

# Sprint #1 Review

## Links

[Pivotal Tracker](#)

[GitHub Repo](#)

[Heroku Deployment](#)

## Sprint Dates

September 28, 2024 - October 11, 2024

## Team Information

Product Owner: Nathaniel Bright  
Scrum Master: Kumaran Gowrisankar  
Developer: Jingrui Mao  
Developer: Zhaoming Xu  
Developer: Zhicheng Chen  
Developer: Chi-Ming Lee  
Developer: Xinyi Zhao  
Developer: Jingyuan Mu

## Sprint Goal

The sprint's goal was to fix the email delivery system, add file upload type in the contract page, change user invite dropdown, along with making UI improvements. We also prioritised understanding the legacy codebase and identifying potential refactorings.

## Sprint Achievements

We completed file upload type and user invite dropdown. We also made UI changes, demonstrated it to the customer to get feedback for further refinement. We explored the codebase and identified tests that we can add to make the system more robust. However, we could not fix the email delivery issue because we were blocked by BVCOG's IT team.

Below is the list of stories implemented

- *Change User Levels from "1/2/3" to "Admin/Gatekeeper/User"*
- *Add "Amendment" to "Contract type" dropdown*
- *In the "Contract type" dropdown*

- Change “Create a Contract” to “Add a Contract”:

## Sprint Backlogs

### Added Backlogs

- RSpec Tests were added.

⚙	(1) Implement RSpec test: ReportsHelper - add examples	Start	<input type="checkbox"/>
⚙	(2) Implement RSpec test: ReportsHelper - add examples	Start	<input type="checkbox"/>
⚙	(3) Implement RSpec test: UsersHelper - add examples	Start	<input type="checkbox"/>
⚙	(4) Implement RSpec test: VendorsHelper - add examples	Start	<input type="checkbox"/>
⚙	(5) Implement RSpec test: ContractMailer - add examples	Start	<input type="checkbox"/>
⚙	(6) Implement RSpec test: ContractDocuments GET /index - add examples	Start	<input type="checkbox"/>
⚙	(7) Fix and implement RSpec test: /contracts POST /create with valid parameters - redirects to the created contract	Start	<input type="checkbox"/>
⚙	(8) Fix and implement RSpec test: /contracts POST /create with invalid parameters - renders a successful response	Start	<input type="checkbox"/>
⚙	(9) Fix and implement RSpec test: /contracts PATCH /update with invalid parameters - renders a successful response	Start	<input type="checkbox"/>
⚙	(10) Implement RSpec test: /reports GET /index - renders a successful response	Start	<input type="checkbox"/>
⚙	(11) Implement RSpec test: /reports GET /show - renders a successful response	Start	<input type="checkbox"/>
⚙	(12) Implement RSpec test: /reports POST /create with valid parameters - creates a new Report	Start	<input type="checkbox"/>
⚙	(13) Implement RSpec test: /reports POST /create with valid parameters - redirects to the created report	Start	<input type="checkbox"/>
⚙	(14) Implement RSpec test: /reports POST /create with invalid parameters - does not create a new Report	Start	<input type="checkbox"/>
⚙	(15) Implement RSpec test: /reports POST /create with invalid parameters - renders a successful response	Start	<input type="checkbox"/>
⚙	(16) Implement RSpec test: /reports PATCH /update with valid parameters - updates the requested report	Start	<input type="checkbox"/>
⚙	(17) Implement RSpec test: /reports PATCH /update with valid parameters - redirects to the report	Start	<input type="checkbox"/>
⚙	(18) Implement RSpec test: /reports PATCH /update with invalid parameters - renders a successful response	Start	<input type="checkbox"/>
⚙	(19) Implement RSpec test: /reports DELETE /destroy - destroys the requested report	Start	<input type="checkbox"/>
⚙	(20) Implement RSpec test: /reports DELETE /destroy - redirects to the reports list	Start	<input type="checkbox"/>
⚙	(21) Implement RSpec test: /users PATCH /update with valid parameters - updates the requested user	Start	<input type="checkbox"/>
⚙	(22) Implement RSpec test: /users PATCH /update with valid parameters - redirects to the user	Start	<input type="checkbox"/>
⚙	(23) Implement RSpec test: /vendors PATCH /update with valid parameters - updates the requested vendor	Start	<input type="checkbox"/>
⚙	(24) Implement RSpec test: /vendors PATCH /update with valid parameters - redirects to the vendor	Start	<input type="checkbox"/>
⚙	(25) Implement RSpec test: UsersController routing - routes to #create	Start	<input type="checkbox"/>
⚙	(26) Fix and implement RSpec test: UsersController routing - routes to #create	Start	<input type="checkbox"/>
⚙	(27) Fix and implement RSpec test: reports/show - renders attributes in <p>	Start	<input type="checkbox"/>
⚙	Fix and implement RSpec test: vendors/edit - renders the edit	Start	<input type="checkbox"/>

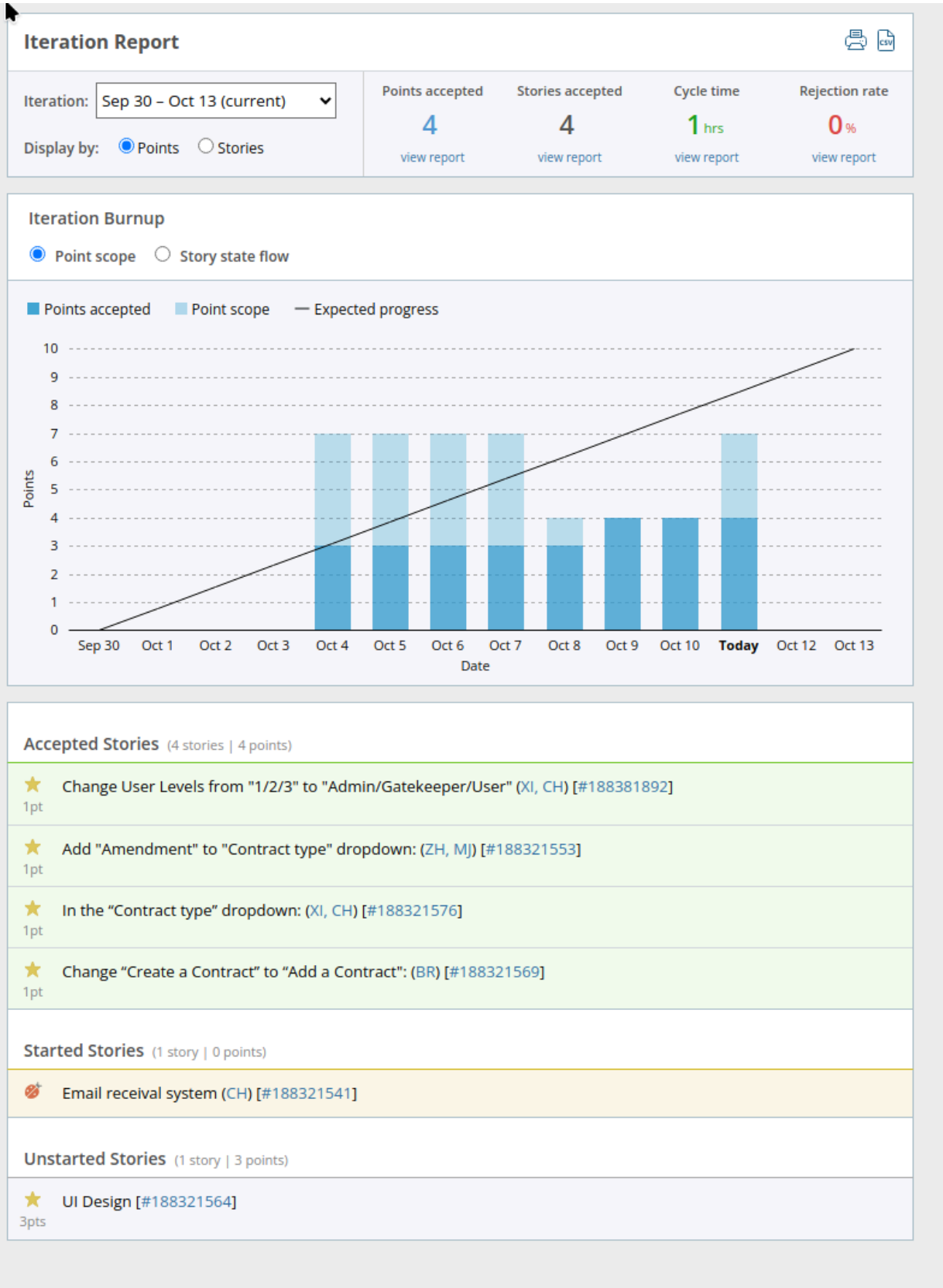
### Modified Backlogs

- None

### Removed Backlogs

- None.

Burn Down Chart

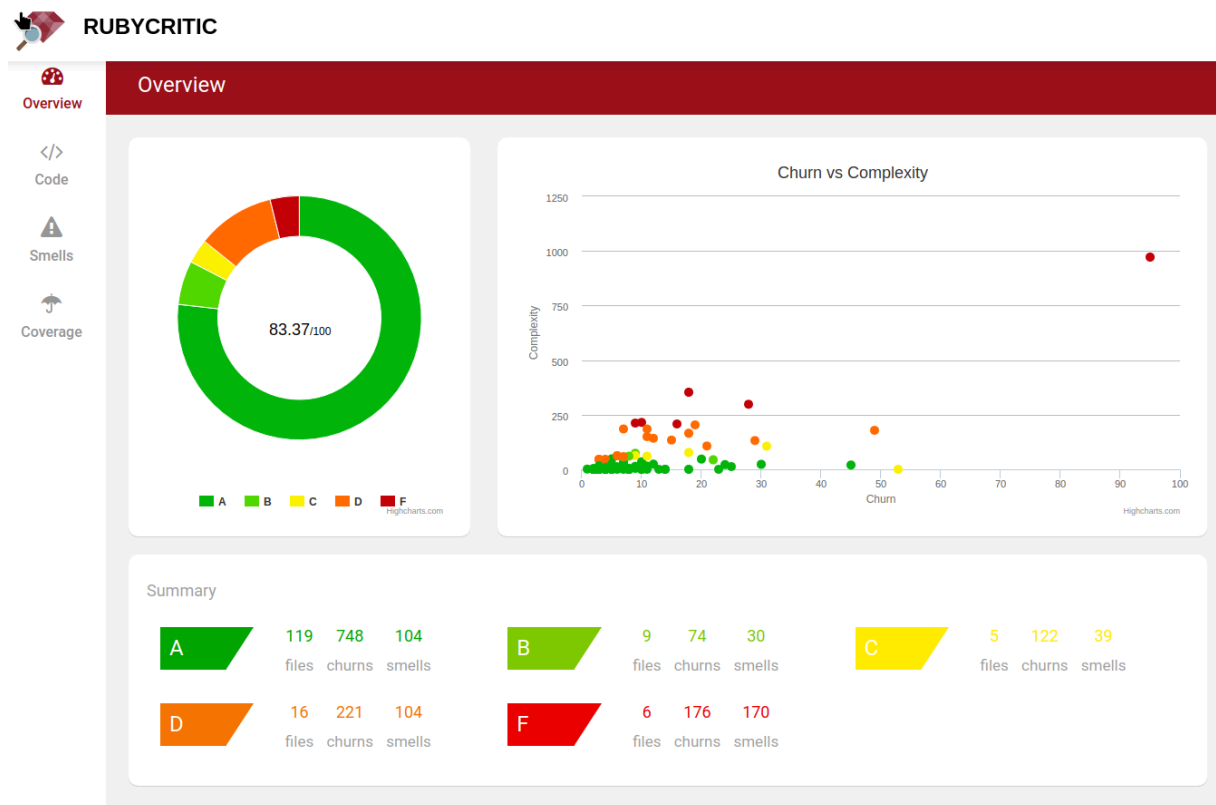


## Legacy Project & Refactoring

- We have not undertaken any refactorings yet.

## Evaluations of Code & Tests

### *Code smells and Style offences*



### *Test coverage*

```
98 scenarios (2 failed, 96 passed)
879 steps (2 failed, 877 passed)
0m23.886s
Coverage report generated for Cucumber Features to /home/kumaran/gdev/tamu/sem1/606/bri
ghtn-bvcog/coverage/cucumber. 710 / 781 LOC (90.91%) covered.
```

Note: This is mostly from legacy code, next sprint report we will show improvement from current status.

## Customer Meeting and Demonstration

Date: 04-10-2024 - 12:30 to 13:40 CDT

Place: 382H Zachary Building, College Station, TX

Participants: Susan Lightfoot and Nathaniel Bright

### Summary:

During this meeting, we reviewed with the client the current state of the program. At this point we had our first MVP with some of the UI changes and a few user stories implemented. During this meeting we brought up the blocker for deployment and email- which they pointed us to IT for the needed decryption key. Lastly, they proposed a brand new project and we discussed this with them. The project was a light budgeting "form" for the guests of their services.

## Sprint #1 Retrospective

- We discussed the need for a more streamlined way to test and merge changes into main. The suggestion was to use a test environment that we can use to merge our changes before merging into main.
- Assumptions that went astray: Email bug was due to code issue
  - During sprint planning we assumed the email delivery was due to a code bug and that it could be fixed from our side. However, it turned out to be an IT issue from BVCOG's side and therefore we had to reach out to them to unblock us. This took longer than expected and therefore we could not fix the bug in this sprint. We plan to finish it in the next sprint
- We failed to accommodate spring break week and mid-term exams: Team members' plans for spring break and preparation plans mid-term exams were not accounted for during the sprint plan and therefore our user stories were not completed in time.
- Better StandUp Intervals - We discussed our work during class discussions, however we found that 1 day between Monday and Wednesday and 4 days between Wednesday and the next Monday were too short and too long respectively.
- Helpful Changes
  - Review team's schedule during sprint planning
  - Challenge assumptions

- Schedule Standups whilst being cognizant of team members' availability