

Forest Service ePermit API

Project Summary

May 31, 2017



Agenda

- Project Review
- Retrospective

Sprint 1

Theme

Setup and documentation

- Completed scaffolding and DevOps tasks and Swagger OpenAPI implementation.
- 113 points, 37 issues

Sprint 1 Retrospective

What went well during the sprint?

- Scaffolding - set up everything for testing pretty quickly
- Communication - prompt response from stakeholders
- Badges
- Got routes defined quickly
- Feedback from Laura on first two pull requests
- Scheduling
- Mid sprint pull requests
- Informal feedback

What didn't go so well during the sprint?

- Have basic API for response
- Acceptance criteria - delayed

What can we do to improve?

- Have acceptance criteria ready for sprint planning.

Sprint 2

Theme

Authentication and schema work

- Completed authentication and validation.
- 84 points, 28 issues

Sprint 2 Retrospective

What went well during the sprint?

- Anticipated LOE well, completed good number of stories.
- Code quality is improving, adding linting was helpful.
- Doing a good job working iteratively.
- Feedback in PR was very helpful.
- Communication overall was strong.
- Schema-based approach to validation is a positive

What didn't go so well during the sprint?

- Size of pull request can making reviewing difficult.
- Burndown chart doesn't accurately reflect velocity.

What can we do to improve?

- Consider pulling to master multiple times through sprint; chunk pull requests to indicate the focus of work.
- Build on effectiveness in addressing PR comments and other feedback.
- Look into changing GitHub settings for burndown chart.

Sprint 3

Theme

Handling non-SUDS data

- Completed database implementation and file upload implementation, and made sprint 2 work schema-based.
- 96 points, 20 issues

Sprint 3 Retrospective

Sprint 3 Retrospective - 4 L's

Forest Service ePermit API ☆ ⓘ Public

Liked

- ORM - using migrations and seeders
- Using a ORM tool for database vs traditional method
- The migrations
- Productive backlog grooming session
- Planning poker!
- The planning poker
- seeing the more complete requests come through the swagger
- that swagger can do file uploads
- All the schemas!!
- The schema
- Pair programming
- Add a card...

Learned

- ORM with sequelize and file uploads in swagger
- a bit more about orms in node
- Sequelize needs better documentation
- Uploading files and json in same request
- Multer versioning issues
- manually populating apis is a bit tedious
- learned that keeping a million different schemas is probably not the best
- Don't view .docx files in Apple Pages
- Factories make life easier
- break down tasks into smaller chunks
- Trello works well for retros
- Backlog grooming and retro activities are effective

Lacked

- Basic API access
- NCI to have direct access to the basic api
- a single endpoint to the basic api to create an application
- the ability to pull in the commits because of the password has issues
- documentation on the features as they are built
- Clear documentation of some packages used.
- Access to end users (API developers)
- Add a card...

Longed For

- intake module developers
- basic api access
- ability to save website field directly in SUDS
- online whiteboard tool?
- API mocking tool
- schemas!
- Clear documentation of some packages used.
- time (eg time to contribute to other OS projects)
- documentation needs
- eAuth documentation
- the ability to spell
- Add a card...

Actions

- Documentation - Use JSDoc for code readability; documentation created as features built
- Add more testing - validation
- Look into mural.ly as a whiteboard tool
- Add a card...

Sprint 4

Theme

Integrate database with file uploads and generate routes based off of schema; Incorporate recent changes to schema mapping.

- 113 points, 26 issues

Sprint 4 Retrospective

Sprint 4 Retrospective - 3 L's

Forest Service ePermit API ☆ 🌐 Public

Liked

- Mock API - simple middleware
- Improved linting
- That the product is down to 2 schemas!
- how great swagger makes demos
- new schema based routing
- simple Mock API routing within middlelayer API
- The validation is mostly complete!
- Collaboration
- Process documentation
- Ease of adding new fields, files, validation.
- the files are handled
- Pair programming to solve problems
- JSDocs output
- improvement through iteration

Add a card...

Learned

- about how the FS security status quo can inhibit iterative development
- how to integrate postgres in travis
- Need to estimate better - include documentation & refactoring time in estimates
- Proper story points
- Documentation illustrates process improvements
- Much easier to document as you go rather than after the fact
- Need to do JSDocs upfront
- High velocity = technical debt

Add a card...

Longed For

- Get you all basic api access!
- earlier addition of PR issues in backlog
- earlier check-in for backlog updates
- More time spent prepping for planning
- Basic API
- more time for planning poker
- More process documentation

Add a card...

Sprint 5

Theme

Process and code documentation

- 67 points, 28 issues

Sprint 5 Retrospective

Sprint 5 Retrospective - 3L's

☆ 🌐 Public

Liked

- JSDocs
- all the jsdocing
- no more jsdocs in code
- collaboration on docs
- readme docs - documentation in the repo
- moxail
- cleaned up repo
- that it is an easy app to dockerize
- coordination with Flexion - they like your swagger
- intake developers on board
- better ZenHub/backlog organization
- good to have users (intake devs)

Add a card...

Learned

- more about packaging express middleware for npm
- good feedback from Flexion
- How scope for dotenv works
- how to fix/avoid linting issues
- How credentials for AWS work
- a bit more about docker

Add a card...

Longed For

- update tasks/issues more often
- A bit better git flow - sprint to sprint branch
- Those documentation changes by tomorrow morning so another dev can test out building out a new permit type tomorrow morning
- A passing build
- More ideas for backlog/icebox

Add a card...

Sprint 6

Theme

Update authorization and documentation

- 70 points, 21 issues

Sprint 6 Retrospective

Start, Stop, Continue - Sprint 6 Retro

☆ 🌐 Public

Rules

In this activity, participants should identify 1) potential additions to the process that might improve efficiency, 2) things the team can stop doing, because they're unnecessary or inefficient, and 3) positive behaviors from this sprint that should become habits.

1. Begin by adding items to the Start list. Everyone takes 2 - 3 minutes to add cards.
2. Take a minute to vote on the items you think are most important. Vote as many times as you like. To vote, click on the card (not the edit icon) and select the vote button under Actions.
3. Discuss as a group the items that received the most votes.
4. Repeat Steps 1 and 2 for the Stop list and Continue list.

Add a card...

Start

Test Driven Development (TDD)
👍 1

setting regular testing meetings with flexion for the remainder of the project to make sure refactors and stuff are cool with them
👍 5

thinking about accessibility/508

focusing on transition plan
👍 2

thinking about how we can focus the next spring on flexible debugging tasks to iterate quickly on debugging the connection to the basic api
👍 1

Add a card...

Stop

more detailed documentation.

Filling sprints to the brim in sprint planning, not allowing time to address PRs, hotfixes.
👍 3

allowing too little time to add bug fixes/PR-related issues
👍 1

seems like sprint velocity has ticked down? this is more of a discussion point than a "stop"
👍 1

Add a card...

Continue

Keeping communication with Flexion going
👍 3

pinging me if you have js questions
👍 2

Figuring out CI/CD pipeline how to minimize env var
👍 1

new feature voting

working around individual schedules

Add a card...

Sprint 7

Theme

Successful permit application through Basic API

- 59 points, 27 issues

Sprint 7 Retrospective

Start, Stop, Continue - Sprint 7 Retro

Directions

In this activity, participants should identify 1) potential additions to the process that might improve efficiency, 2) things the team can stop doing, because they're unnecessary or inefficient, and 3) positive behaviors from this sprint that should become habits.

1. Begin by adding items to the Start list. Everyone takes 2 - 3 minutes to add cards.
2. Take a minute to vote on the items you think are most important. Vote as many times as you like. To vote, click on the card (not the edit icon) and select the vote button under Actions.
3. Discuss as a group the items that received the most votes.
4. Repeat Steps 1 and 2 for the Stop list and Continue list.

Add a card...

Start

Testing earlier to highlight field discrepancies

Getting on proxy stuff earlier so that could have implemented it before testing/ when port opening

Final code review

"Usability test" the schema spec

Schema spec

Transition activities

Transition :(

Release 1.0 :)

Do a project-length retro before the project wraps up

Add a card...

Stop

New features

Add a card...

Continue

Being highly communicative and responsive (both on TTS and NCI side)

Keeping lines of communication open

the flexibility of the validation.json

debugging (cuz its kinda fun)

documenting how this thing works - i.e. the spec

Update documentation

Update dependencies

Working collaboratively

Improve code coverage

quick responses across teams

Improve code quality

Using swagger to have demonstrable demos

Add a card...

Sprint 8

Theme

Finish strong!

- 40 points, 15 issues

Project Retrospective

S.C.I.T

- **Successes**
- **Challenges**
- **Insights and Take-Aways**

S.C.I.T

Successes

- What was the most gratifying or professionally satisfying part of the project?
- What did we do well?
- What helps us to be successful as a team?
- Which of our methods or processes worked particularly well?
- Where did we get lucky?

Challenges

- What was the biggest impediment?
- Which of our methods or processes were difficult or frustrating to use?
- What was painful but necessary?
- What's still keeping you awake at night? What did we leave unresolved?
- What still puzzles us?

Insights and Takeaways

- What advice would you give yourself if you were to go back to the start of the project?
- What should we have learned from this project a year from now?
- What else could we do better next time?

