

CHAPTER 1

Methodology

Iris issues and sprints are being tracked using Trello. These are organised using the *kanban* system and custom labels inside of Trello. Trello allows filtering of cards by their labels, meaning issues related to a particular sprint can be the only cards visible. This makes tracking sprints and issues for the sprint easy to manage.

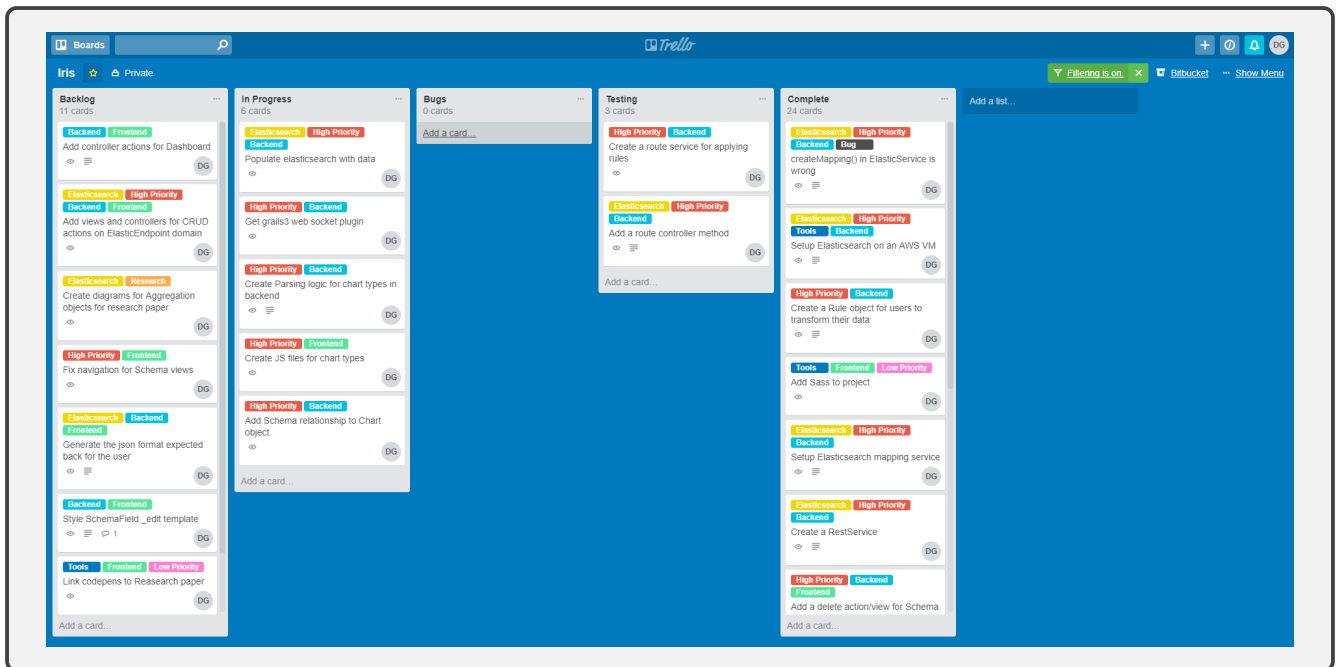


Figure 1.1 – Screenshot of the Iris Trello board.

1.1 Trello

Trello was used to keep track of issues and sprints in Iris. The sprint numbers correspond to the version of Iris at the time of the sprint. Below is a list of the issues for each sprint of Iris. This list is generated using a python script written by the author which interacts with the Trello API to retrieve the issues from the Trello database based on their sprint numbers.

Sprint 0.0.1

- Scope project with Onaware
- Draw ER Diagram for User owning Dashboards
- Research d3 wrapper libraries
- Draw diagrams for Frontend
- Setup rails 3.3.1 + groovy 2.4.11 + java 8u144 on devices
- Research Dashboard libraries for serialization
- Generate Domain classes based off UML diagram
- Look into Elasticsearch Java api and see if it can be used for building complex aggregations dynamically
- Use Grails to generate all controllers and views for Domains
- Add Spring Security Plugin for User Domain
- Build Elasticsearch Objects
- Update to latest version of jQuery
- Update to Bootstrap 4
- Add controller actions for Schema Domain

Sprint 0.0.2

- Style SchemaField edit template
- Add views and controllers for CRUD actions on ElasticEndpoint domain
- Generate the json format expected back for the user
- Fix navigation for Schema views
- Add levels to javascript aggregation logic
- Setup Selenide on all machines
- Create a route service for applying rules
- Add a route controller method

-
- Populate elasticsearch with data
 - Get rails3 web socket plugin
 - Link codepens to Research paper
 - createMapping() in ElasticService is wrong
 - Create diagrams for Aggregation objects for research paper
 - Add Schema relationship to Chart object
 - Setup Elasticsearch on an AWS VM
 - Create a Rule object for users to transform their data
 - Add Sass to project
 - Setup Elasticsearch mapping service
 - Create a RestService
 - Add a delete action/view for Schema domain
 - Deleting Elasticsearch index is not working from front end
 - Create ElasticEndpoint domain
 - Create a Domain for storing Elasticsearch endpoint addresses
 - Research updating Elasticsearch mappings
 - Move over to branch 0.0.2
 - Add Logging to app
 - Add controller actions for Dashboard

Sprint 0.0.3

- Display aggregation execution result in playground
- Schema level counter in JS should match a DOM element to keep it more dynamic
- Iris demo presentation
- Write gradle task to kill geckodriver.exe
- Try adding underscore.js to project to stop javascript error
- Make aggregations executable in Aggregation Playground
- Create param link for Schema for routing data
- Create script to generate data for Iris demo
- Add Navbar navigation
- Create Selenide Test for demo

-
- Make a test index in Elasticsearch to test out sockets

Sprint 0.0.4

- Unsubscribe the chart from onChartLoad
- Add download chart image button to each chart widget
- Check if billboard is storing all data points on charts
- Execute Chart aggregation on dashboard load
- Add download chart data button to each chart widget
- Add modal to prompt user for a revision comment when they click update on dashboard
- Add a dashboard to the bootstrap file
- Add CRUD functionality to widgets on Dashboard
- Add update functionality for Dashboards
- Create XL Modal for dashboards
- Set and resize chart to be the same as the widget area
- Create JS files for chart types
- Dashboard isRendering flag not working in some cases
- Add date by default to all json objects going into Elasticsearch
- Create Parsing logic for chart types in backend
- Load a saved dashboard client side
- Limit socket messages to dashboards which are currently rendering
- Save dashboard server side
- Use browser cache to store a widget's aggregation attribute
- Use data- HTML attribute for storing widget information
- billboard.js charts are not displaying correctly
- Need to add chart Id to subscription charts

Sprint 0.0.5

- Add StateChartList to Dashboards
- Remove all the Grails/Groovy aggregation builder code to reduce code and war size
- Add titles to charts
- Add StateChartDisk to Dashboards

-
- Add an admin user in production environment for deployment
 - Add revision history for Dashboards
 - Dashboard Revision objects are getting new UID rather than copying older ones
 - Add pattern colours to charts
 - Turn off stompjs debug logging for client speed improvement
 - Change the Aggregation section on Dashboards, to contain a UI rather than raw JSON
 - Edit or Remove extra aggregation attributes from aggregation builder views

Sprint 0.0.6

- Accept raw data from data source without running aggregations
- Add X-Auth-Token header and JWT to each agent
- Set up VS Code Latex work Shop and config settings in detailed description
- Add JWT token to prod and dev bootstrap envs
- Add code editor for creating rules on the schema creation page
- Add most recent document option to aggregations
- cloc command
- Put iris-crypto-rates on raspberry pi and schedule it as a cron job
- Add dynamic REST call to all agents to grab url from iris for agent
- Deploy iris v0.0.6