GitHub - Pull Request training

DIME Analytics

October 29, 2020

DIME - The World Bank - https://www.worldbank.org/en/research/dime

Who is this training for?

Who is this training for?

Anyone comfortable branching, committing and merging in Git/GitHub who is ready to be introduced to a more advanced workflow using Git/GitHub to ensure high quality code.

Content

THIS TRAINING HAS TWO PARTS

Part 1

Introducing the *branch-PR-merge* cycle

A best practice framework for implementing each task in a PR

Part 2

Introducing the Gitflow philosophy

A best practice framework for working with PRs in your project

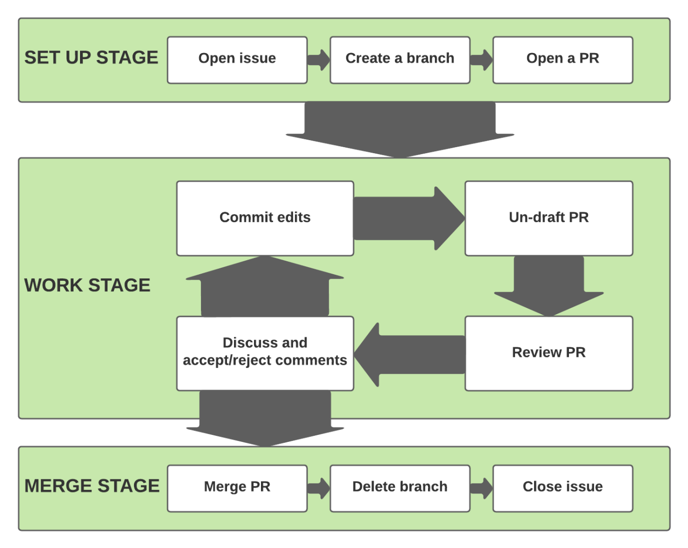
Part 1:
Best practices for the branch-PR-merge cycle

The branch-PR-merge cycle

Three stages with multiple steps in each

Complete a full *branch-PR-merge* cycle for **all tasks**

We will go over all steps but focus today on the **work stage**



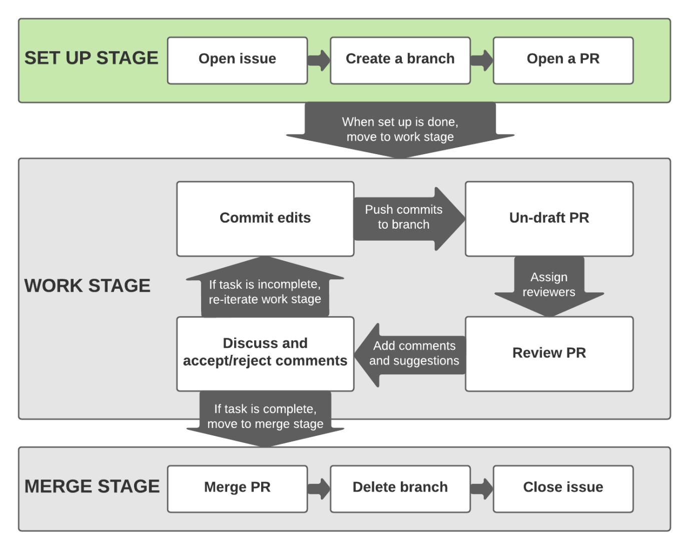
Stage: The set up stage

SET UP STAGE

Stage: The set up stage

Purpose of the set up stage:

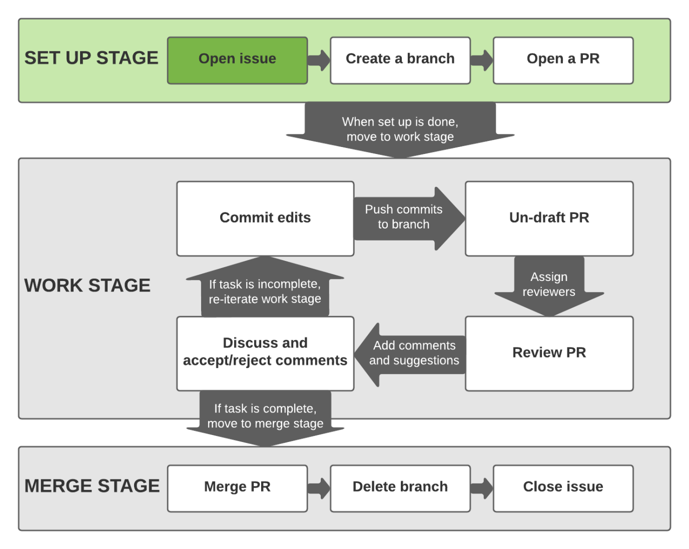
- Document the task you will work on
- Create a space in your repo to work on the task



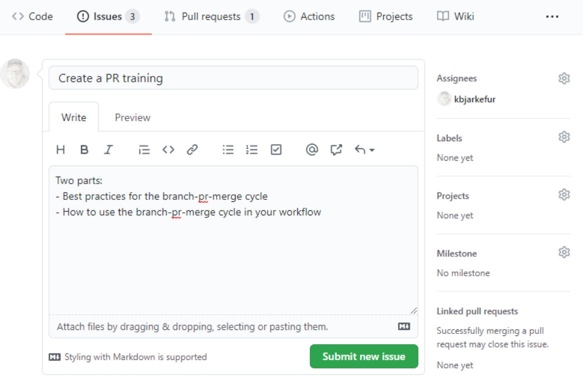
SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
Review PR
Add comments and suggestions
Discuss and accept/reject comments
If task is incomplete, re-iterate work stage
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Step: Create an issue

- Issues is a far superior way to document tasks compared to emails or in-person meetings!
- They allow other team members to provide input on how to solve the task
- Very quick tasks that do not require discussion may not need an issue


SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
If task is incomplete, re-iterate work stage
Discuss and accept/reject comments
Add comments and suggestions
Review PR
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Create an issue



Create a PR training

Two parts:
- Best practices for the branch-pr-merge cycle
- How to use the branch-pr-merge cycle in your workflow

Attach files by dragging & dropping, selecting or pasting them.

Styling with Markdown is supported

Assignees: kbjarkefur

Labels: None yet

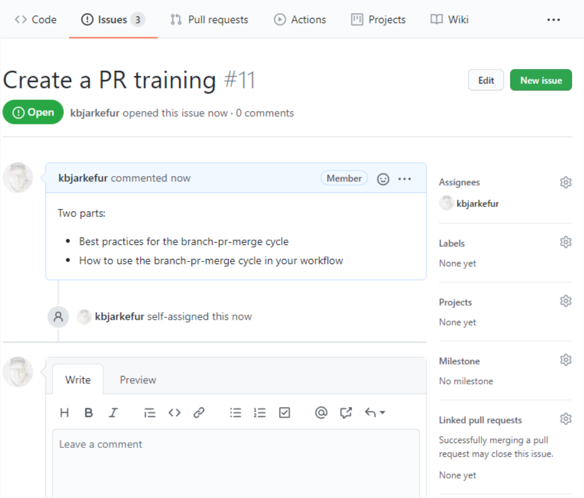
Projects: None yet

Milestone: No milestone

Linked pull requests: Successfully merging a pull request may close this issue. None yet

- Click the issues tab and then `New Issue`
- Describe the task and assign yourself or someone else to complete the task
- Ask the rest of the team of input if applicable
- Add labels to facilitate future searches

Browse an issue



Create a PR training #11

Open kbjarkefur opened this issue now · 0 comments

kbjarkefur commented now

Two parts:

- Best practices for the branch-pr-merge cycle
- How to use the branch-pr-merge cycle in your workflow

kbjarkefur self-assigned this now

Leave a comment

Assignees: kbjarkefur

Labels: None yet

Projects: None yet

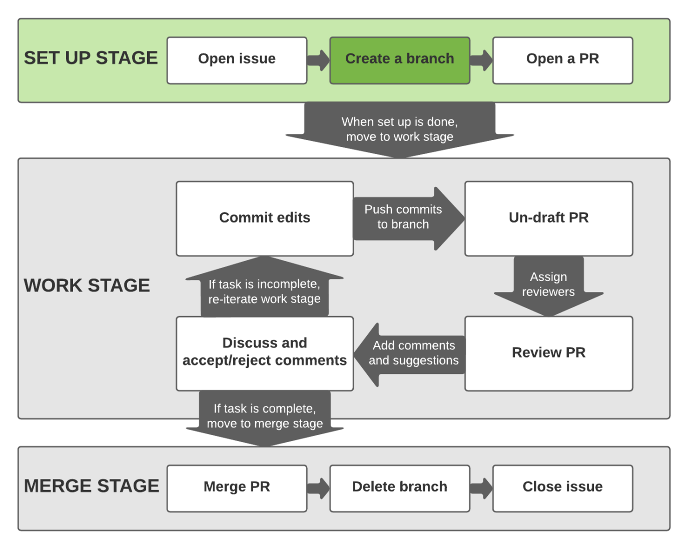
Milestone: No milestone

Linked pull requests: Successfully merging a pull request may close this issue. None yet

- Take note of the issue number (also found in end of issue URL) so you can refer to it
- Anyone can provide input using the comment field
- Assignees and labels can be updated as the task evolve
- Future team members can browse closed issues

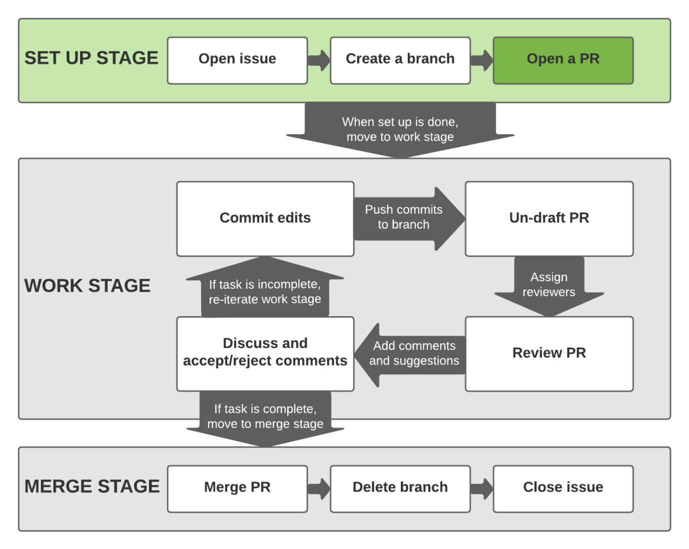
Create a branch

- Create branch on GitHub.com ▸ Step-by-step
- We create a branch to have a space where work on this task does not interfere with work on other tasks
- Name the branch after the task and add the issue number as suffix - for example: `pr-training-11`

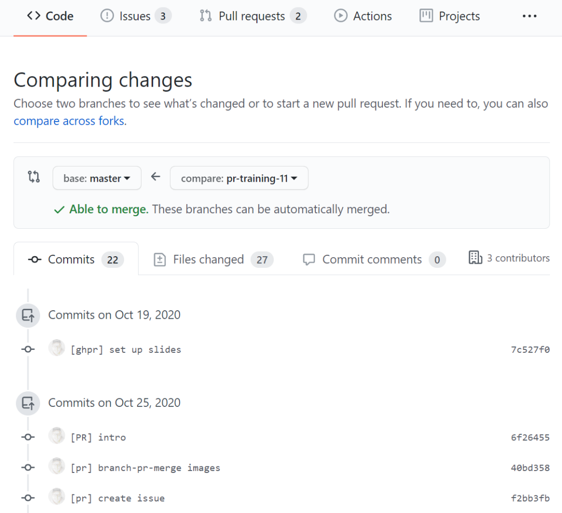

SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
Review PR
Add comments and suggestions
Discuss and accept/reject comments
If task is incomplete, re-iterate work stage
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Open the PR

- Create a PR **before** you implement the task - this creates a space on GitHub.com where the progress of this task can be followed
- You can create a *“draft PR”* that cannot be merged while in draft status - signals work in progress


SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
Review PR
Add comments and suggestions
Discuss and accept/reject comments
If task is incomplete, re-iterate work stage
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Create a PR



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base: master ← compare: pr-training-11

✓ Able to merge. These branches can be automatically merged.

Commits on Oct 19, 2020

```
[ghpr] set up slides                                7c527f0
```

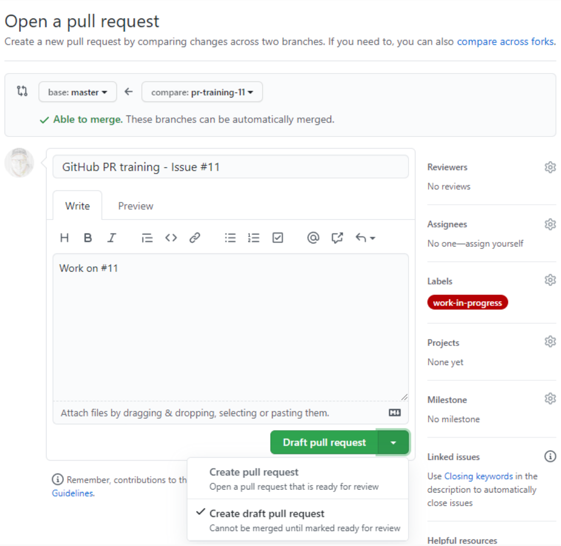
Commits on Oct 25, 2020

```
[PR] intro                                          6f26455
[pr] branch-pr-merge images                         40bd358
[pr] create issue                                   f2bb3fb
```

Recap of PR creation best practices:

- Go to the *"Pull request"* tab
- Make sure that you are comparing the correct branches
- Always browse the commits and the files changes before opening the PR
- You need at least one commit - Create/edit a README describing the task

Create a draft PR



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: master ← compare: pr-training-11

✓ Able to merge. These branches can be automatically merged.

GitHub PR training - Issue #11

Work on #11

Attach files by dragging & dropping, selecting or pasting them.

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: work-in-progress

Projects: None yet

Milestone: No milestone

Linked issues: Use Closing keywords in the description to automatically close issues

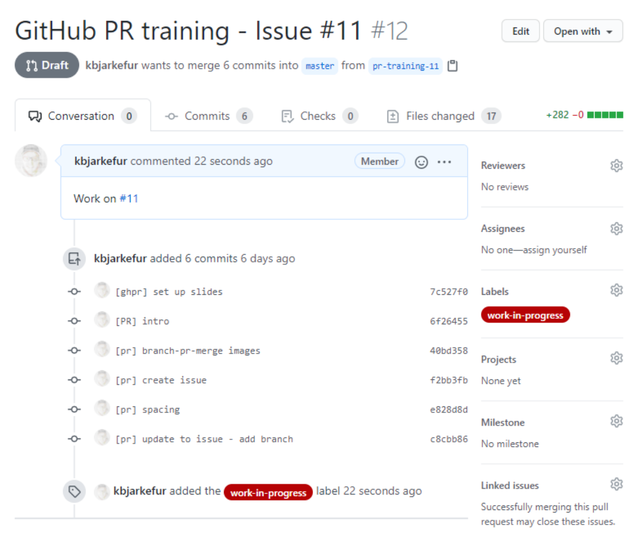
Remember, contributions to th… Guidelines.

Create pull request
Open a pull request that is ready for review

✓ Create draft pull request
Cannot be merged until marked ready for review

- Give the PR a name associated with the branch name but more descriptive
- Add #11 in the description to create a hyper link to issue - add `work-in-progress` label
- Click the arrow on the green button and select *Create draft pull request*

Browse draft PR



GitHub PR training - Issue #11 #12

Edit | Open with ▾

Draft kbjarkefur wants to merge 6 commits into `master` from `pr-training-11`

Conversation 0 | Commits 6 | Checks 0 | Files changed 17 | +282 −0

kbjarkefur commented 22 seconds ago · Member

Work on #11

kbjarkefur added 6 commits 6 days ago

```
[ghpr] set up slides                 7c527f0
[PR] intro                           6f26455
[pr] branch-pr-merge images          40bd358
[pr] create issue                    f2bb3fb
[pr] spacing                         e828d8d
[pr] update to issue - add branch    c8cbb86
```

kbjarkefur added the work-in-progress label 22 seconds ago

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: work-in-progress

Projects: None yet

Milestone: No milestone

Linked issues: Successfully merging this pull request may close these issues.

- A draft PR has the color code gray
- You can follow someone's work here - but no one is required to review anything yet
- The author will eventually remove the draft status and ask for reviews

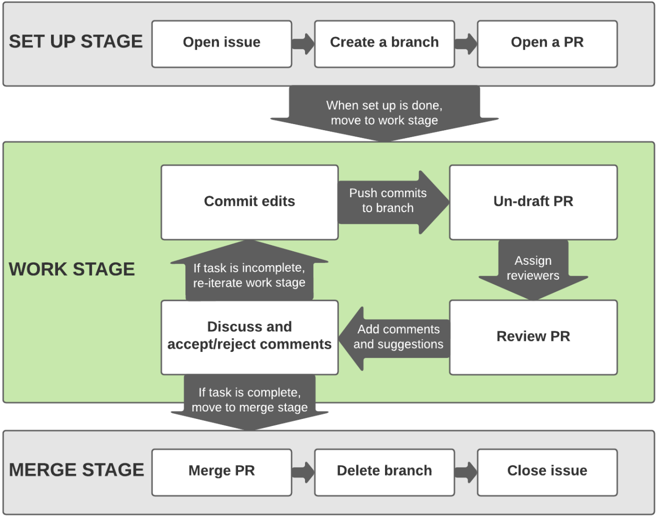
Stage: The work stage

WORK STAGE

Stage: The work stage

Purpose of the work stage:

- Write the code that implements this task
- Use the review features in a PR to improve code quality
- Iterate this stage until you have high-quality code



SET UP STAGE

Open issue → Create a branch → Open a PR

When set up is done, move to work stage

WORK STAGE

Commit edits → Push commits to branch → Un-draft PR

Assign reviewers

Review PR → Add comments and suggestions → Discuss and accept/reject comments

If task is incomplete, re-iterate work stage

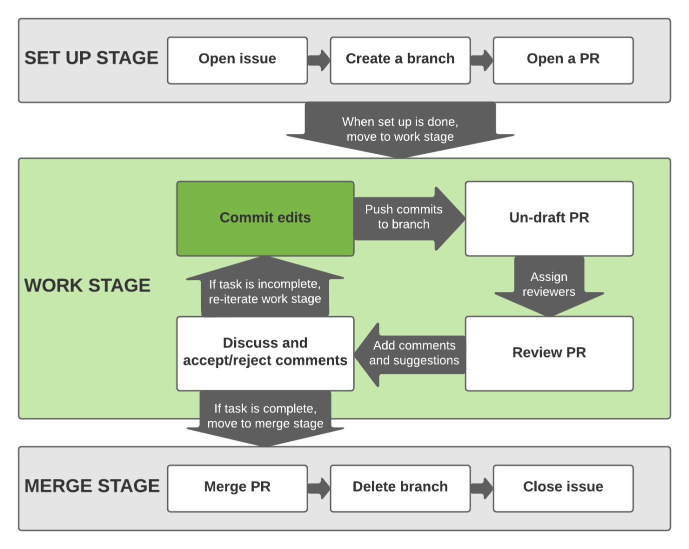
If task is complete, move to merge stage

MERGE STAGE

Merge PR → Delete branch → Close issue

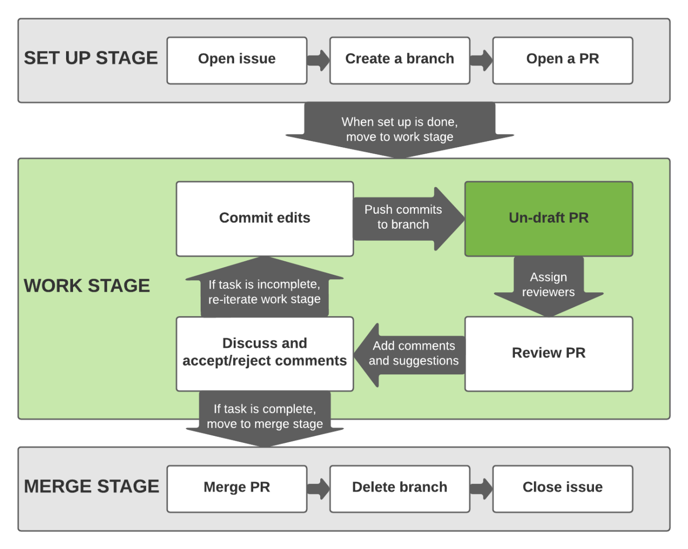
Make commits

- The author commits edits as usual
- The draft PR will be updated as more commits are pushed to the branch
- Keep pushing commits until the first implementation of the task is complete


SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
Review PR
Add comments and suggestions
Discuss and accept/reject comments
If task is incomplete, re-iterate work stage
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

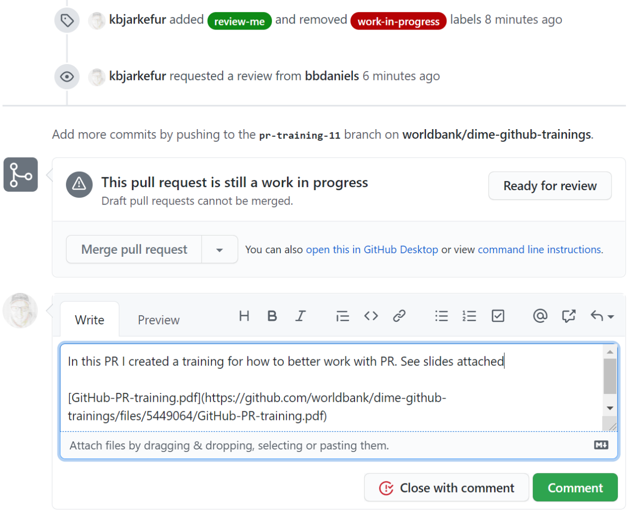
Un-draft the PR

- Document what has been implemented in the branch
- Un-drafting indicates that a PR is ready to review



SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
If task is incomplete, re-iterate work stage
Discuss and accept/reject comments
Add comments and suggestions
Review PR
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Un-draft the PR



kbjarkefur added review-me and removed work-in-progress labels 8 minutes ago

kbjarkefur requested a review from bbdaniels 6 minutes ago

Add more commits by pushing to the `pr-training-11` branch on worldbank/dime-github-trainings.

This pull request is still a work in progress

Draft pull requests cannot be merged.

In this PR I created a training for how to better work with PR. See slides attached

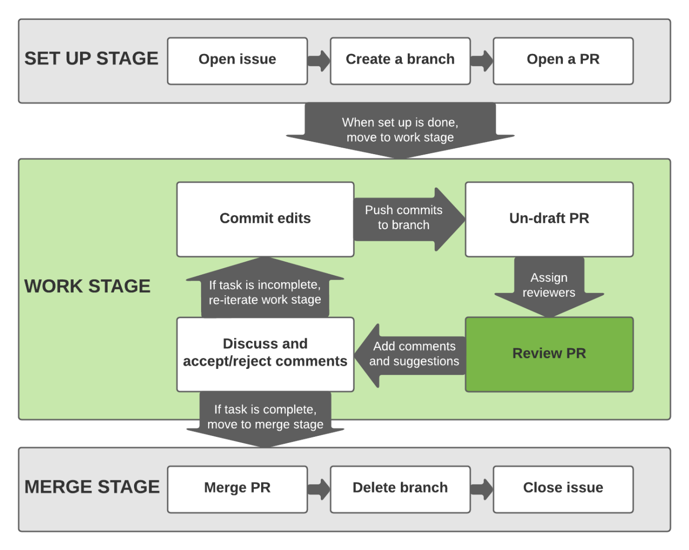
[GitHub-PR-training.pdf](https://github.com/worldbank/dime-github-trainings/files/5449064/GitHub-PR-training.pdf)

Ready for review
Merge pull request
You can also open this in GitHub Desktop or view command line instructions.
Write
Preview
Attach files by dragging & dropping, selecting or pasting them.
Close with comment
Comment

- Replace the *work-in-progress* label with *review-me*
- Assign one or several people to review your PR
- Notify the reviewer(s) using email or slack as GitHub notifications may be muted
- Click *“Ready for review”* to remove draft status

Review the PR

- PR reviews are incredibly important for high-quality code
- Use the PR to review code edits in the *file changes* tab
- The reviewer adds code suggestions and comments



SET UP STAGE

Open issue → Create a branch → Open a PR

When set up is done, move to work stage

WORK STAGE

Commit edits → Push commits to branch → Un-draft PR

Assign reviewers

Review PR → Add comments and suggestions → Discuss and accept/reject comments

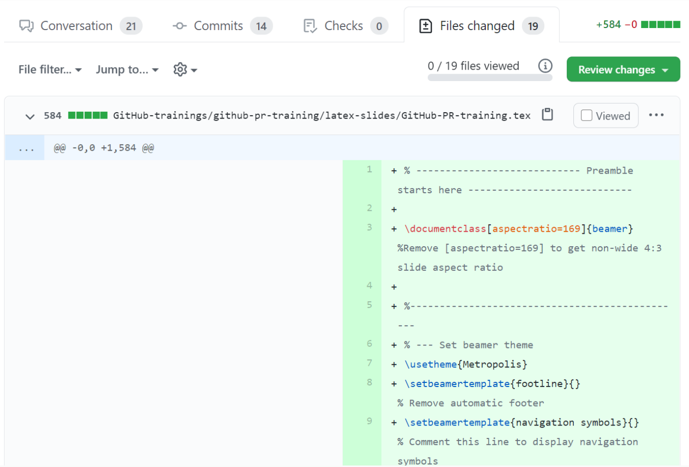
If task is incomplete, re-iterate work stage

If task is complete, move to merge stage

MERGE STAGE

Merge PR → Delete branch → Close issue

Review the PR



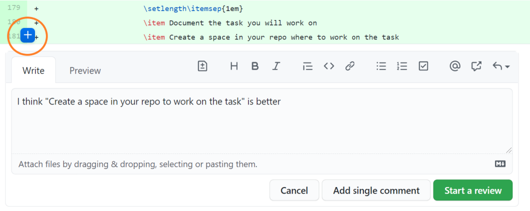
584 GitHub-trainings/github-pr-training/latex-slides/GitHub-PR-training.tex

```
@@ -0,0 +1,584 @@
+ % --------------------------- Preamble
  starts here ---------------------------
+
+ \documentclass[aspectratio=169]{beamer}
  %Remove [aspectratio=169] to get non-wide 4:3
  slide aspect ratio
+
+ %------------------------------------------
  ---
+ % --- Set beamer theme
+ \usetheme{Metropolis}
+ \setbeamertemplate{footline}{}
  % Remove automatic footer
+ \setbeamertemplate{navigation symbols}{}
  % Comment this line to display navigation
  symbols
```

- Start your review by reading what the author or anyone else have already said in the *"Conversation"* tab

- Then go to the *"Files changed"* tab - this is where you will review the new content of the files

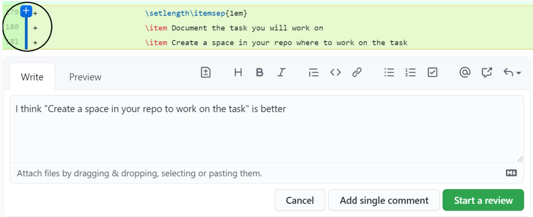
Line comments



```
+                  \setlength\itemsep{1em}
+                  \item Document the task you will work on
+                  \item Create a space in your repo where to work on the task
```

I think "Create a space in your repo to work on the task" is better

Attach files by dragging & dropping, selecting or pasting them.



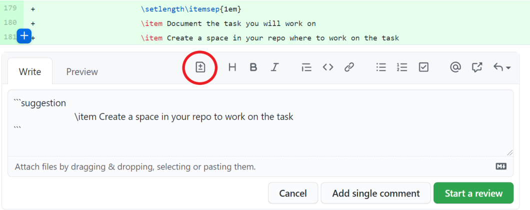
```
+                  \setlength\itemsep{1em}
+                  \item Document the task you will work on
+                  \item Create a space in your repo where to work on the task
```

I think "Create a space in your repo to work on the task" is better

Attach files by dragging & dropping, selecting or pasting them.

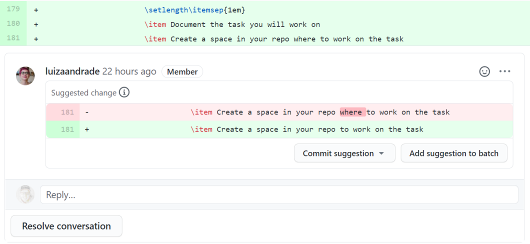
- Make comments on specific lines of as much as possible - line comments are much easier to keep track of
- Hover with the mouse over the line number for a line you want to comment on and click the plus sign (blue circle) to open comment field
- Hold shift when click and drag mouse to comment multiple lines (black circle)

Line suggestions



```
+                    \setlength\itemsep{1em}
+                    \item Document the task you will work on
+                    \item Create a space in your repo where to work on the task
```

~~~
```suggestion
               \item Create a space in your repo to work on the task
```
~~~



```
+                    \setlength\itemsep{1em}
+                    \item Document the task you will work on
+                    \item Create a space in your repo where to work on the task
```

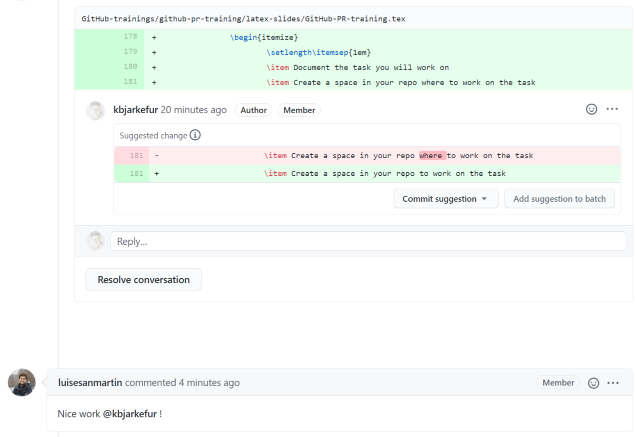
luizaandrade 22 hours ago Member

Suggested change

```
181 -                \item Create a space in your repo where to work on the task
181 +                \item Create a space in your repo to work on the task
```

- You can also suggest an edit directly to the code if the edit is not too big
- Click the suggestion button (red circle) and edit the line of code in the comment field
- The author will then see the exact difference in your suggestion
- Suggestions can be done to multi-line comments too and can be combined with regular comments

Overall comments



GitHub-trainings/github-pr-training/latex-slides/GitHub-PR-training.tex

```
+            \begin{itemize}
+                \setlength\itemsep{1em}
+                \item Document the task you will work on
+                \item Create a space in your repo where to work on the task
```

kbjarkefur 20 minutes ago Author Member

Suggested change

```
-                \item Create a space in your repo where to work on the task
+                \item Create a space in your repo to work on the task
```

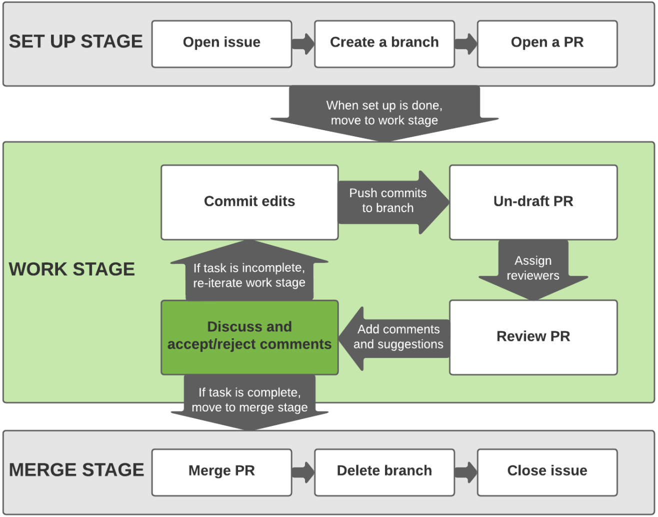
luisesanmartin commented 4 minutes ago Member

Nice work @kbjarkefur !

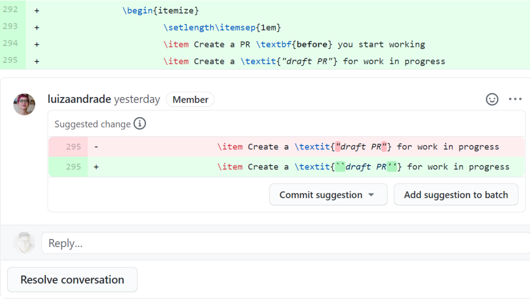
- Make overall comments and comments that cannot be done to a specific section of the code in thread in the *"Conversation"* tab
- Tag the author so they get a notification that someone have reviewed the PR
- Suggestions are also listed in this tab

Discuss and accept/reject

- The author discusses, accepts or rejects suggestions and replies to comments
- The author, the reviewer(s) and/or anyone else in the project team then decide if the work stage needs to be repeated


SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
Review PR
Add comments and suggestions
Discuss and accept/reject comments
If task is incomplete, re-iterate work stage
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Review suggestions



```
+             \begin{itemize}
+                     \setlength\itemsep{1em}
+                     \item Create a PR \textbf{before} you start working
+                     \item Create a \textit{"draft PR"} for work in progress
```

luizaandrade yesterday Member

Suggested change

```
-                     \item Create a \textit{"draft PR"} for work in progress
+                     \item Create a \textit{``draft PR''} for work in progress
```

- Respond to code suggestions before comments as suggestions tend to be quicker to address
- Suggestions and line comments are listed chronologically in the *"Conversation"* tab and by line number in the *"File changes"* tab
- Accept the suggestion with *Commit Suggestion*, ignore/delete it with *Resolve conversation* or comment to ask for more information

Approve PR

- When all suggestions and comments have been responded to, the team should discuss if a task is done or need more work
- If major work is still required, revert the PR to draft status, add back the *work-in-progress* label and iterate the work stage again
- Move on to the merge stage only when the task is complete and the team agrees the code is high-quality

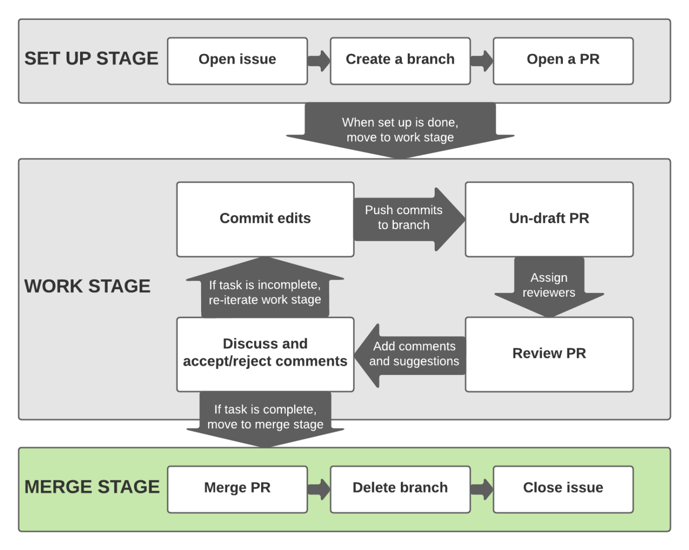
Stage: The merge stage

MERGE STAGE

Stage: The merge stage

Purpose of the merge stage:

- Merge the quality controlled PR
- Make sure the task is properly documented and clean up the repo



SET UP STAGE

Open issue

Create a branch

Open a PR

When set up is done, move to work stage

WORK STAGE

Commit edits

Push commits to branch

Un-draft PR

Assign reviewers

Review PR

Add comments and suggestions

Discuss and accept/reject comments

If task is incomplete, re-iterate work stage

If task is complete, move to merge stage

MERGE STAGE

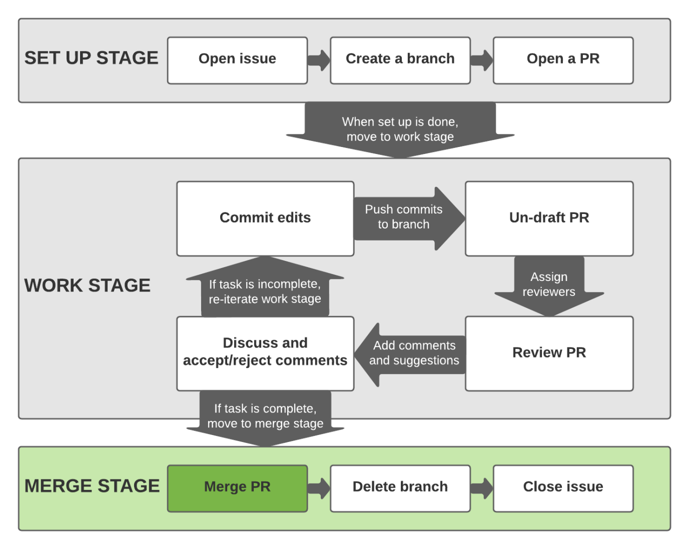
Merge PR

Delete branch

Close issue

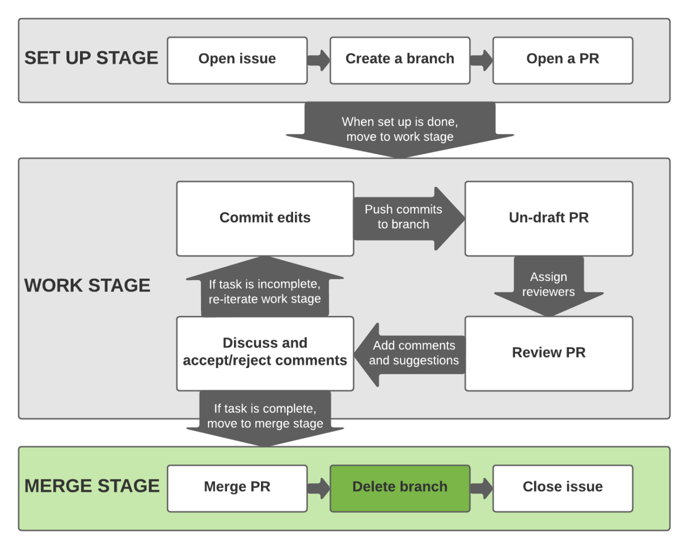
Merge the PR

- Make sure that the PR is well documented - this is where future team members will read how something was done and why
- Then you can merge your well-reviewed and well-documented code
- **Test run your code after merging it!**


SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
If task is incomplete, re-iterate work stage
Assign reviewers
Discuss and accept/reject comments
Add comments and suggestions
Review PR
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Delete the branch

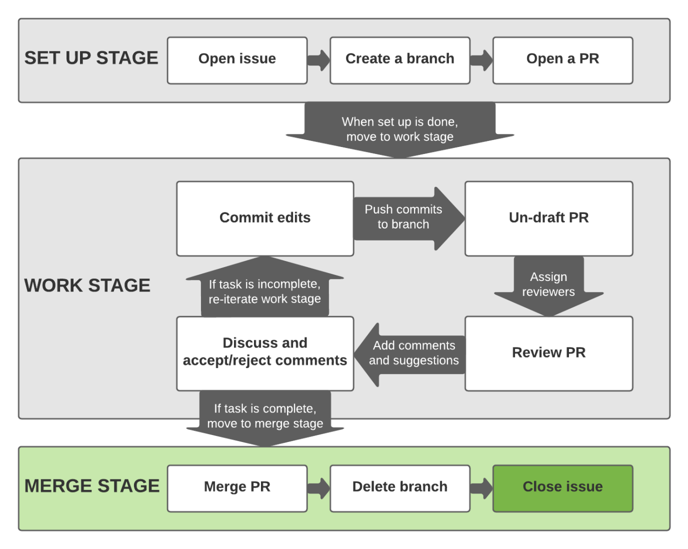
- Always delete branches after they are merged
- If you want to keep working in a branch with the same name, then re-create that branch
- Otherwise you risk working on an outdated version of the repo


SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
Review PR
Add comments and suggestions
Discuss and accept/reject comments
If task is incomplete, re-iterate work stage
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Close issue

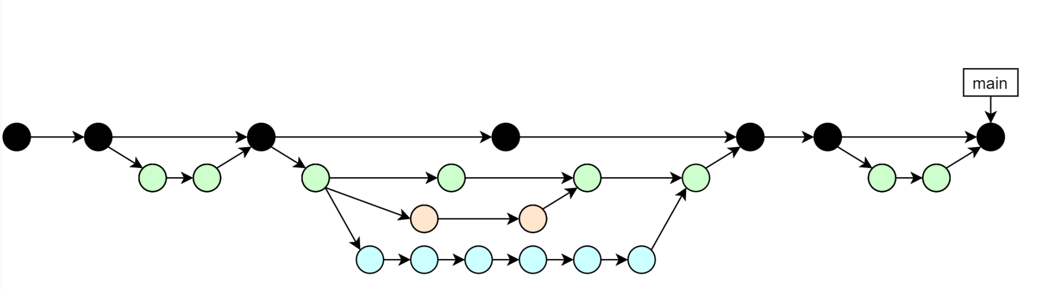
If you created an issue for this task:

- Make sure that the PR number is referenced in the issue
- For example: *"Issue resolved in PR #12"*
- Then close the issue


SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
Review PR
Add comments and suggestions
Discuss and accept/reject comments
If task is incomplete, re-iterate work stage
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Part 2:
Gitflow - How to fit the “branch-PR-merge” cycle into your workflow

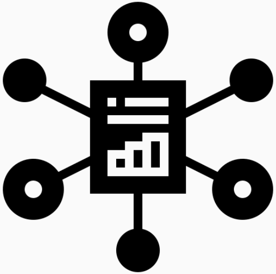
The network graph



- A typical git network graph - each dot is a commit
- Several branches have been opened and merged, but currently only the *main* branch is open

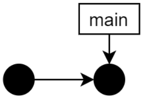
Gitflow

- **Gitflow** is an idea or a philosophy of how to organize work in git - it is not a software you install
- It is developed by computer scientists and you will find a lot of resources for it online
- We will present a simplified version adapted for research
- Two branch types (later we add a third type):
  - The **main branch** (formerly master branch) - should never be worked in directly
  - **Feature branches** - this is where you do all your work



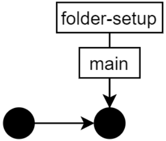
Simple branch-PR-merge cycle

Start a branch-PR-merge cycle



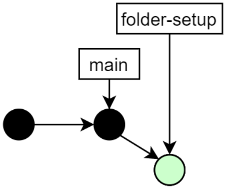
- This can either be the beginning of a repo or at a point where all previous branches were merged to the main branch
- Let's say your next task is to set up folders - apply the *branch-PR-merge* cycle

Set up stage - 1



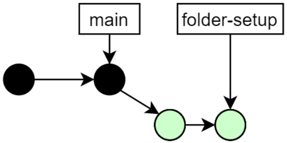
- Create a new *feature* branch at the point where you want the the *branch-PR-merge* cycle to start from
- Two branches pointing to the same commit are identical by definition

Set up stage - 2



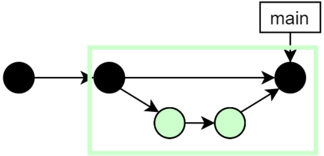
- Create a new short README or quickly edit an existing and commit - (or make a quick commit for the task) - at least one commit is needed to follow this workflow
- Open up a draft PR so there is a good place for your team to follow your work

Work stage



- Complete the task using one or many commits - in this example only one more commit was needed
- Assign someone to review the PR - iterate until you are all satisfied

Merge stage

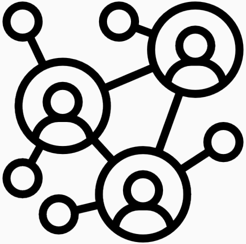


- Merge the PR and delete the *feature* branch
- The green box represents a full *branch-PR-merge* cycle in the network graph

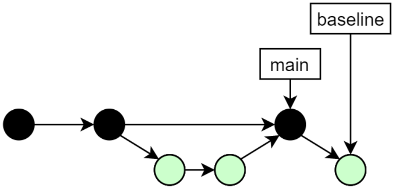
Nested branch-PR-merge cycles

Nested branch-PR-merge cycles

- Many high-level tasks are not simple - multiple team members working over multiple months
- Cleaning the baseline data, for example, is one big task composed of many smaller tasks
- Solve this by using nested *branch-PR-merge* cycles
- Third type of branch: **Develop branch** - a branch that is a high-level task that will have several *feature* branches

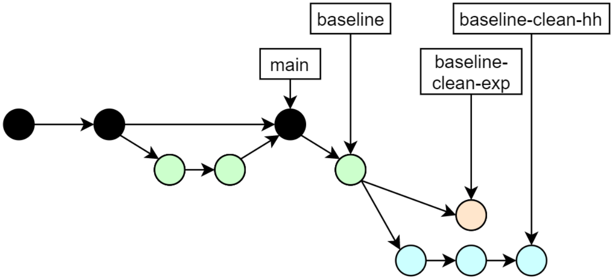


Create a develop branch



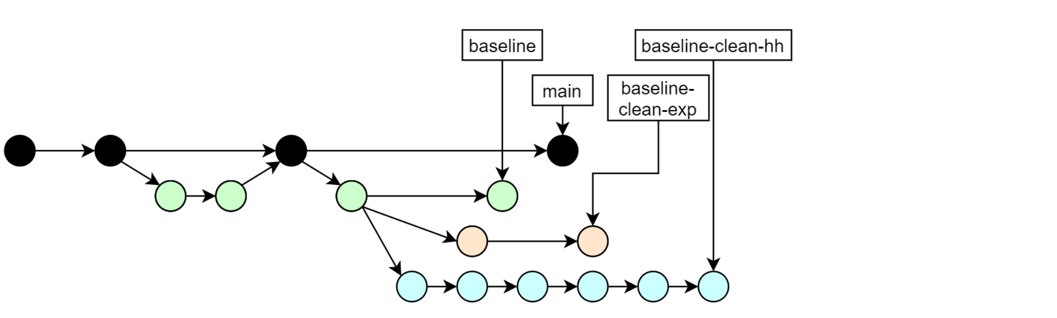
- Create a *develop* branch - name it after the high-level task - complete set up stage
- *Develop* branches often do not have a Github issue - but the *feature* branches off of it often have corresponding GitHub issues
- Include *develop* branch name as prefix to all *feature* branches off it

Create feature branches of the develop branch



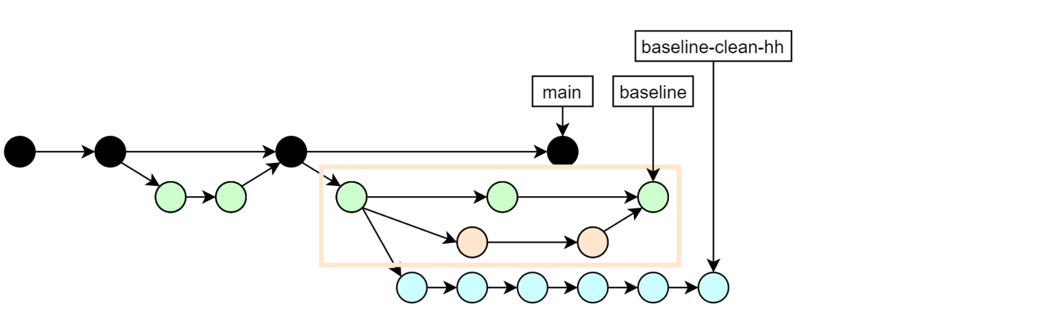
- Create as many *feature* branches off the *develop* branch as you need
- Split up the high-level task in as many smaller tasks as needed - try to split up tasks until each is small enough for one person to complete in max a week or two
- Create new *feature* branches off the *develop* branch even after it progresses

Work in the branches



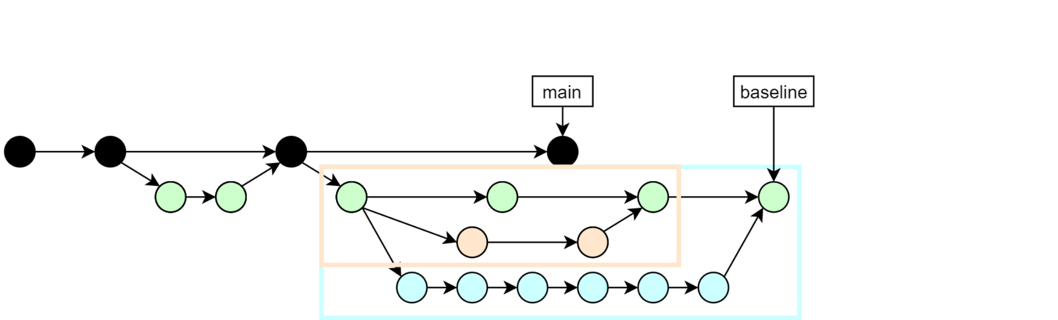
- Keep working on the *feature* branches until they are done
- When a task is complete: assign a reviewer - make comments/suggestions - accept/reject/discuss

Merge a feature branch



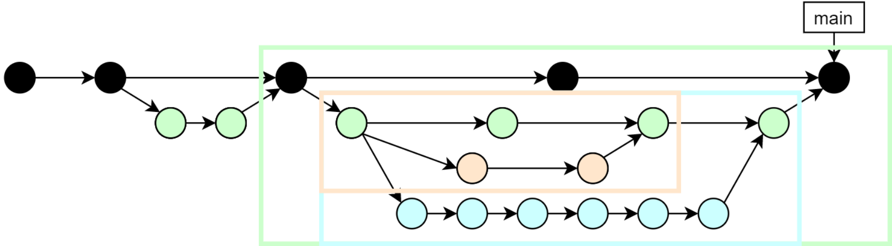
- The branch `baseline-clean-exp` was approved in the last step in the work stage and was merged and deleted in the merge stage
- The orange box is one *branch-PR-merge* cycle

Merge a feature branch



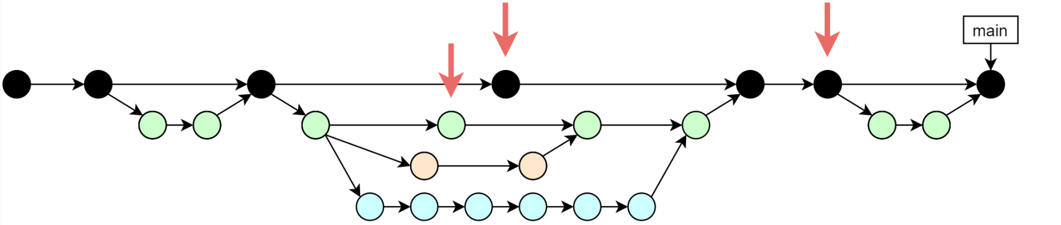
- The branch `baseline-clean-hh` was then also approved in the last step in the work stage and was merged and deleted in the merge stage
- The blue box is also a *branch-PR-merge* cycle

Merge a develop branch



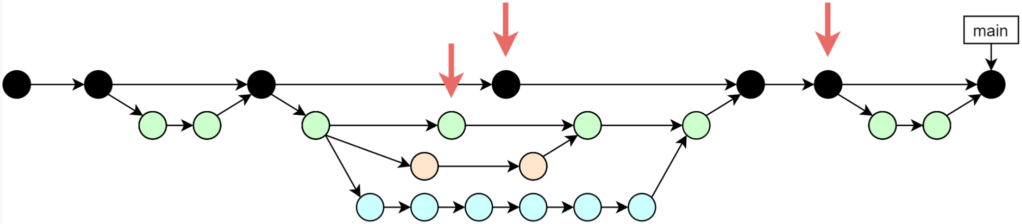
- Then the branch `baseline` is ready for a review - assign a reviewer - comment/suggest - accept/reject/discuss
- The green box is also a *branch-PR-merge* cycle
- This network graph shows a nested *branch-PR-merge* cycle

Work directly in main/develop



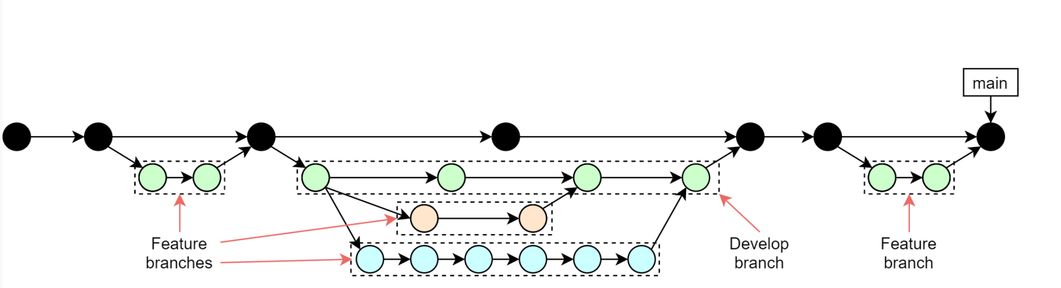
- When could it be ok to work directly on a *main/develop* branch? - see red arrows

Work directly in main/develop



- When could it be ok to work directly on a *main*/*develop* branch? - see red arrows
- When updating documentation!
  - Documentation about the repo in the *main* branch
  - Documentation about the high-level task in the *develop* branch

The network graph



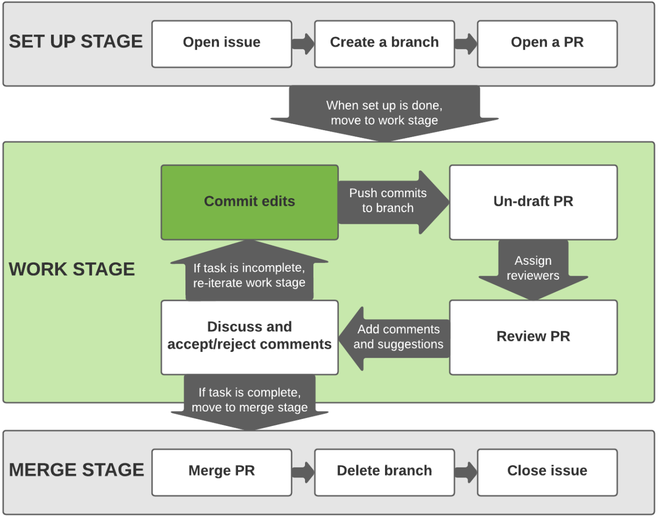
- There is no technical git difference between the Gitflow types of branches - the only difference is what workflow that is expected
  - **Main branch**: The branch where all branches originates from
  - **Develop branch**: Non-*main* branch with *feature* branches
  - **Feature branch**: Branches of the *main* branch or *develop* branches with no other branches from it

Nest cycles in the work stage

- *Branch-PR-merge* cycles should be nested in the work stage

- The *develop* branch is kept in the *Commit edits* step while the *branch-PR-merge* cycles are completed for the feature branches

- If needed, convert a *feature* branch to a *develop* branch here


SET UP STAGE
Open issue
Create a branch
Open a PR
When set up is done, move to work stage
WORK STAGE
Commit edits
Push commits to branch
Un-draft PR
Assign reviewers
Review PR
Add comments and suggestions
Discuss and accept/reject comments
If task is incomplete, re-iterate work stage
If task is complete, move to merge stage
MERGE STAGE
Merge PR
Delete branch
Close issue

Thank you!

Questions?

Slides can be found here: `https://osf.io/e54gy/`

Useful links

- All DIME Analytics GitHub trainings: https://osf.io/e54gy/
- Other DIME Analytics GitHub resources: https://github.com/worldbank/dime-github-trainings. For example:
  - DIME Analytics GitHub Templates (for example .gitignore): https://github.com/worldbank/dime-github-trainings/tree/master/GitHub-resources/DIME-GitHub-Templates
  - DIME Analytics GitHub Roles: https://github.com/worldbank/dime-github-trainings/blob/master/GitHub-resources/DIME-GitHub-Roles/DIME-GitHub-roles.md
- Markdown cheat sheet (how to format text on GitHub.com): https://www.markdownguide.org/cheat-sheet/
- DIME GitHub Account admin info and instructions: https://github.com/dime-worldbank/dime-account-admin

LaTeX/Git Version control

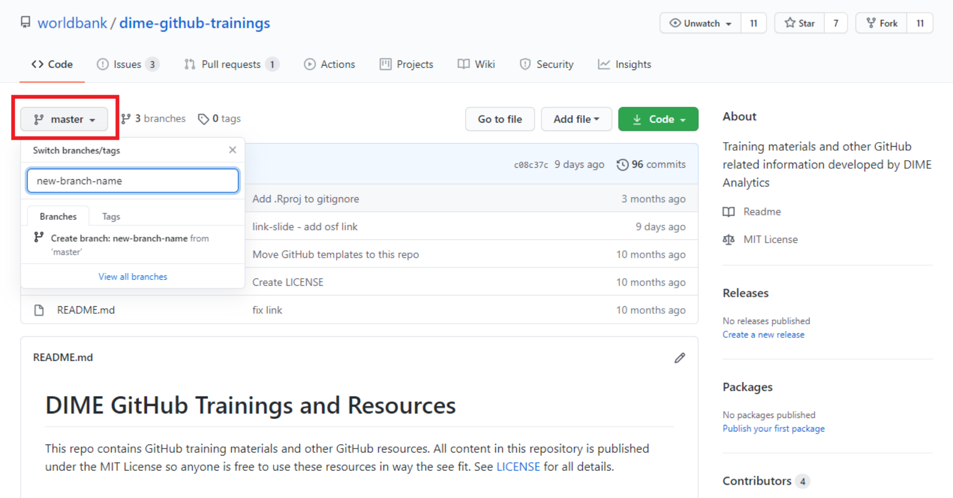
Most recent commit at the point of compiling this LaTeX Beamer presentation:

`https://github.com/worldbank/dime-github-trainings/commit/40a95f94162e500dc0d10c38ede0ed4cbe43877b`

For instructions on how to include URLs to most recent commit used to generate document in Git, see: `https://gist.github.com/kbjarkefur/88820e5a5365b3f707b6b20aee57cf8a`

Appendix

Create branch on GitHub.com



Add .Rproj to gitignore	3 months ago
link-slide - add osf link	9 days ago
Move GitHub templates to this repo	10 months ago
Create LICENSE	10 months ago
fix link	10 months ago

About

Training materials and other GitHub related information developed by DIME Analytics

Readme

MIT License

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

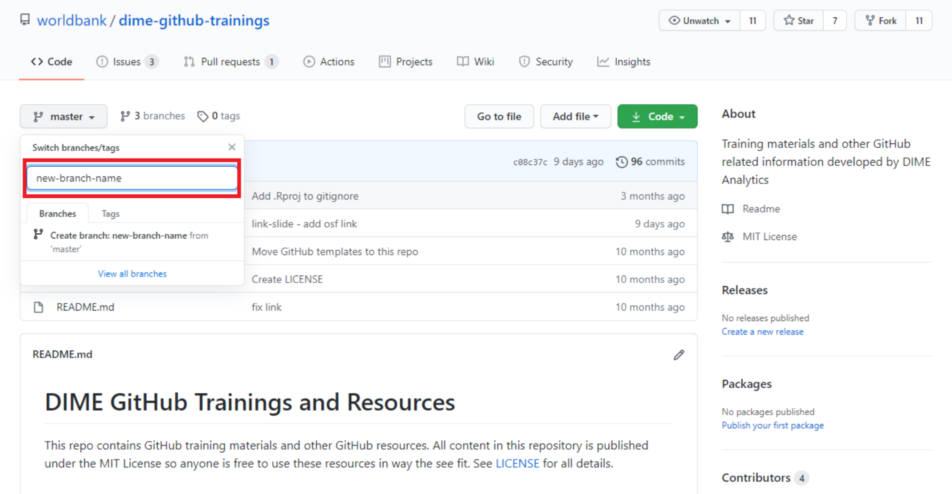
Contributors 4

README.md

DIME GitHub Trainings and Resources

This repo contains GitHub training materials and other GitHub resources. All content in this repository is published under the MIT License so anyone is free to use these resources in way the see fit. See LICENSE for all details.

Create branch on GitHub.com



worldbank / dime-github-trainings

new-branch-name

Create branch: new-branch-name from 'master'

	Add .Rproj to gitignore	3 months ago
	link-slide - add osf link	9 days ago
	Move GitHub templates to this repo	10 months ago
	Create LICENSE	10 months ago
README.md	fix link	10 months ago

About

Training materials and other GitHub related information developed by DIME Analytics

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

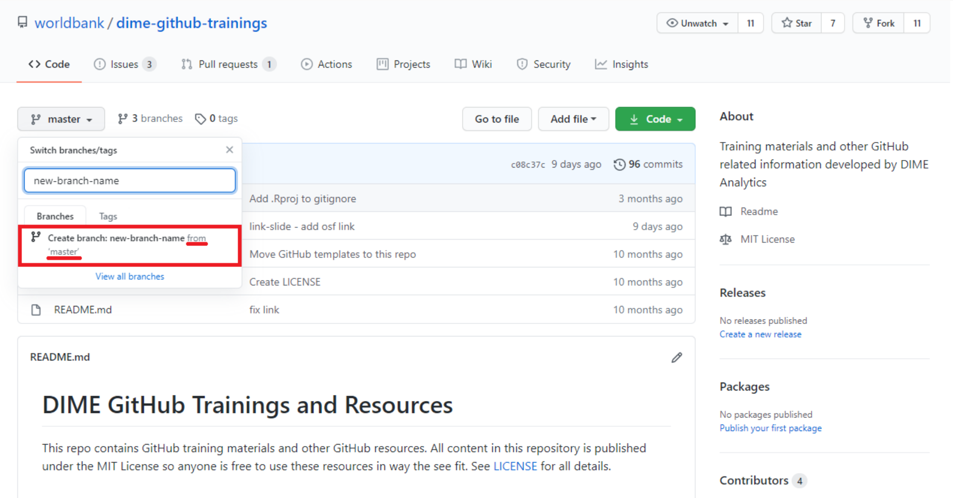
Contributors 4

README.md

DIME GitHub Trainings and Resources

This repo contains GitHub training materials and other GitHub resources. All content in this repository is published under the MIT License so anyone is free to use these resources in way the see fit. See LICENSE for all details.

Create branch on GitHub.com



new-branch-name

Create branch: new-branch-name from 'master'

c08c37c 9 days ago 96 commits

Add .Rproj to gitignore 3 months ago

link-slide - add osf link 9 days ago

Move GitHub templates to this repo 10 months ago

Create LICENSE 10 months ago

README.md fix link 10 months ago

About

Training materials and other GitHub related information developed by DIME Analytics

Readme

MIT License

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 4

README.md

DIME GitHub Trainings and Resources

This repo contains GitHub training materials and other GitHub resources. All content in this repository is published under the MIT License so anyone is free to use these resources in way the see fit. See LICENSE for all details.