(master)

GitHub Flow for DPL

v1.0.1

TYPE-feature-description
(branch)
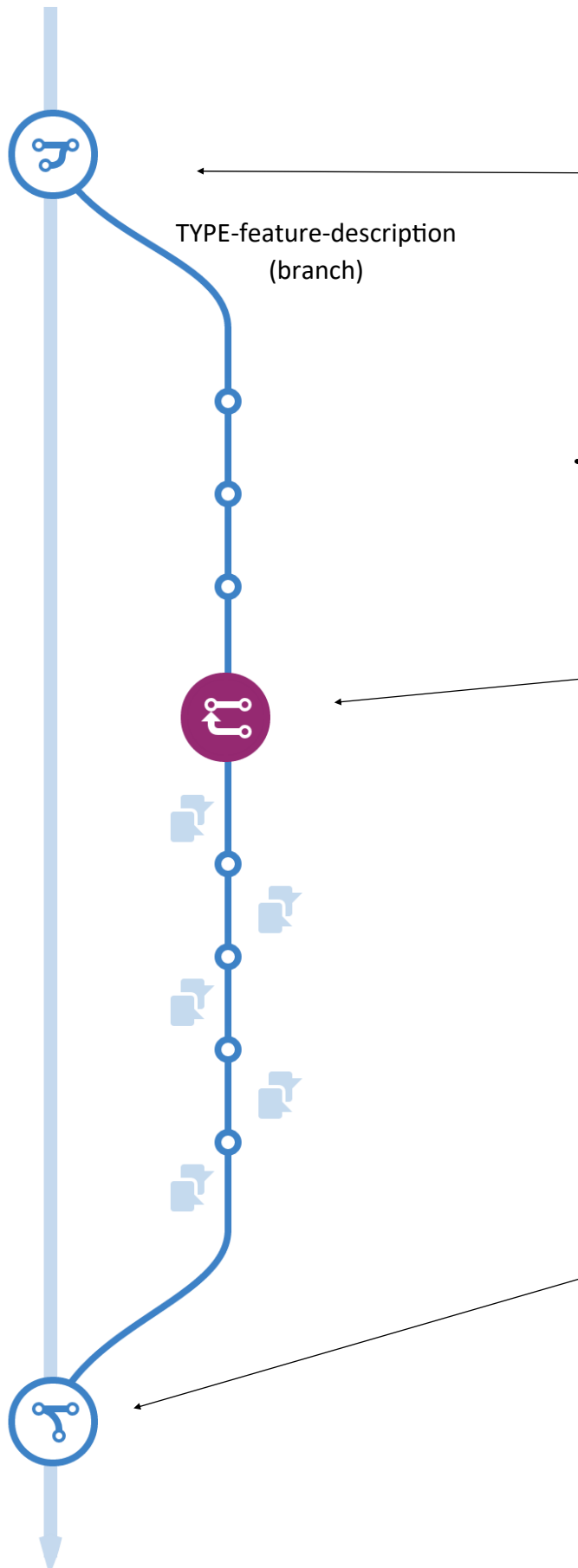
1. Create new branch

1. Checkout master
2. Pull (to get latest changes)
3. Create a new branch
   (`TYPE-feature-description`)

2. Build the feature

1. `CONTRIBUTING.md` instructions
2. Acceptance criteria (feature specifics)
3. Definition of done
4. Update Trello card comments
5. Test in browsers and devices
6. Update change log record in Trello card

3. Commit to Git

1. See commit message style guide

4. Open a pull request

1. See commit message style guide
2. Push to GitHub
3. Update Trello card with link to GitHub
   pull request. Move card to QA.

5. QA code

1. Checkout branch
2. Build (grunt)
3. Review code and documentation against
   acceptance criteria and definition of done
4. (Edit / commit / push to GitHub)
5. Test in production (`-alpha`, `-rc1`)

6. Update version and change log

1. Bump version number (`v1.2.3`)
2. Update change log from Trello.
3. Build (`grunt`)
4. Commit and push to GitHub

7. Merge

1. Merge branch into master
   (See commit message style guide)
2. Deploy `/core` to `/cdn/x.y.z`
3. Deploy `/docs` to `/dpl/x.y.z`

8. Tag and clean-up

1. Checkout master and build (`grunt`)
2. Tag master (`v1.2.3`)
3. Delete branches (local and remote)
4. Close any GitHub issues for this feature
5. Move Trello card to Done
6. Update in T4 (HTML code and DPL
   version)

Image source: https://guides.github.com

Commit message style guide cheat sheet

https://github.com/standrewsdigital/digital-code-style-guide

```
TYPE (Component) Brief summary

Longer description with more details, such as a `code` being introduced

within the context of a `function()`.

Multiple paragraphs may be added as required.

Bulleted lists:

* are

* also

* okay

Closes #123
```

TYPE	Description
BREAK	A breaking change such as removing a feature.
DOCS	Changes to the documentation (readme, API docs, etc.).
FEAT	New/updated feature in production code.
FIX	Bug fix in production code.
FORMAT	Code formatting, code comment change, etc; (compiled-code neutral).
MAINT	Updating dev-related maintenance ("non-production code") files.
MERGE	Pull request to merge branch into another branch or master.
TEST	Adding missing tests or editing tests.

Example commit message:

```
FEAT (ScrollSpy) Add ScrollSpy into long-form-content pattern


ScrollSpy was previously removed to reduce file size but is

now required for a number of projects. This has been added

back in, including full documentation.


Closes #32
```