

SPRINT RETROSPECTIVE TOTAL BRIECALL





- Purpose of Retrospective
- Gather Data
 - Quick look back on our performance
 - Gather new feedback directly from team
- Review Feedback & Improvement Selection
 - Determine improvement/change to implement
- Wrap up



- Maintain focus on continuous team improvement
- Help answer these questions:
 - What worked well?
 - What did not work for us?
 - What can we improve going forward?
- Help address challenges and triumphs that are fresh in our minds
- Help identify an improvement or change to implement for subsequent cycles



SPRINT PERFORMANCE



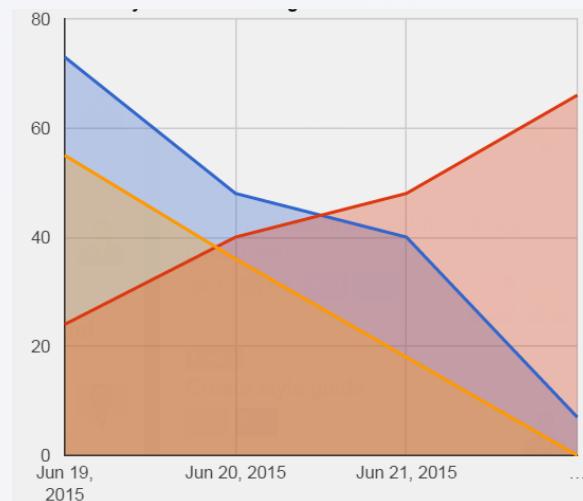
Key Achievements

- Setup Agile project tools (Trello, Slack, Github, etc.)
- Hosted user interview session and collected user stories and feedback
- Hosted sprint planning meeting with team to review user stories in Trello and plan out work for first, second and third iterations (estimated hours and assigned task owners)
- Finalized solution design documentation in Github, as well as configuration management and continuous monitoring documentation
- Finalized user interface mockups using InVision and hosted design review meeting with end users
- Collected end user feedback on mockups via InVision tool
- Coordinated three rounds of usability testing with end users; collected feedback for inclusion in next iteration
- Finalized API documentation
- Finalized installation instructions
- Created application icon and favicon
- Finalized application unit tests
- Deployed final version of the prototype application for usability testing

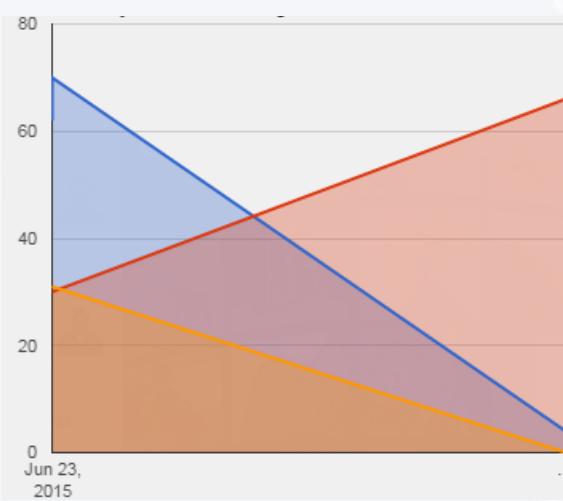




Iteration #1



Iteration #2



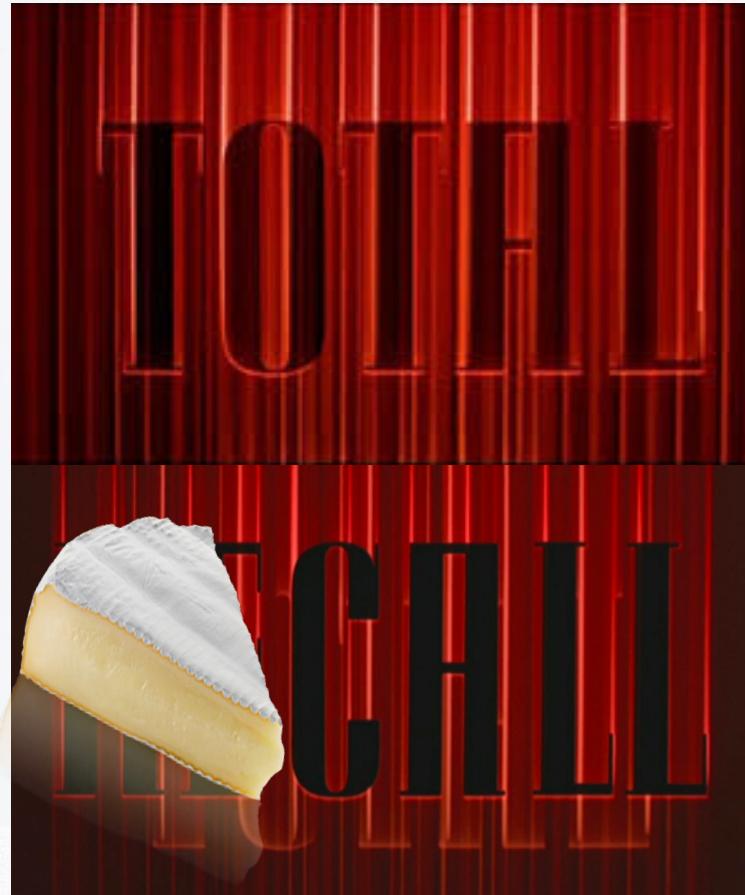
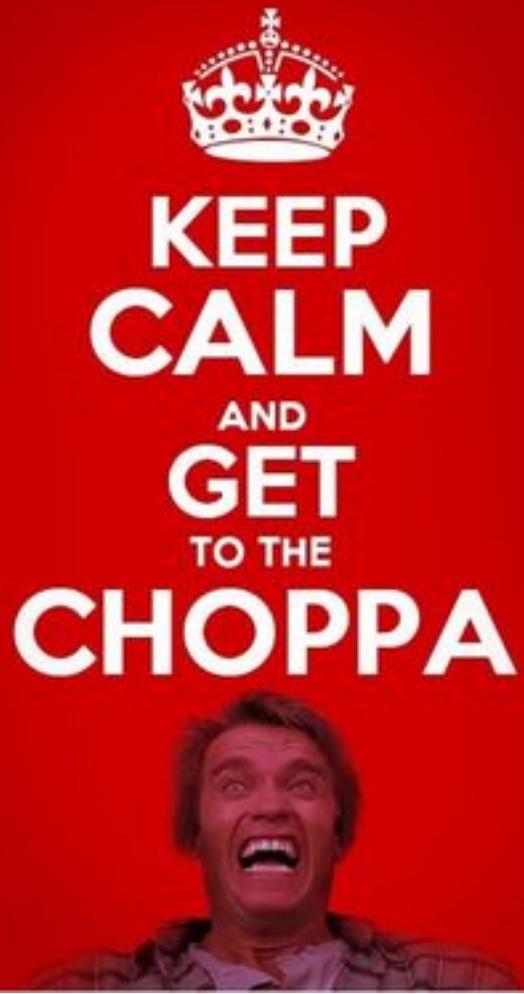
Iteration #3



Sprint burndown percentage averaged over ~96% completion rate for last 3 sprints (estimated vs. completed)

Let's eat
some brie to
celebrate!





GATHER
DATA

Retrospective Quadrant

Ground rules:

1. Write at least 1 sentence on each post-it note. Must be relevant to the sprints, tools or process used for the 18F RFQ response
2. Quadrants include: Things you like, Things you dislike, Ideas we should try and Team shout-outs
3. Post the completed post-it note in the applicable quadrant on the whiteboard



Like



Dislike



Ideas we
should try



Appreciations,
Shout-outs

REVIEW FEEDBACK





Like:

- Exploring available open source tools to enhance sprint development process - including Travis CI, Code Climate, etc.
- Team communication in Slack, including team chats, build/unit test notifications
- Adding automated unit tests to minimize failures
- Use of code quality tools like Code Climate
- Heroku/build environment setup helped streamline build process

Dislike:

- Trying to get API docs to generate across multiple platforms, lots of trial and error
- Effort spent on additional features that ended up being de-scoped
- Sample dataset and API limited in scope
- Requirements for RFQ high-level and vague

Ideas we should try:

- Train/host brown bag sessions with team members on recently used open source tools
- Develop wiki/training documents for new tools/concepts
- Conduct assessment on new open source tools to determine complexity / ease of implementation with the rest of the development staff
- Implement Istanbul code coverage tool for other internal projects

Shout-outs, Appreciations:

- All team members should be complimented on their hard work over the last three sprint cycles to complete this effort

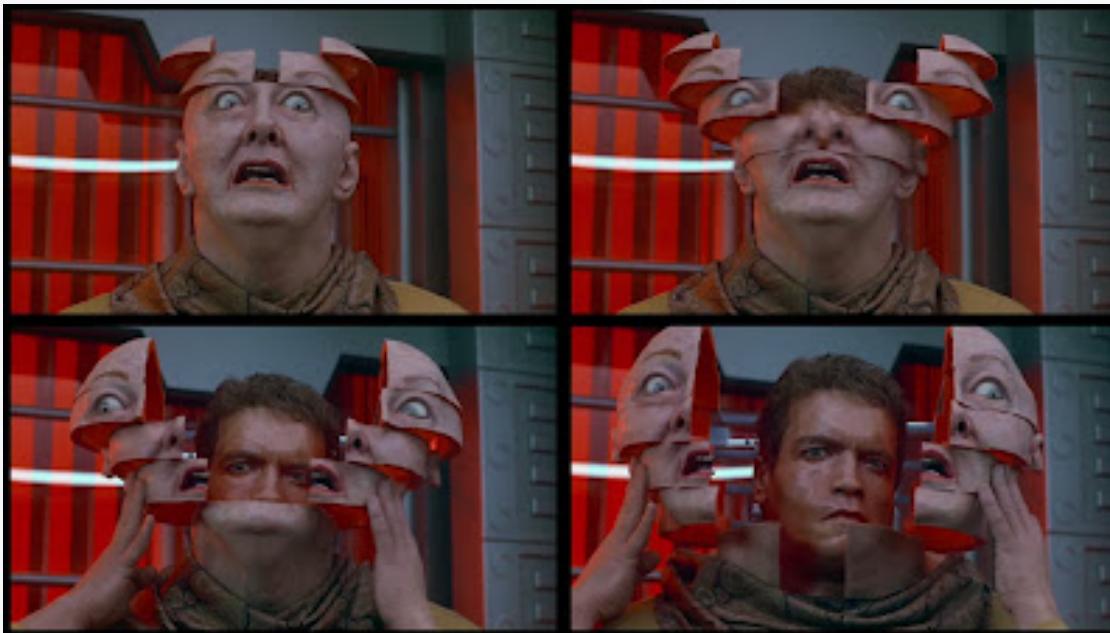
- Conducted polling on at least 1 change to include in our sprints going forward
 - Determined how team will measure the success/failure of the change



- Implement the 'Istanbul' code coverage tool for other internal projects
- Measure of success: Tool implemented and running for at least one of our internal projects in the next 1-2 sprints



Questions?
Comments?





TeraLogics

