CMSC388T

Project Management Tools
Today’s Lecture

1. **Pull Request**
   - How do you review code?

2. **Purpose of Tools**
   - Why do we use project management tools and what advantages do they have?

3. **Github Project Boards**
   - How does github incorporate these project management tools?
Branching, Pull Requests

Pull requests, Review, and Fork
When do we Branch and Why is it useful?
Branching

Allows for multiple team members to modify code and work on different assignment for a project.

Makes a local copy of the current code that allows a team member to modify and then push to be merged with the main branch.
1. Create Local Branch

Copies the main branch to allow user to make local changes and team members to work on the same project simultaneously.

```
git checkout [branch-name] -b
```
2. Push to Main

Push the local branch to the repository which will make the branch to be visible to all teammates.

Note: pushing the branch to the remote does not mean merging with the main branch

```git
git push -u origin [branch-name]
```
3. Pull Request (PR)

Create a pull request and assign reviewers to review and make comments on the changes.

This gives the teammate responsible for the task a chance to fix any potential bugs or problems that could arise after merging with the main branch.
4. Merge to Main

Once the reviewers approve the branch pushed to the main it is ready to then be merged with the main branch which will add the changes made from the local branch that was once copied to the main.

This process allows for multiple teammates to work on separate issues and merge them all in the end simultaneously.
### Branching to Feature vs Branching to Feature-Item

#### Branching to Feature
- Copy feature branch from Main branch
- Can have multiple people working on same feature
- Creates a branch that will be the subject of a new feature to implement

#### Branching to Feature-Item
- Copy feature-item branch from feature branch
- Typically one person works on each feature-item
- Creates a branch from the feature branch to address a specific item for the new feature
**Feature-Item Branching Process**

1. **Main Branch**
   - Start at Main Branch

2. **Feature Branch**
   - Copy from Main Branch to create a Feature Branch

3. **Feature-Item Branch**
   - Copy from Feature Branch to make an item specific branch

4. **Feature-Item Branch**
   - Make edits on feature-item branch to develop the feature-item

5. **Feature Branch**
   - Push changes to feature Branch and open Pull Requests, accept and merge items to feature branch

6. **Main Branch**
   - Once all feature-items are merged on Feature branch, push feature to Main and open PR
Project Management Tools

What are they and why are they useful?
Project Management Tools

- Organizational method for developers and managers to coordinate edits, and progress on work in one place
- Useful to manage projects so they stay on schedule
- Creates a central place to keep track of all team member progress
# Industry Application Examples

<table>
<thead>
<tr>
<th>Jira</th>
<th>Trello</th>
<th>Airtable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug, Issue Tracker, Test Case Management</td>
<td>Boards &amp; Cards to prioritize and organize Team tasks</td>
<td>Boards &amp; Cards to prioritize and organize multiple teams or company wide</td>
</tr>
<tr>
<td><strong>Pivotal Tracker</strong></td>
<td><strong>Slack</strong></td>
<td><strong>Teams</strong></td>
</tr>
<tr>
<td>Tracker for backlog, project breakdown, team transparency</td>
<td>Communication Tool that allows for integration with other applications like git</td>
<td>Microsoft communication tool that has conference call integration</td>
</tr>
</tbody>
</table>
Kanban Board

- Organizes team progress and transparency for assignments
  - Examples: Trello, Airtable, Jira
- Allow for teams to differentiate individual progress
- Team members can indicate their progress on a project with updating which assignments are in progress, yet to begin, or ready for review
- Very common application in the Industry World

Trello  Jira  Airtable
Scrum Board

- Organize and manage a timeline of issues and tasks in order to achieve a desired goal
- Primarily focus on ensuring that project assignments and goals adhere to the timeline
- Plan sprints (typically 2-3 week long goals) a team wants to accomplish while Kanban boards focus on individual team member tasks
3. Using Github Project Boards

How github allows you to use these tools?
Github Project Management

- Github has an integrated Project Management Feature that utilizes Kanban boards to keep track of team members progress and assignments.
- Creates a transparent, organized, and collaborative atmosphere for team projects.
- Keeps track of and organizes branching, pull requests, reviews, and merges.
Github Project Management

The Github site has a tab to organize multiple projects for teams.

Create projects to organize & collaborate with team members.
Collaborate with Github Project Manager

Customize the organization method of your project manager with the different templates that Github Project Manager has to offer. We recommend using the Automated Kanban with reviewers board.

<table>
<thead>
<tr>
<th>Templates</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
</tr>
<tr>
<td>Start from scratch with a completely blank project board. You can add columns and configure automation settings yourself.</td>
</tr>
<tr>
<td>Basic kanban</td>
</tr>
<tr>
<td>Basic kanban-style board with columns for To do, In progress and Done.</td>
</tr>
<tr>
<td>Automated kanban</td>
</tr>
<tr>
<td>Kanban-style board with built-in triggers to automatically move issues and pull requests across To do, In progress and Done columns.</td>
</tr>
<tr>
<td>Automated kanban with reviews</td>
</tr>
<tr>
<td>Everything included in the Automated kanban template with additional triggers for pull request reviews.</td>
</tr>
<tr>
<td>Bug triage</td>
</tr>
</tbody>
</table>
| Triage and prioritize bugs with columns for To }
Collaborate with Github Project Manager

Create Cards in order to create tasks for team members
Collaborate with Github Project Manager

Convert the new card to an ‘issue’ - this indicates that the card needs to be addressed and can then be assigned to the team member in charge of addressing it.
Collaborate with Github Project Manager

describe the issue and then convert to an issue
The ‘issue’ card has now been created and is ready to be assigned.
Collaborate with Github Project Manager

Assignees are people responsible for the task.

Labels are the type of issue that needs to be addressed.
Collaborate with Github Project Manager

Use the Kanan Board to keep track of progress among teammates
Collaborate with Github Project Manager

Link Pull Requests to Issue Cards in order to keep track of all assignments.

You can ‘link’ a Pull Request to an issue card. Select the PR that you opened and would like to link to the card.

Click on the blue issue title to expand and see full details of the card.
Collaborate with Github Project Manager

Once the Assigned Reviewer approves the PR, it can be updated as such on the Project Board.
Collaborate with Github Project Manager

Use the Project to have transparency within group members about stages of assignments
Clicker Quiz: Which of the following is an example of a Kanban Board?
Clicker Quiz: Which of the following is an example of a Kanban Board?
Description of the project and contains useful information for someone who visits the repository or project
- What is the project purpose
- How should it be used
- Why is it useful
- Who contributes to project
# ReadME.md

This repository has a few templates for README files and some notes about which type of information you could write on them.

## Prepare the content

- Readme files are made for developers (including you!), but also could be used for the final users.
- So while you are writing your readme files please consider a few things:

1. What is about?
   - Describe the content of your project or repository
   - Explain things the users would have a hard time understanding right away
2. What steps need to be taken?
   - Do they need to install any software?
   - Is there any hardware requirements or dependencies?
   - After the installation, how do they compile or run the code?
3. Execution examples
   - You could provide examples of execution with code and screenshots
4. Other things you could add:
   - Table of content
     - Text cases
     - Know bogs
     - Version
     - Contributors
     - License
     - References

## Templates

- _`cheat/tech.md`
  - A simple markdown cheat sheet (in construction)
- _`assignment.md`
  - An example of an university assignment
## Sub-lists

1. You could have
   1. ordered sub-lists
   2. Same thing
      - for unordered
      - sub-lists
2. Doesn’t matter if the main list is
   1. ordered or unordered
   2. you could mix them
      - in any level
   3. You could also align some text with
      - other items
      - without list-style
   4. ... without list-style

1. You could have
   1. ordered sub-lists
   2. Same thing
      - for unordered
      - sub-lists
2. Doesn’t matter if the main list is
   1. ordered or unordered
   2. you could mix them
      - in any level
   3. You could also align some text with
      - other items
      - without list-style

## 5. Links

URLs and URLs in angle brackets will automatically get turned into links as in `<www.mozilla.org/en-US/firefox/>`

... URLs and URLs in angle brackets will automatically get turned into links as in `<www.mozilla.org/en-US/firefox/>`

You could also paste the link in a new line without angle brackets:

... https://www.mozilla.org/en-US/firefox/

Or you could use inline-style link:
Firefox(https://www.mozilla.org/en-US/firefox/)

... Firefox(https://www.mozilla.org/en-US/firefox/)
# 6. Images


# 7. Youtube videos

They can't be added directly but you can add an image with a link to the video like this:

[![image alt text](https://github.com/inessadl/readme/blob/master/img/ue4.jpg)](https://www.youtube.com/watch?v=8Jd79v3ok)
# TestRepo

```java
```
System.out.println("Hello World");
```
```

```python
```
print("Hello World")
```
```