# Kan-CLI

Command line Kanban board

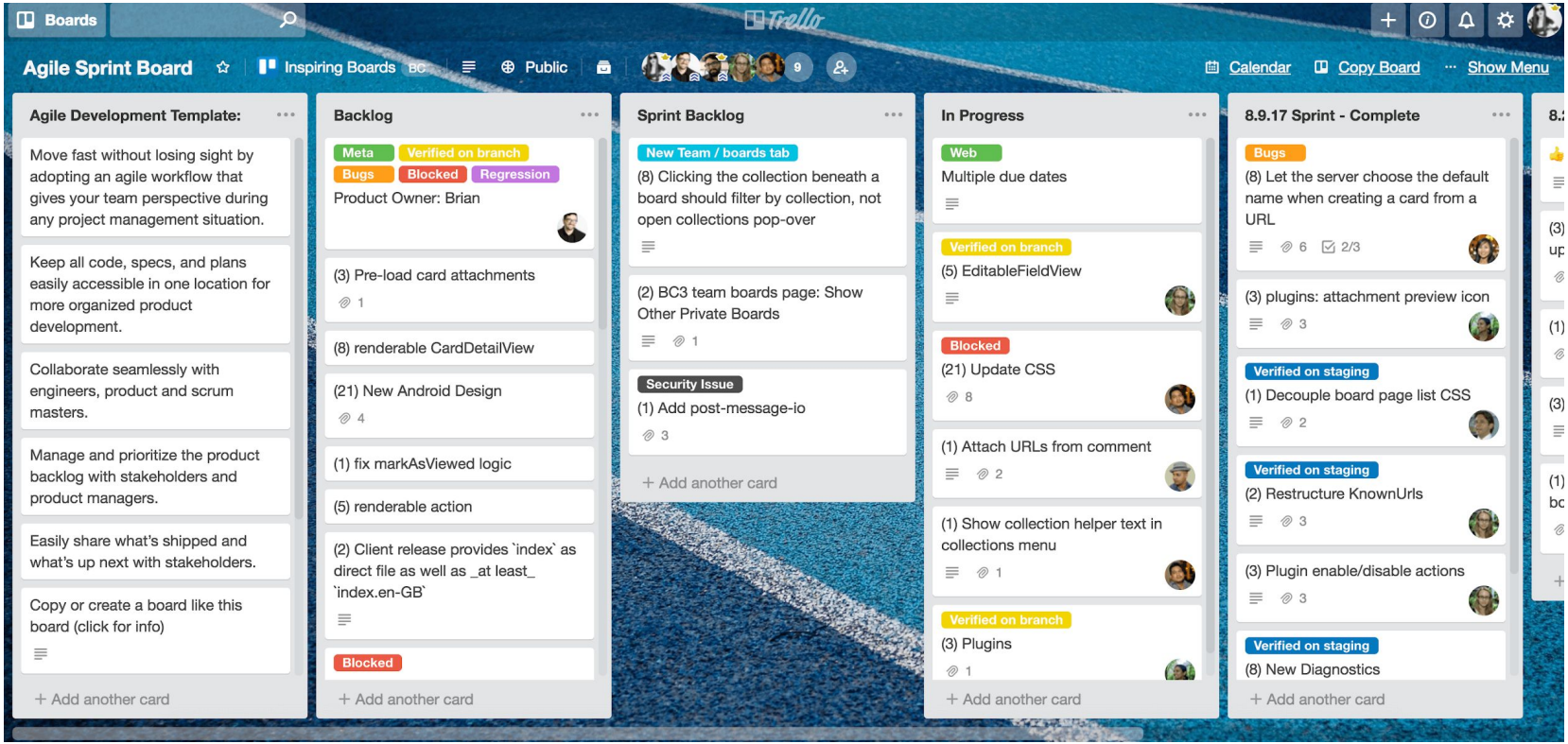
# App Overview

> “A **kanban board** is an agile project management tool designed to help visualize work, limit work-in-progress, and maximize efficiency (or flow).” - https://www.atlassian.com/agile/kanban/boards

Kanban boards allow individuals or teams to:

- Organise tasks into groups.
- Visualise the status of tasks within a project.
- Limit work in progress and maximise efficiency.

# Kanban Board Example




(https://blog.trello.com/hs-fs/hubfs/Screen_Shot_2018-07-05_at_2.43.52_PM.png?width=2876&name=Screen_Shot_2018-07-05_at_2.43.52_PM.png)

# Features

Users can:

- Create boards for different projects
- Create multiple Lists within each board
- Create cards and add them to lists (and limit the number of cards that can be added)
- Move cards between lists
- Switch between boards

# Classes

**Board**

- A title (String)
- A creation date (Time object)
- A collection of lists (Hash data type; List title => List object)

The user will be prompted to create a board object when the app is run. They can also create more boards from the menu. Each time a board is created it is set as the “current board”. The user is able to change the current board.

**List**

- Title (String)
- Card limit (Integer - the maximum number of cards the list can contain, user defined)
- Cards (Hash: card id => card object)

List objects can be created at any time from the menu, they are stored within the currently active board object. Lists are stored in a hash - the key is the title and the value is the list object itself.

# Classes

**Cards**

- ID
- Description
- Creation date

# Functionality

Boards have methods for adding and deleting lists, as well as displaying in the terminal.

Lists have functionality for adding and deleting cards. Moving cards is achieved by adding to the new list and then deleting from the old list.

Cards don’t have functionality as they are just placeholders for task description.

# Code

```
# Store all of the users variables
# which describe the state of the application
$state = {
    "user" => {},
    "boards" => {},
    "current board" => nil
}
```

The state variable stores all of user input.

- The username and password of the current user
- Each time a user creates a board it is added to the “boards” hash.
- The currently active board is used to determine which board should be displayed in the terminal. This variable is updated whenever the user creates a new board.

# Code

Control flow is handled through a combination of the input_loop function and display_menu function

```ruby
# Methods as array elements (https://stackoverflow.com/questions/13948910/ruby-methods-as-array-elements-how-do-they-work)
input_flow = [method(:create_board), method(:create_list), method(:create_card)]

# Start the loop at a given position
if start_position == "board"
    input_flow.each { |m| m.call }
elsif start_position == "list"
    input_flow[1..2].each { |m| m.call }
elsif start_position == "card"
    input_flow[2].call
end

# Show the user their updated board
$state["current board"].display_board
```

# Code

display_menu

```
# Define the menu
menu = $prompt.select("What next?") do |menu|
    menu.default 1

    menu.choice 'Create a New Board', 5
    menu.choice 'Switch Board', 6
    menu.choice 'Add a List', 1
    menu.choice 'Add a Card', 2
    menu.choice 'Move card', 3
    menu.choice 'Exit', 4
end
```

# Code

display_menu

```
# Handle user input
case menu
when 1
    input_loop("list")
when 2
    input_loop("card")
when 3
    move_card
when 4
    exit
when 5
    input_loop("board")
when 6
    switch_board
end
```

# Demonstration