

projmgr

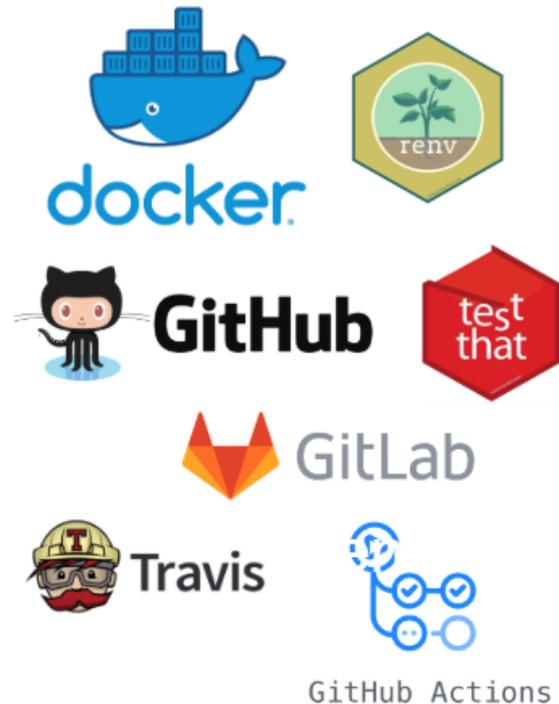
Managing the human dependencies of your project

Emily Riederer (@emilyriederer)

Project Management

Developer Tools

Workflow is *automated* and *integrated*



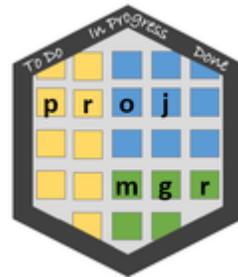
Project Management

Project management is *manual* and *fragmented*

- Email
- Slack
- Meetings
- Slides
- Reports
- Siloed software (e.g. Jira, Trello)

projmgr enables GitHub-based project management via R

- **Integrated** with GitHub to eliminate number of tools
- **Responsive** to existing workflow and infers metadata from organic interaction
- **Automated** reporting and analysis for fewer ancillary tasks



Demo

```
library(projmgr)
```

```
## Warning: package 'projmgr' was built under R version 4.0.5
```

```
repo <- create_repo_ref('emilyriederer', 'project-demo')
```

```
## Requests will authenticate with GITHUB_PAT
```

```
repo
```

```
## + Repository Owner: emilyriederer  
## + Repository: project-demo  
## + From URL: https://api.github.com/  
## + Authenticating with: GITHUB_PAT
```

Define plan.yml:

```
- title: Application Design
  description: >
    Define requirement and prototype initial
    Shiny application design
  due_on: 2020-07-30T12:59:59Z
  issue:
  - title: Conduct user interviews
    body: Understand user needs and pain points
    labels: [team-design]
  - title: Prototype interface
    body: >
      Create minimum viable interface with shinyipsum
      to get user feedback
    assignees: emilyriederer
    labels: [team-dev]
  - title: Solicit user feedback on design
    labels: [team-design]

- title: Application Development
  description: Build production grade Shiny app
  due_on: 2020-09-30T12:59:59Z
  issue:
  - title: Enhance frontend design
    labels: [team-dev]
  - title: Write backend business logic
    labels: [team-dev]

- title: Beta Launch Application
  description:
  due_on: 2020-11-30T12:59:59Z
  issue:
  - title: Deploy application to server
    labels: [team-dev]
  - title: Conduct user research study
    labels: [team-design]
```

Define plan.yml:

```
- title: Application Design
  description: >
    Define requirement and prototype initial
    Shiny application design
  due_on: 2020-07-30T12:59:59Z
  issue:
  - title: Conduct user interviews
    body: Understand user needs and pain points
    labels: [team-design]
  - title: Prototype interface
    body: >
      Create minimum viable interface with shinyipsum
      to get user feedback
    assignees: emilyriederer
    labels: [team-dev]
  - title: Solicit user feedback on design
    labels: [team-design]

- title: Application Development
  description: Build production grade Shiny app
  due_on: 2020-09-30T12:59:59Z
  issue:
  - title: Enhance frontend design
    labels: [team-dev]
  - title: Write backend business logic
    labels: [team-dev]

- title: Beta Launch Application
  description:
  due_on: 2020-11-30T12:59:59Z
  issue:
  - title: Deploy application to server
    labels: [team-dev]
  - title: Conduct user research study
    labels: [team-design]
```

Define plan.yml:

```
- title: Application Design
  description: >
    Define requirement and prototype initial
    Shiny application design
  due_on: 2020-07-30T12:59:59Z
  issue:
- title: Conduct user interviews
  body: Understand user needs and pain points
  labels: [team-design]
- title: Prototype interface
  body: >
    Create minimum viable interface with shinyipsum
    to get user feedback
  assignees: emilyriederer
  labels: [team-dev]
- title: Solicit user feedback on design
  labels: [team-design]

- title: Application Development
  description: Build production grade Shiny app
  due_on: 2020-09-30T12:59:59Z
  issue:
- title: Enhance frontend design
  labels: [team-dev]
- title: Write backend business logic
  labels: [team-dev]

- title: Beta Launch Application
  description:
  due_on: 2020-11-30T12:59:59Z
  issue:
- title: Deploy application to server
  labels: [team-dev]
- title: Conduct user research study
  labels: [team-design]
```

Define plan.yml:

```
- title: Application Design
  description: >
    Define requirement and prototype initial
    Shiny application design
  due_on: 2020-07-30T12:59:59Z
  issue:
  - title: Conduct user interviews
    body: Understand user needs and pain points
    labels: [team-design]
  - title: Prototype interface
    body: >
      Create minimum viable interface with shinyipsum
      to get user feedback
    assignees: emilyriederer
    labels: [team-dev]
  - title: Solicit user feedback on design
    labels: [team-design]

- title: Application Development
  description: Build production grade Shiny app
  due_on: 2020-09-30T12:59:59Z
  issue:
  - title: Enhance frontend design
    labels: [team-dev]
  - title: Write backend business logic
    labels: [team-dev]

- title: Beta Launch Application
  description:
  due_on: 2020-11-30T12:59:59Z
  issue:
  - title: Deploy application to server
    labels: [team-dev]
  - title: Conduct user research study
    labels: [team-design]
```

Upload to GitHub:

```
plan <- read_plan('plan.yml')
post_plan(repo, plan = plan)
```

- 3 Milestones
- 7 Issues
- 2 Labels
- 1 Assignee

in

2 lines versus ~75 button clicks

Milestones from plan

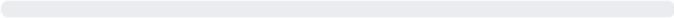
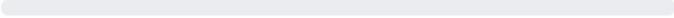
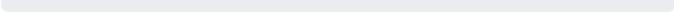
emilyriederer / project-demo

Unwatch 1 Star 0 Fork 0

Code Issues 7 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Labels Milestones New milestone

3 Open 0 Closed Sort

<h3>Beta Launch Application</h3> <p>Due by November 30, 2020 Last updated less than a minute ago</p> <p>0% complete 2 open 0 closed</p> <p>Edit Close Delete</p>	
<h3>Application Development</h3> <p>Due by September 30, 2020 Last updated less than a minute ago</p> <p>Build production grade Shiny app</p> <p>0% complete 2 open 0 closed</p> <p>Edit Close Delete</p>	
<h3>Application Design</h3> <p>Due by July 30, 2020 Last updated less than a minute ago</p> <p>Define requirement and prototype initial Shiny application design</p> <p>0% complete 3 open 0 closed</p> <p>Edit Close Delete</p>	

Issues from plan

emilyriederer / project-demo

Unwatch 1 Star 0 Fork 0

Code Issues 7 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Filters is:issue is:open Labels 11 Milestones 3 New issue

7 Open 0 Closed Author Label Projects Milestones Assignee Sort

- Conduct user research study** team-design
#7 opened 42 minutes ago by emilyriederer [Beta Launch App...](#)
- Deploy application to server** team-dev
#6 opened 42 minutes ago by emilyriederer [Beta Launch App...](#)
- Write backend business logic** team-dev
#5 opened 42 minutes ago by emilyriederer [Application Deve...](#)
- Enhance frontend design** team-dev
#4 opened 42 minutes ago by emilyriederer [Application Deve...](#)
- Solicit user feedback on design** team-design
#3 opened 42 minutes ago by emilyriederer [Application Design](#)
- Prototype interface** team-dev 
#2 opened 43 minutes ago by emilyriederer [Application Design](#)
- Conduct user interviews** team-design
#1 opened 43 minutes ago by emilyriederer [Application Design](#)

Retrieve issues from GitHub

```
issues <- get_issues(repo, state = 'all') %>% parse_issues()
```

Generate text-based reports

```
report_plan(plan, show_ratio = FALSE)
```

Application Design

- Conduct user interviews
- Prototype interface
- Solicit user feedback on design

Application Development

- Enhance frontend design
- Write backend business logic

Beta Launch Application

- Deploy application to server
- Conduct user research study

```
report_progress(issues)
```

Beta Launch Application (0 % Complete - 0 / 2)

- Conduct user research study
- Deploy application to server

Application Development (0 % Complete - 0 / 2)

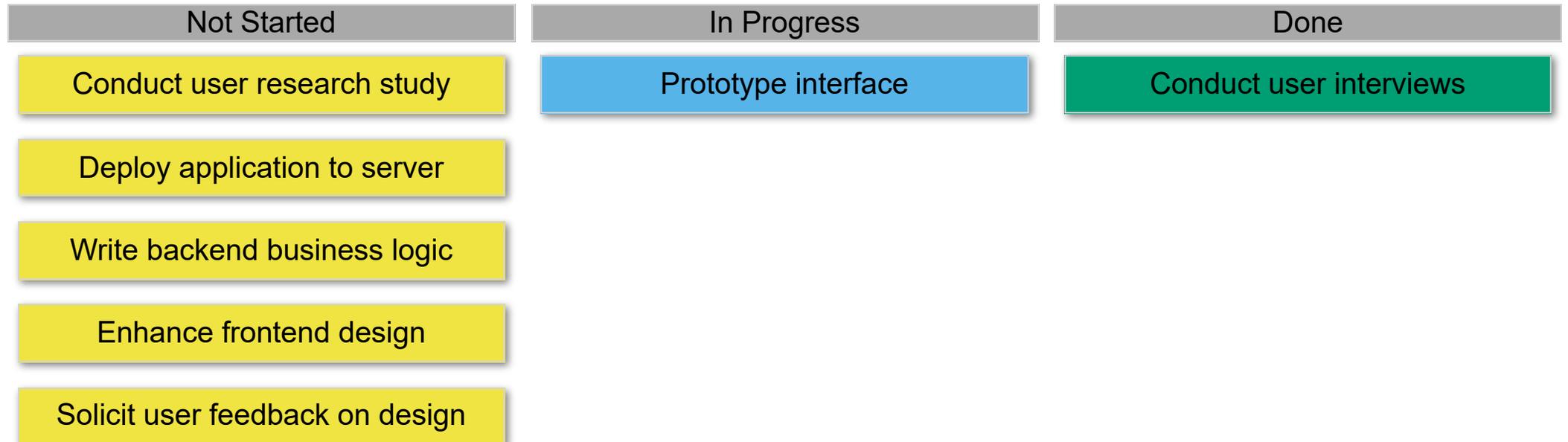
- Write backend business logic
- Enhance frontend design

Application Design (33 % Complete - 1 / 3)

- Solicit user feedback on design
- Prototype interface
- Conduct user interviews

Share more details with charts and dashboards

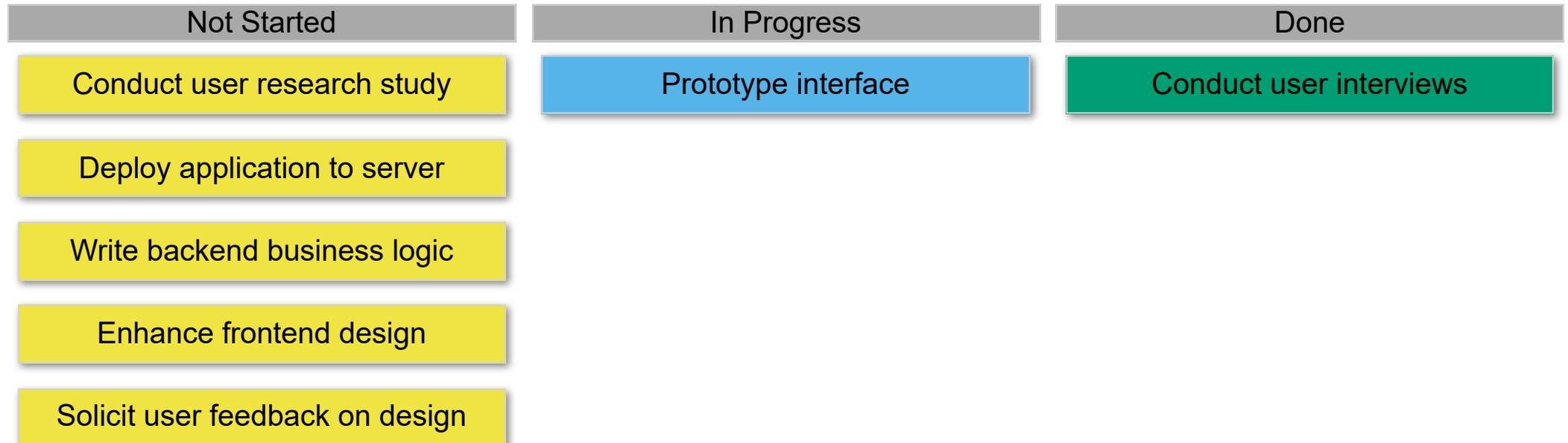
```
report_taskboard(issues,  
                 in_progress_when = is_assigned()  
                 )
```



Other views available including Gantt and waterfall charts

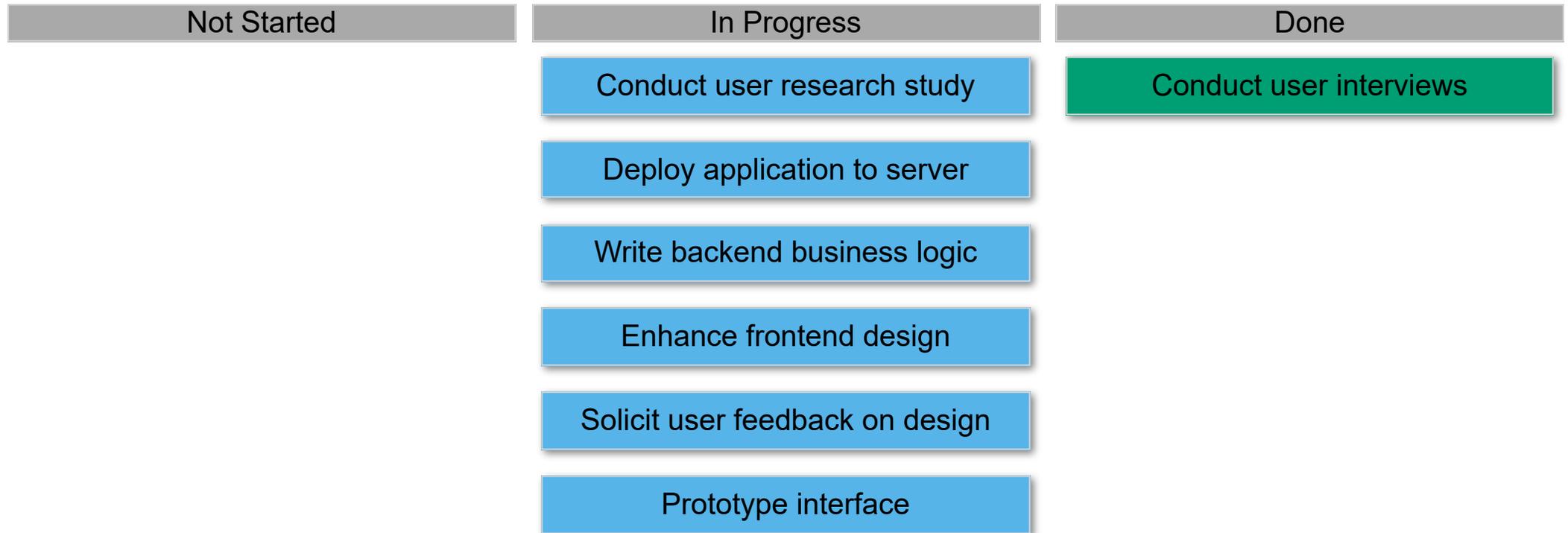
Customize reporting to your GitHub conventions

```
report_taskboard(issues,  
    in_progress_when = is_assigned()  
)
```



Customize reporting to your GitHub conventions

```
report_taskboard(issues,  
    in_progress_when = is_labeled()  
)
```



Customize reporting to your GitHub conventions

```
report_taskboard(issues,  
    in_progress_when = is_labeled_with('in-progress')  
)
```

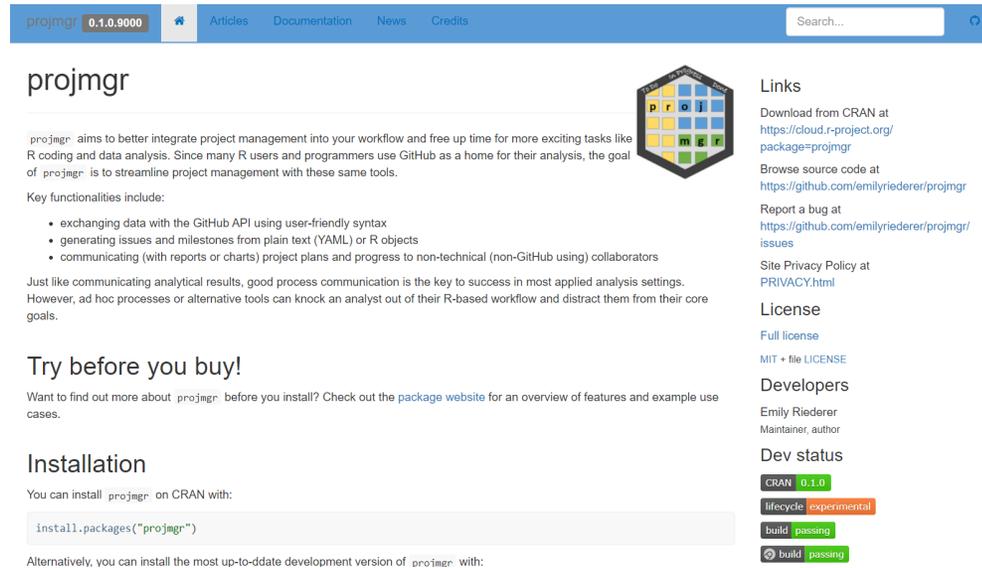
Not Started	In Progress	Done
Conduct user research study	Prototype interface	Conduct user interviews
Deploy application to server		
Write backend business logic		
Enhance frontend design		
Solicit user feedback on design		

Taskboard helpers provide a flexible set of options

Currently implemented:

```
## has_n_commits  
## is_assigned  
## is_assigned_to  
## is_created_before  
## is_due  
## is_due_before  
## is_in_a_milestone  
## is_in_milestone  
## is_labeled  
## is_labeled_with  
## is_part_closed
```

Check out the website!



The screenshot shows the homepage of the projmgr website. At the top, there is a navigation bar with the projmgr logo, version 0.1.0.9000, and links for Articles, Documentation, News, and Credits. A search bar is also present. The main content area features the projmgr logo, a description of the tool's purpose, and a list of key functionalities. A 'Try before you buy!' section encourages users to check the package website for more details. An 'Installation' section provides instructions for installing projmgr from CRAN, including a code snippet: `install.packages("projmgr")`. A 'Links' section on the right provides links to download from CRAN, browse source code on GitHub, report a bug, and view the site privacy policy and license. The license is MIT. The developers section lists Emily Riederer as the maintainer and author. The dev status section shows that the CRAN version is 0.1.0, the lifecycle is experimental, and the build is passing.

<https://emilyriederer.github.io/projmgr/>

Workflows

- Send emails
- File reprep
- Generate reporting website

Use Cases

- Project planning
- KPI tracking
- Hackathon facilitation

Open to feature requests!