mae - Open Source Community

5 followers   United States of America   https://cleandependency.org   ospoteam@fanniemae.com

Pinned Order updated.

clean-dependency-project.github.io Public

repo for gh-pages

pandas-fnma Public

Cleaner version of pandas v1.5.2 for python3 users who are unable to upgrade to python4

● Python 3

snakeyaml1-fnma Public

cleaner verion of snakeyaml v1.33 for users who are unable to upgrade to snakeyaml v2.x

● Java ☆ 1 2

clean-dependency-project Public

Provide secure OSS libraries to the products and projects that we care about

☆ 2 1

spring-framework53-fnma Public

cleaner version of Spring Framework 5.3 for users unable to upgrade to jdk 17 and spring framework 6

☆ 1

Repositories

View as: Public

You are viewing the README and pinned repositories as a public user.

You can create a README file visible to anyone.

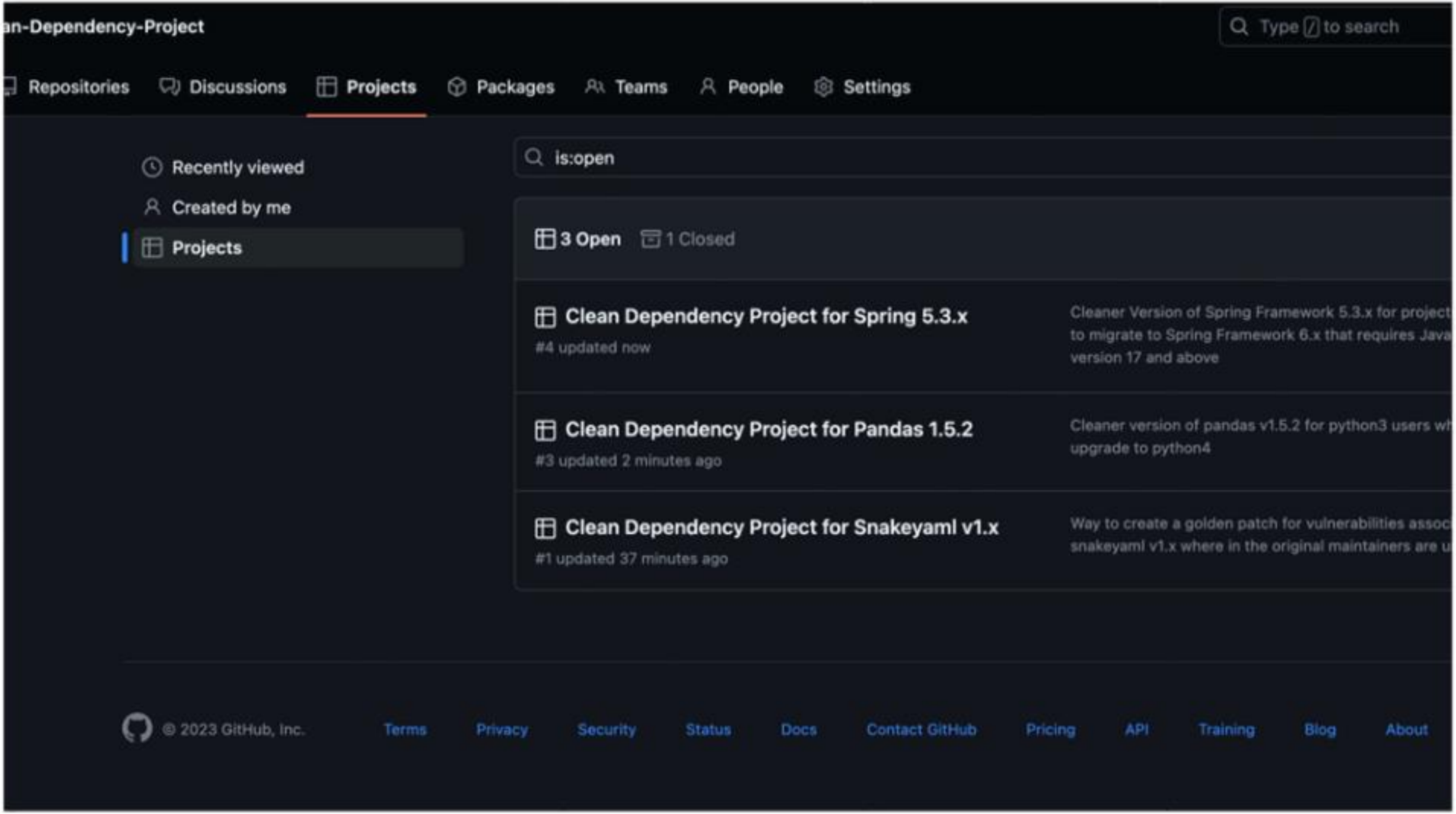
Get started with tasks that most successful organizations complete.

Top discussions this past month

Discussions are for sharing announcements, creating conversation in your community, answering questions, and more.

Start a new discussion

People



Clean Dependency Project for Spring 5.3.x
#4 updated now

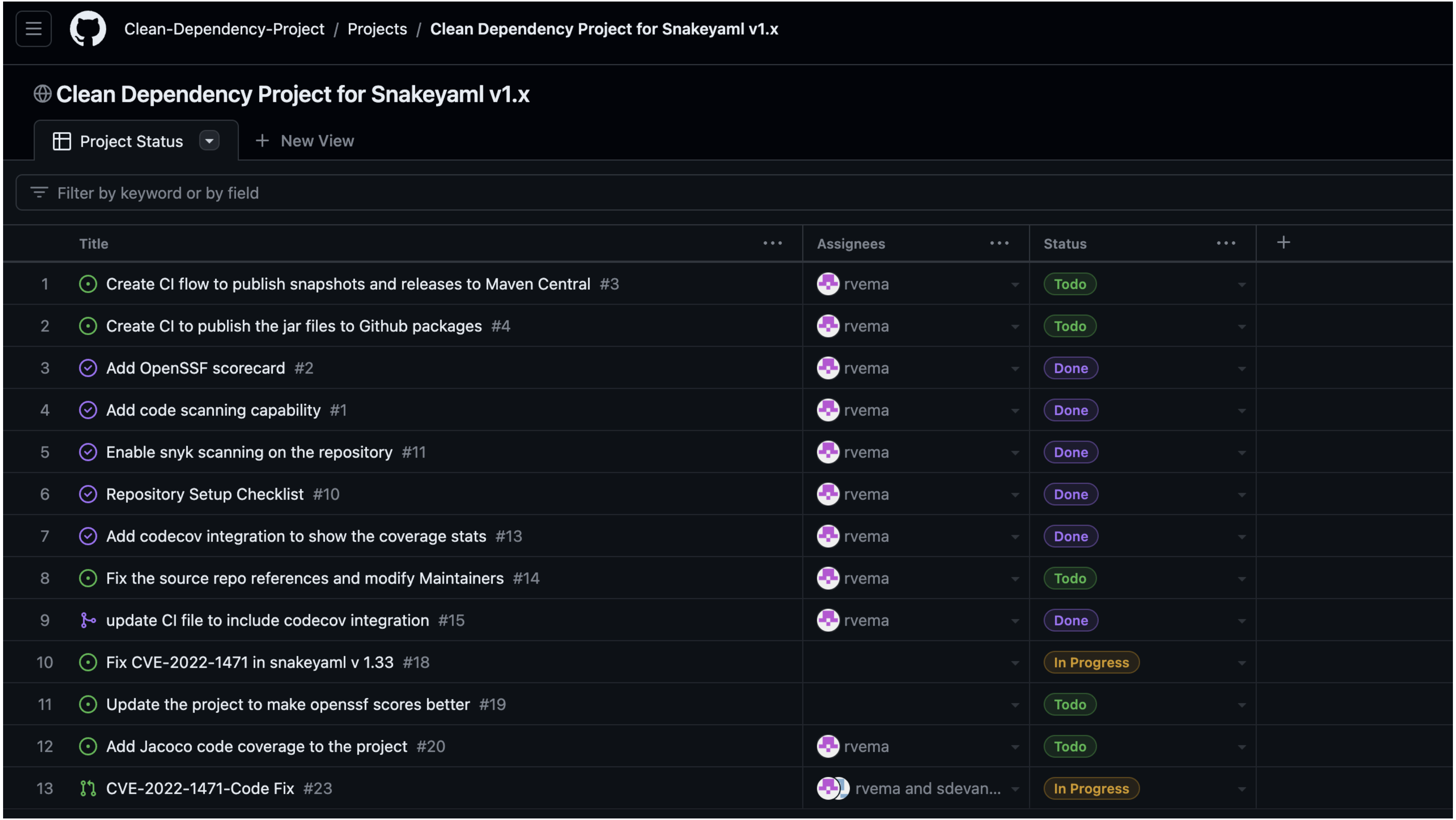
Cleaner Version of Spring Framework 5.3.x for project
to migrate to Spring Framework 6.x that requires Java
version 17 and above

Clean Dependency Project for Pandas 1.5.2
#3 updated 2 minutes ago

Cleaner version of pandas v1.5.2 for python3 users wh
upgrade to python4

Clean Dependency Project for Snakeyaml v1.x
#1 updated 37 minutes ago

Way to create a golden patch for vulnerabilities assoc
snakeyaml v1.x where in the original maintainers are u



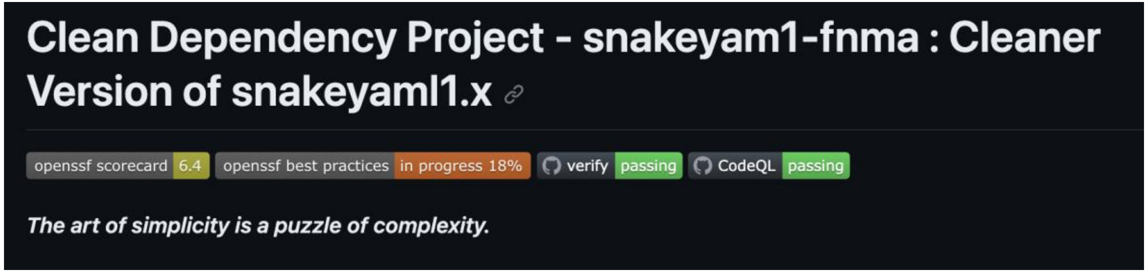
Clean-Dependency-Project / Projects / Clean Dependency Project for Snakeyaml v1.x

Clean Dependency Project for Snakeyaml v1.x

Project Status | + New View

Filter by keyword or by field

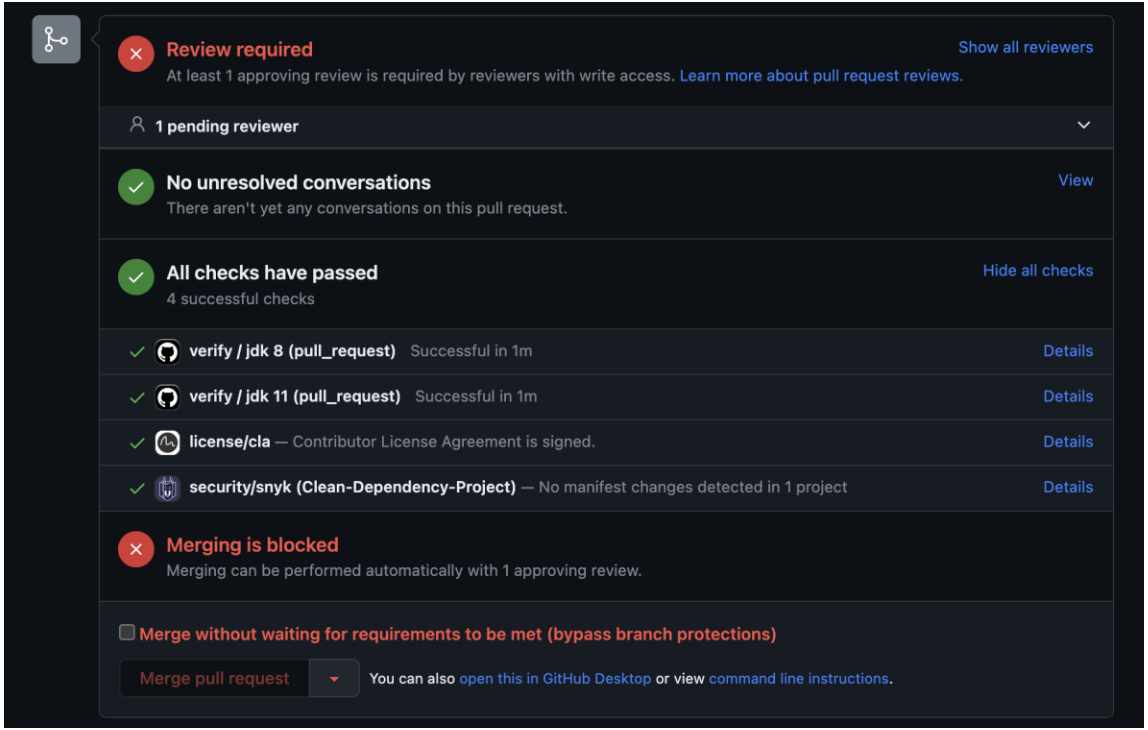
	Title	Assignees	Status
1	Create CI flow to publish snapshots and releases to Maven Central #3	rvema	Todo
2	Create CI to publish the jar files to Github packages #4	rvema	Todo
3	Add OpenSSF scorecard #2	rvema	Done
4	Add code scanning capability #1	rvema	Done
5	Enable snyk scanning on the repository #11	rvema	Done
6	Repository Setup Checklist #10	rvema	Done
7	Add codecov integration to show the coverage stats #13	rvema	Done
8	Fix the source repo references and modify Maintainers #14	rvema	Todo
9	update CI file to include codecov integration #15	rvema	Done
10	Fix CVE-2022-1471 in snakeyaml v 1.33 #18		In Progress
11	Update the project to make openssf scores better #19		Todo
12	Add Jacoco code coverage to the project #20	rvema	Todo
13	CVE-2022-1471-Code Fix #23	rvema and sdevan...	In Progress



Clean Dependency Project - snakeyam1-fnma : Cleaner Version of snakeyaml1.x

openssf scorecard 6.4 | openssf best practices in progress 18% | verify passing | CodeQL passing

The art of simplicity is a puzzle of complexity.



Review required — Show all reviewers

At least 1 approving review is required by reviewers with write access. Learn more about pull request reviews.

1 pending reviewer

No unresolved conversations — View

There aren't yet any conversations on this pull request.

All checks have passed — Hide all checks

4 successful checks

- ✓ **verify / jdk 8 (pull_request)** Successful in 1m — Details
- ✓ **verify / jdk 11 (pull_request)** Successful in 1m — Details
- ✓ **license/cla** — Contributor License Agreement is signed. — Details
- ✓ **security/snyk (Clean-Dependency-Project)** — No manifest changes detected in 1 project — Details

Merging is blocked

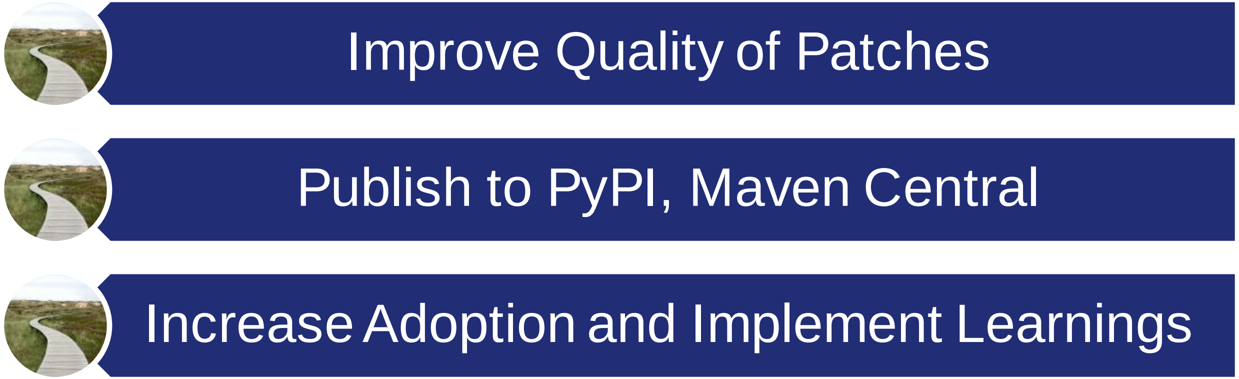
Merging can be performed automatically with 1 approving review.

□ Merge without waiting for requirements to be met (bypass branch protections)

You can also open this in GitHub Desktop or view command line instructions.

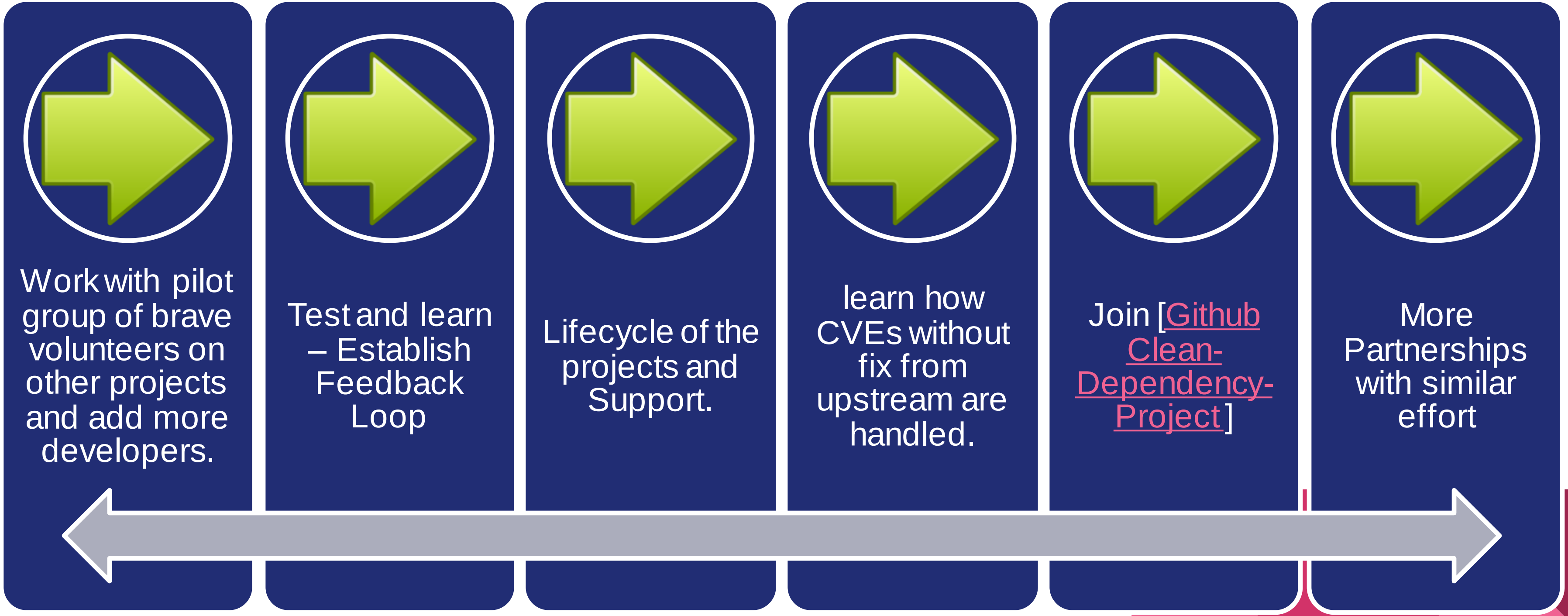
Next Steps for Clean Dependency Project

Next Steps…



- Improve Quality of Patches
- Publish to PyPI, Maven Central
- Increase Adoption and Implement Learnings

Path Forward …..



Work with pilot group of brave volunteers on other projects and add more developers.

Test and learn – Establish Feedback Loop

Lifecycle of the projects and Support.

learn how CVEs without fix from upstream are handled.

Join [Github Clean-Dependency-Project]

More Partnerships with similar effort

Resources and links

- OpenDRI & GeoNode: A Case Study for Institutional Investments in Open Source
- Project Page
- Github Clean-Dependency-Project

Questions?

Thank you!

ragha_vema@fanniemae.com

@rvema @rvema