



Plan.ly

CS 673
Team 1 - UNO
Iteration 0



Team Overview

- Team **UNO**
- Working to build Plan.ly, an event planning tool

Name	Role
George Wright	Team Leader
Vibhu Bhatia	Backup Team Leader
Aysha Zenab Kenza	Security Leader
Karen Sommer	Design/Implementation Leader
Zach Schandorf-Lartey	Configuration Leader
Chris Kulig	Requirements Leader
Matt Dowding	Quality Assurance Leader

Plan.ly Overview

- Event planning and management hub
- Track event details, location, date, and time
- Designate co-owners and manage invite list
- Create to-do lists to track task completion
- View personal feed of all upcoming events



Modified SCRUM

- Daily Meetings replaced by Once / twice a week meetings, given the timeframe and the scope of project.
 - Meetings on Sunday cover Retrospective as well as concrete objectives for next sprint.
 - Additional Meeting on Monday, if needed to discuss any additional agenda.
- Product Increment - Result of every iteration produces a workable product that can be deployed.
- Ranked Backlog - Since some features are more important initially than other features we are working on a ranked backlog to build a workable MVP fast and then refine on it.



Modified SCRUM

- Adaptable Product Dev. - Due to different working schedules of each member progress is measured at the weekly meeting rather than daily therefore, users can complete their user stories at their convenience.
- Mentorship Process - A Technical workshop takes place once a week so that more experienced people can teach basic scripting to those unfamiliar with that language.





Requirements

-Functional



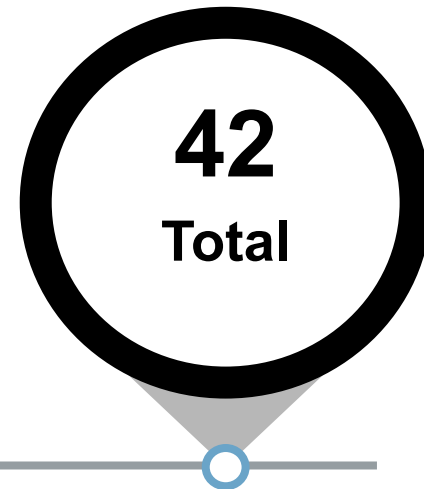
Core features



Nice to have
features



Cool to have
features



Requirements

Create private event

As a web site member, I want to create a private event, so that I can share details as the description, location and starting and ending date and time, of the event.

View my private event

As an event owner, I want to view my private event, so that I see how my event is going.

Receive Account Creation Email Notification

As a web site user/member, I want to receive an email notification so that I can sign in or create an account to attend the event

Log in

As a web site/user I want to be able to log into my account to securely gain access to my resources

-Functional

Sign up

As a website user I want to be able to create an account so that I can create, view, and manage my events

Create Invite List

As an event owner, I want to create an invitee list so that I can have a record of the people coming




Requirements

-Non Functional

01	Django	<ul style="list-style-type: none">• Django is an open-source framework for backend web applications• Simplicity, flexibility, reliability, and scalability
02	React	<ul style="list-style-type: none">• It is open-source, front end, JavaScript library for building user interfaces or UI components.• The main purpose of React is to be fast, scalable, and simple
03	MongoDB	<ul style="list-style-type: none">• Allow to store your records without worrying about the data structure such as the number of fields or types of fields to store values
04	OAuth	<ul style="list-style-type: none">• OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.

Quality Assurance Plan

1	Metrics	<ul style="list-style-type: none">• This section has two types of metrics: product metrics and process metrics• Define what metrics we will be using
2	Standard	<ul style="list-style-type: none">• development branch will have set up and explanation documentation for developers to read when they branch off of.• Code should be thoroughly commented so that functions describe their purpose, making it easier for reviewers to look over pull requests.
3	Agile Testing	<ul style="list-style-type: none">• Requirement Analysis• Design• Execute• Report• Re-Test• Release Testing 
4	Review Process	<ul style="list-style-type: none">• Using CI/CD pipeline that doesn't allow code to be merged to the main branch without going through a pull request process with the team leader, backup team leader, or configuration leader as reviewers
5	Defect Management	<ul style="list-style-type: none">• As bugs show up they should be entered into pivotal and the QA Leader is responsible for filling out and keeping track of the bugs



Code Management

```
main
|
|---development
|   |
|   |---feature/feature-name
|   |---fix/fix-name
|   |---branch-name
|
|---fix/fix-name
```

main: release branch, deploys to Heroku

development: stable working branch, merges into main

feature/feature-name: Pulled from development. Working branch for a specific feature, merges into development

fix/fix-name: hotfix working branch

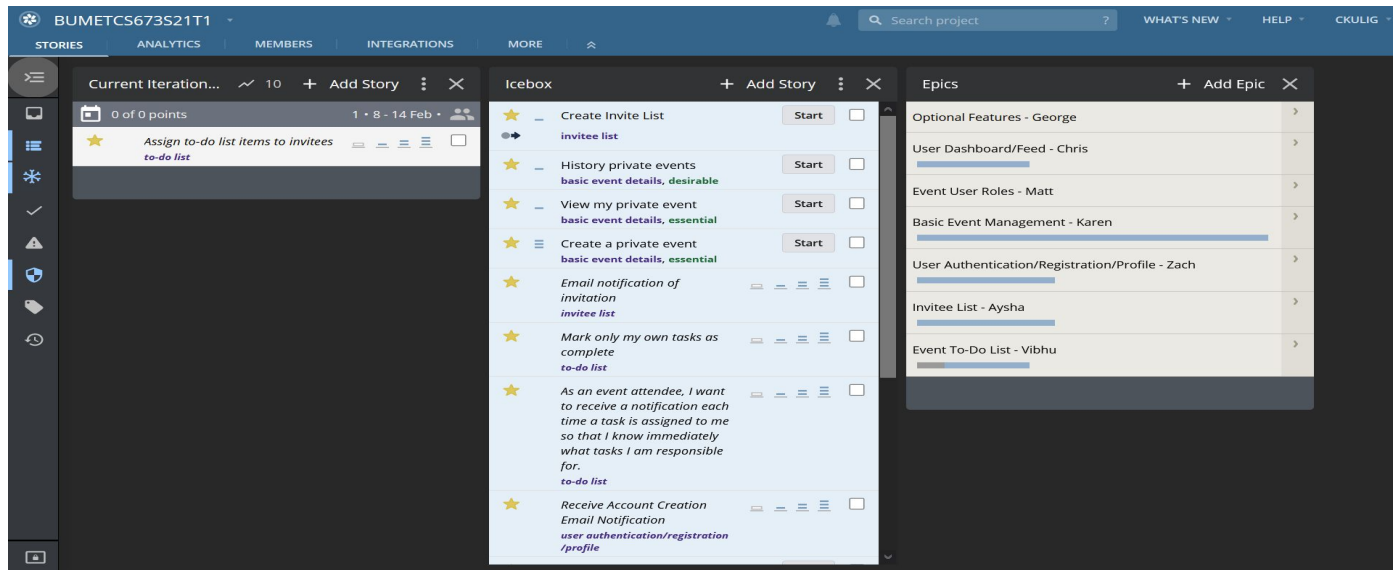
When a feature is completed, a pull request is created to merge it into development. After it is reviewed, it is merged

Code must follow coding standards outlined by pylint and eslint respectively, and will be linted whenever a pull request is made.

Functions and classes must have comments describing their purpose

Pivotal

- Assess difficulty & implementation time
- Isolate unrefined stories (Icebox)
- Manage developed stories through iterations
- Organize related stories through Epics
- Analyze metrics - keep pace
- Uphold the values of the Scrum methodology





Risk Management



More time-intensive user stories will fall to a subset of team members to implement

- Hosting workshops for less experienced team members to become familiar with frameworks and deployment process
- Assigning most crucial/essential tasks to experienced engineers in earlier iterations of the project → smaller/easier tasks for newer engineers as the project goes on.

Thank you

Questions?

