Using OpenRewrite Recipes for Plugin Modernizatio…

Using OpenRewrite Recipes for Plugin Modernization or Automating Plugin Build Metadata Updates

Project Proposal: https://docs.google.com/document/d/1e1QkprPN6fLpFXk_QqBUQlJhZrAl9RvXbOXOiJ-gAuY/edit?usp=sharing
GitHub Repository: https://github.com/jenkinsci/plugin-modernizer-tool
Project: https://github.com/orgs/jenkinsci/projects/30
Slack Channel: https://cdeliveryfdn.slack.com/archives/C071YTZ807N
Meeting link: http://meet.google.com/oyq-tbpg-aaa

Meeting #1
Tuesday, May 14, 2024

Attendees: Sridhar, Valentin, Bruno, Rajiv

- We discussed the project and the workflow we are going to follow.
- We discussed having a document where we will be updating notes related to this project.
- Sridhar shared about his unavailability due to his exams.
- Discussion on updating the readme and references to add to the repository.

Meeting #2
Tuesday, May 21, 2024

Attendees: Sridhar, Valentin, Bruno, Rajiv

- We discussed the existing action items that have been completed, and the action items we need to do
- Sridhar will have to fork the existing repo before merging the PR
  - Usually, that's the way we proceed with GSoC contributors
  - They will gain full access to the jenkinsci repo at the end of the project
- What is the status and use of the Google Docs file and the GItHub discussions?
  - Let's use GitHub for the … discussions
  - Let's use Google Docs for the final decisions, the points we agreed on, and so on
- Tim from openRewrite chimed in on the Slack channel
  - We'll stay on the open-source part of the project, we won't use moderne.io, unless they promise to give lifetime access to the Jenkins project
- Sridhar will be done with the exams on the 28th of May.

Meeting #3
Thursday, May 30, 2024

Attendees: Sridhar, Valentin, Bruno, Rajiv

- Sridhar divided his proposal into modules and created a roadmap with the modules in the GitHub project linked to the GitHub repository
    - One is for the core
    - One is for the CLI
    - One is for openrewrite recipes
- Should we use a specific module to extract the metadata or do it in the core module?
    - Valentin thinks we'll see that later on when we have implemented something
    - Sridhar should create a fourth module to do so
- Sridhar will code from tomorrow on
- What should we use to implement the comémand line? PicoCLI or other tools already used in Jenkins?
    - Valentin is ok to start with picoCLI.
- Sridhar is now on holiday, so he will be able to work more these days
- Sridhar wants to use TDD
- Sridhar was wondering if using shell script is mandatory or recommended
    - Valentin would prefer using only Java/OpenRewrite if that's possible but isn't against using Shell scripts if mandatory

---

Meeting #4
Tuesday, June 4, 2024

Attendees: Sridhar, Valentin, Bruno

Notes:

- Sridhar created a PR with test cases to parse the CLI
- No blocking point for the moment
- Some repos don't have Java but Groovy
    - Sridhar is trying to find a way to distinguish between them
- Some plugins aren't part of the Jenkinsci organization, we should just ignore them for the time being
- It is the same for plugins that don't use Java (Ruby, Groovy, etc…) or use gradle.
- Valentin created the files and processes necessary to start the CD part (automatic release)
- Sridhar started to work on the first module, and looking to find important things that can be automated.

Action points:

☑ ~~Sridhar will try to complete the second module by the end of next week.~~

Meeting #5
Tuesday, June 11, 2024

Attendees: Sridhar, Valentin, Bruno

Notes:

- We can rely on updatecenter to get the list of actual published plugins
- We need to get the pom files for the plugins
  - Clone or check out
  - Why do we use a custom parser?
    - Openrewrite can read and parse the pom.xml. It will create the tree, we can create some visitors and get some information
    - Here is an example.
    - This will be much more reliable
    - Pom.xml is pretty complex, with inheritance for example, and transitive dependencies so doing it by hand is a no-go
- Storing in a database is out of the scope for the time being
  - Let's store it locally for now, like in a file system cache.
- What about the update center updates?
  - Let's get the file, caching it
  - If we see a difference from time to time, then we can invalidate the cache and make new checks
  - What will the changes look like? New plugins? They should already be up to date. Old plugins with new changes? Why not, but what kind of changes? Time will tell.
- Developer sections are not used anymore on Jenkins pom.xml.
  - In fact that even could be a recipe to clean such pom
- The recipes can be used in read-only mode, to grab the metadata, without transforming the existing code (like with this one).
- Sridhar will work on the core module in the coming week

Action items

☑ ~~Sridhar will have to move to openRewrite pom parsing~~

☑ ~~Sridhar should forget about the database for the time being, and store metadata locally~~

☑ ~~Valentin will find an example of pom parsing with openRewrite. The example is about checking a pom.xml file is a Jenkins pom.xml~~

Meeting #6
Tuesday, June 18, 2024

Attendees: Sridhar, Valentin, Bruno

Notes:

- Sridhar added a new way to parse the pom.xml, thanks to the walker.
- Valentin tested the rewrite maven plugin after Tim from Openrewrite suggested it
- We should use the Maven Invoker on the plugins we want to modernize
- Sridhar was blocked on Sunday, so Valentin and Sridhar had a call to unblock him
- Let’s reuse what is already there (Jenkins Openrewrite from Steve Hill)
    - Create other meta recipes
    - Reuse the existing recipes as is
-

Action items:

- ☐ Contact Steve Hill who worked on the previous recipes for Jenkins

---

Meeting #7
Tuesday, June 25, 2024

Attendees: Sridhar, Valentin, Bruno

Notes:

- Next week, Valentin will be off, so he won’t be able to join the two office hours
    - Bruno will be able to take the notes
- The slide deck is at: …
    - Sridhar would like to make a demo during the presentation
- Sridhar: no plan for next week, for this week, improvement of the tool
- How to automate the cloning
    - With jgit?
        - How to get the same version of jgit as the one used in the git client plugin?
            - Updatecli manifest could help, but that’s a lot of effort for not so much gain
        - Git client plugin as a bom, with dependency management pom scope
            - This might work
        - We could start with a pinned version of the jgit product
- Sridhar should get some reviews from other mentors when Valentin will be off

Action items:

- ☑ ~~Sridhar: should create a few slides~~
- ☑ ~~Sridhar should give us a blank presentation before the 11th (by this Friday would be better, next week would be without Valentin, and between the 7th and the 11th is kind of late, but would work too as Valentin will be back)~~
- ☑ ~~Bruno should get in touch with Alyssa to get the link to the right slide deck~~
- ☑ ~~Sridhar should add some more coverage~~

Meeting #8
Tuesday, July 2, 2024

Attendees: Sridhar, Bruno

Notes:

- Sridhar will send an email to the Jenkins' developers Google Group to get some feedback about the #41 issue (Save metadata to file system)
  - If we don't get much feedback from the list, we could open a helpdesk ticket
  - Then, we could ask the question to community.jenkins.io
- Sridhar will create the PR/work on the jgit/add a file to repo after the midterm evaluation
- Sridhar will also work on the metadata extraction

Action items:

☑ ~~Sridhar should give us a blank presentation before the 11th (by this Friday would be better, next week would be without Valentin, and between the 7th and the 11th is kind of late, but would work too as Valentin will be back)~~

☑ ~~Sridhar will schedule a meeting to do a blank presentation~~

Meeting #9
Tuesday, July 9, 2024

Attendees: Sridhar, Valentin, Bruno

Notes:

- Release of the tool to perform (we need to check if it works correctly because it's different from a Jenkins plugin)
  - This should be done after the mid-term presentation
- Sridhar will give us a rehearsal presentation today
  - The time is good (8 minutes), but it could be a bit longer
- Gitpod fails for the time being because of the JDK11 installed by default
  - Bruno could prepare a PR to solve this
- Sridhar will work on integrating git/github after the mid-term evaluation

Action items:

☑ ~~Sridhar should find a way to speed things up when it comes to running the demo: https://gitpod.io/?autostart=true#https://github.com/jenkinsci/plugin-modernizer-tool could help~~

Meeting #10
Tuesday, July 16, 2024

Attendees: Sridhar, Valentin, Bruno

Notes:

- Releases:
    - Having the first release was not a piece of cake. RPU is difficult to use with multi-module projects.
    - Valentin disabled the GitHub action so every pull request does not create a new release.
- Good presentation last week
    - Demo and tools went well
    - Mark enjoyed the demo and the tool
- Would it make sense to compile at least the resulting code after recipes are applied?
    - Maybe, we'll think of it
- Could we also generate a better title for the PRs, or let the user choose the PR title?
    - Could it be linked to the recipes title by default
    - Could we configure the PR title by an option on the command line or an environment variable?
- We should add more recipes by next week, and improve the rest afterwards
    - More recipes on the code changes would be nice
- OpenRewrite has a new version. Will it break anything?
- Does anyone have the history for the hub4j GitHub API repo? It looks like it was created by a person named Koshuke. Would that be "our" Koshuke?

Action items:

- □ Sridhar should focus on adding more recipes this week
- ☑ ~~We should also think of storing the metadata somewhere~~
- ☑ ~~Bruno should ask Mark about the hub4j GitHub API repo~~

Meeting #11
Tuesday, July 23, 2024

Attendees: Sridhar, Bruno

Notes:

- Tried adding more recipes, but the plugins aren't compiling with the current recipes.

- Would it make sense to compile at least the resulting code after recipes are applied?
    - Some old plugins will need to be compiled with JDK8 before doing anything else.
        - Shouldn't we first apply a list of recipes that would help them migrate to JDK11? We can't check by compiling if the plugin is working to begin with.
- The CSV files generated are useful for a human being, not so much for a process wanting to process them later on.
    - Sridhar proposes to use the JSON format instead
- Could we also generate a better title for the PRs, or let the user choose the PR title?
    - Could it be linked to the recipes title by default
    - Could we configure the PR title by an option on the command line or an environment variable?
- We should add more recipes by next week, and improve the rest afterwards
    - More recipes on the code changes would be nice
- OpenRewrite has a new version. Will it break anything?
- To enter new recipes, we could use the openrewrite documentation to automate the modification of the file. For the time being, it's a human process.
- Classified the plugins from the SDK used, and found 169 using Java 7, and 460 using Java 8. We should try to find a set of recipes in order to help these plugins migrate to JDK8 first, then JDK11 later on. Bruno may share its insights on the set of recipes he found 8 months ago.
    - Cookbook is not yet implemented, so will have to do it by hand
    - Openrewrite supplies something that is kind of similar, we could bundle it with our jar
- Sridhar will attempt to use composite recipes
- We're hoping Valentin is okay and will be back soon. Bruno will do his best to review upcoming PRs.

Action items:

☐ Sridhar should focus on adding more recipes this week again

☑ ~~Sridhar should create an issue about the fact we're creating PRs for recipes that did not work (thus creating an empty PR)~~

☐ Sridhar should try to solve this by next week

Meeting #12
Tuesday, July 30, 2024

Attendees: Sridhar, Valentin

Notes:

- Valentin asked about the extended period of the project
    - Continuing until November or finishing within the standard period?

- We switched from checkstyle plugin to spotless (from the parent pom) for code formatting and style. This is similar to other plugins and tools on the Jenkins ecosystem
- Switch from the custom YAML format to OpenRewrite declarative recipes
- We have some minor issue for plugin using JDK 8, we cannot run the recipes compiled on JDK 17
    - Idea is to the the setJavaHome for compile with JDK 8 and JDK 17 when we run the recipes
    - OpenRewrite document this by changing the JDK when running from the IDE but we are using the maven rewrite plugin called by the maven invoker
- Possible fix for JGit to closed correctly https://github.com/jenkinsci/plugin-modernizer-tool/pull/134
    - Valentin tried to run some recipes across ~30 plugin he maintains

Action items:

☑ ~~Sridhar needs to raise a PR which caches JDKs locally~~

☑ ~~Sridhar should focus on switching JDKs while modernizing the plugins~~

Meeting #13
Tuesday, August 6, 2024

Attendees: Sridhar, Valentin,

Notes:

- Valentin ask about the extended period
    - We will keep the standard coding period due to availability and continue with normal OSS contributions on the tool
- 2 points
    - Store the metadata about the plugins somewhere
    - Extract the JDK version for the plugin instead of providing this information by user
- Refactoring code recipes, more important to baseline existing plugin (archetype, build with recent JDKs, bom, dependencies etc…)
- Small issue found on Linux system when the JDK is extracted (execution flag +x is lost on binaries). JDK is not usable by the tool

Action Points:

☐ Sridhar needs to extract plugin metadata to csv or json file.

☑ ~~Sridhar needs to add execution permission after extraction~~

Meeting #14
Tuesday, August 13, 2024

Attendees: Sridhar, Valentin

Notes:

- Still, metadata is not yet stored in a common place
- Sridhar has holidays and will be able to work more hours in the next 2 weeks

We talk about the metadata storage and we have some open point

- We suggested an external database (like MongoDB)
    - Where to store it?
    - What about security? Who can access and populate the data?
- What is the relationship between the MetadataExtractor and the transformation recipes
    - Is the extraction must be done before (by some system on jenkinsci infra?)
    - Is it something done on the developer machine
- 2 weeks remaining. Working on a feature with database and jenkins-infra seems a bit big feature. Perhaps is better to just work on some poc and small enhancement and/or bug fixes

Action Points:

- ☐ Stabilize the tool, fix latest issues, test coverage and minor improvement

Meeting #15
Tuesday, August 20, 2024

Attendees: Sridhar, Valentin, Bruno

Notes:

- We have to store the metadata somewhere.
    - Could we store that in the existing repository for the time being in another branch?
    - We could also create another repository to do so
    - Later on, we could ask the Jenkins infrastructure team to create a database, or something else
    - There is a specific command that creates the metadata. It is a recipe.
- Could we use a specific tag on the recipes, so we know which recipes will change the JDK target?

Action Points:

- □ Try to close as many issues as possible, without creating any new feature ;-)

Tab 2