EBU6305

Interactive Media Design and Production

Production and Management

Dr. Xianhui Che

x.che@qmul.ac.uk

Queen Mary
University of London

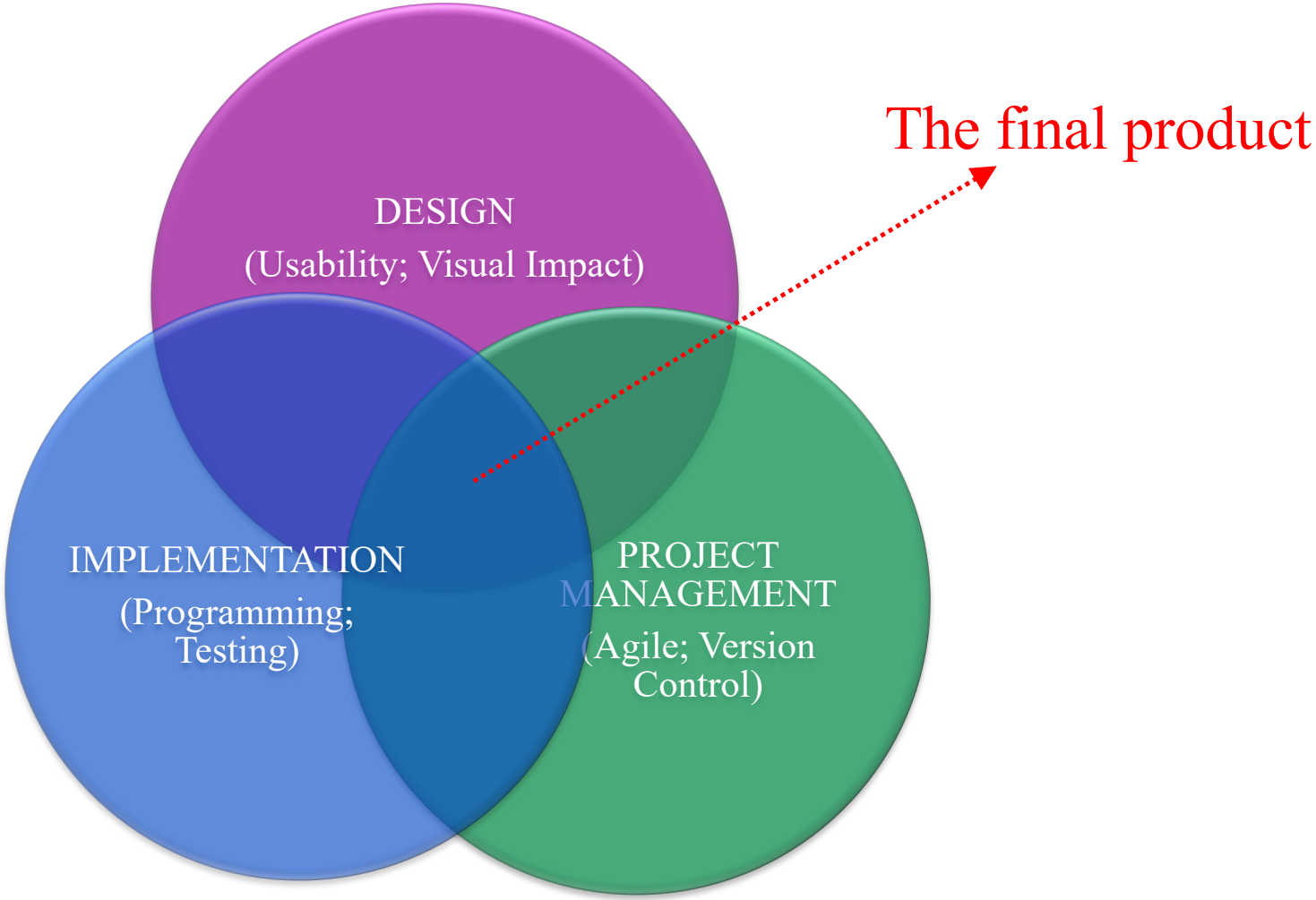
Learning Objectives

- Manage the design process and evaluate outcomes.
- The ability to reflect upon and assess their own progress.
- Get familiar with common good practice of production management in the industry

Topics

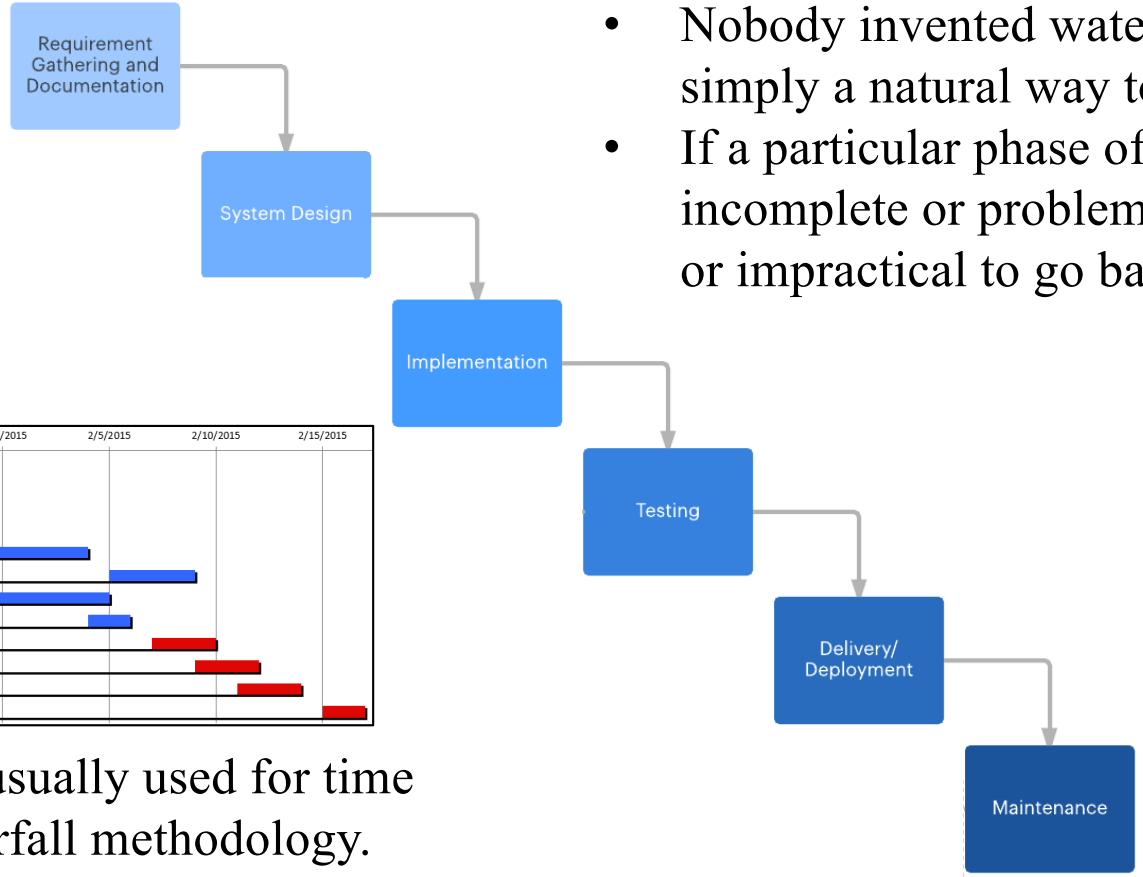
- Agile Management
- Version Control

Production Process

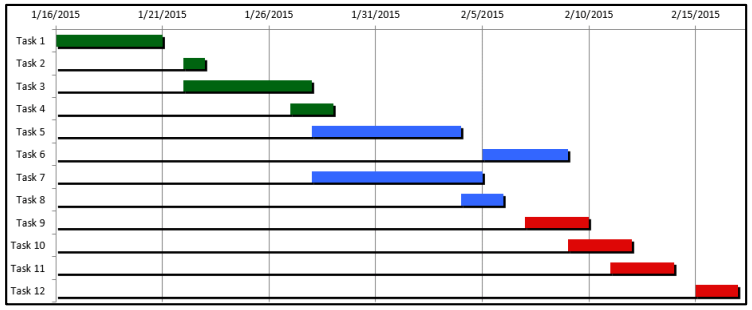


Traditional Production

- Waterfall management allows the production to progress phase by phase.



- Nobody invented waterfall method, which is simply a natural way to proceed.
- If a particular phase of production is incomplete or problematic, it would be costly or impractical to go back and make changes.

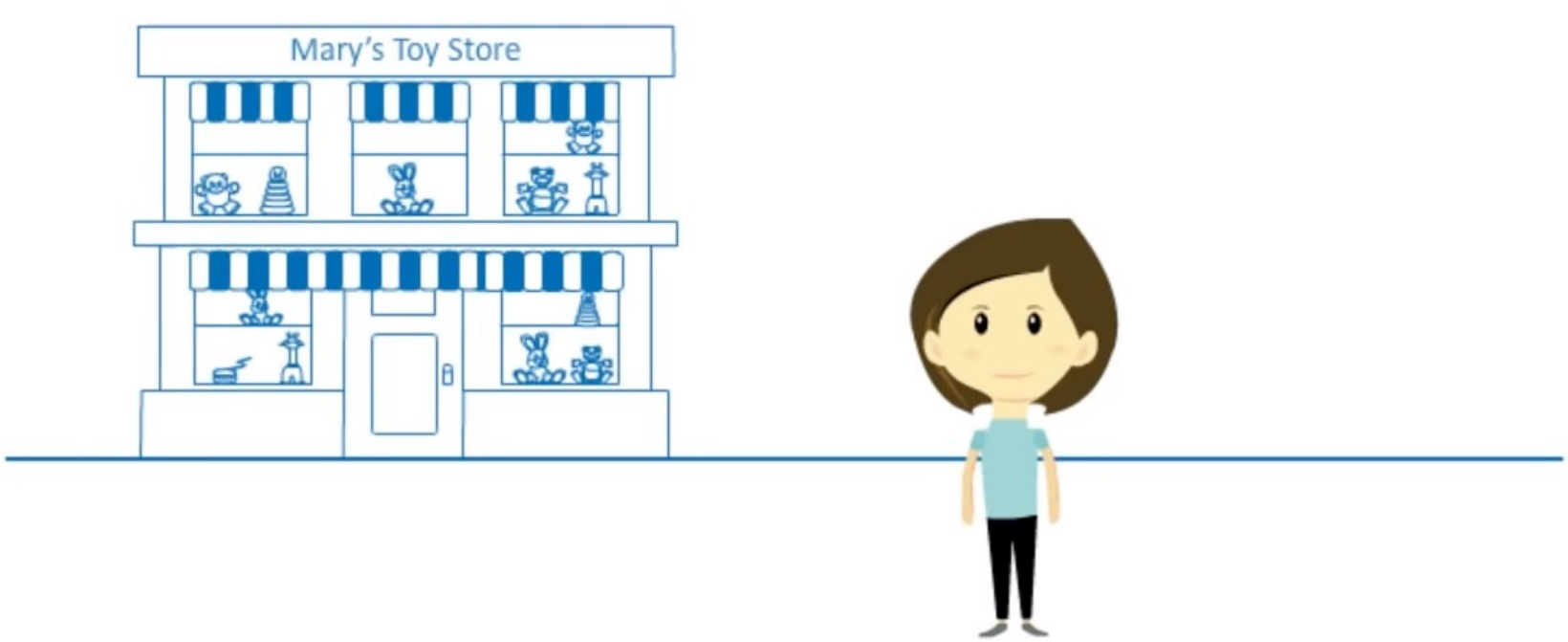


Gantt charts are usually used for time planning in Waterfall methodology.

The Unified Process

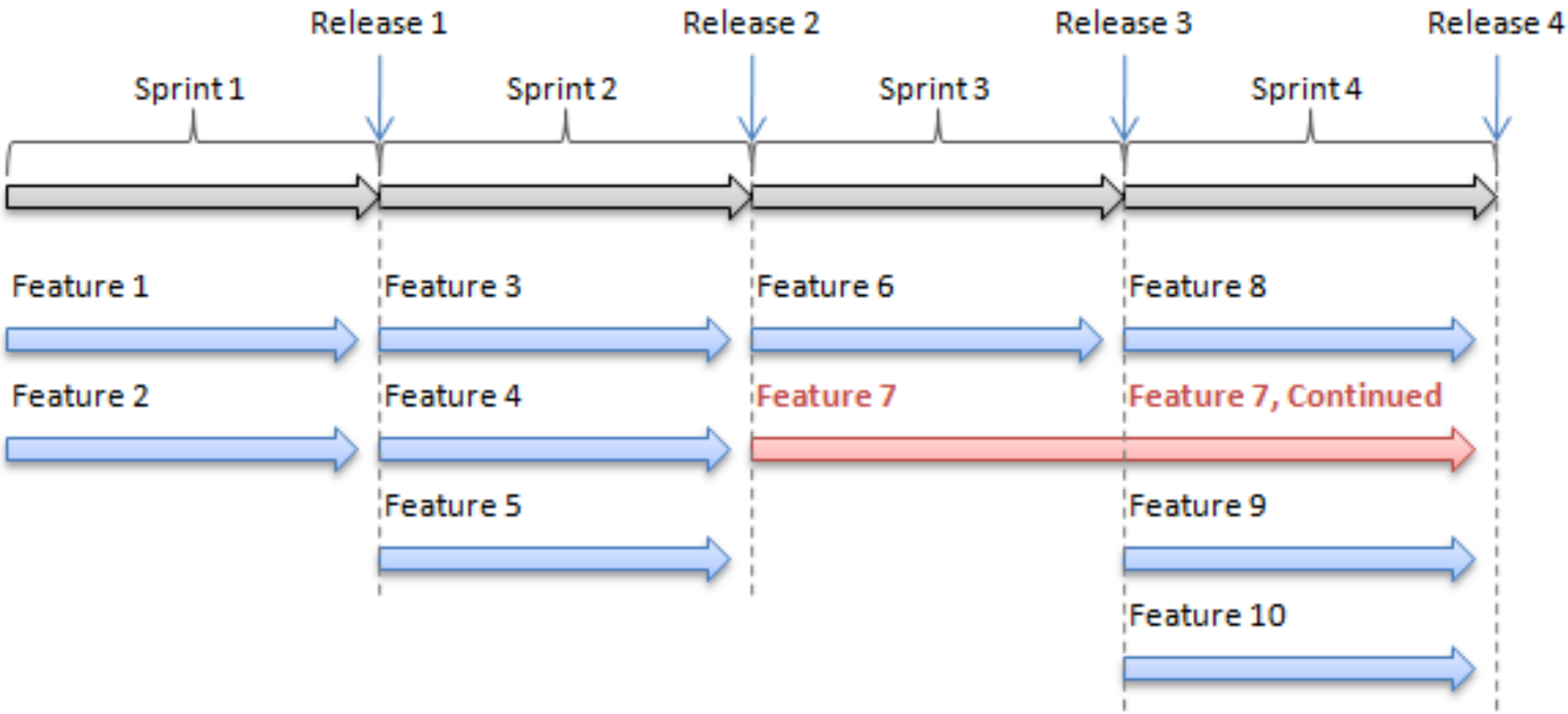
- Iterative and Incremental.
  - Commercial projects continue many months and years.
  - To be most effective - break the project into iterations.
- Every iteration - identify use cases, create a design, and implement the design .
- Every iteration is a complete development process.
- This methodology is called Agile.

Introductory Video for Agile



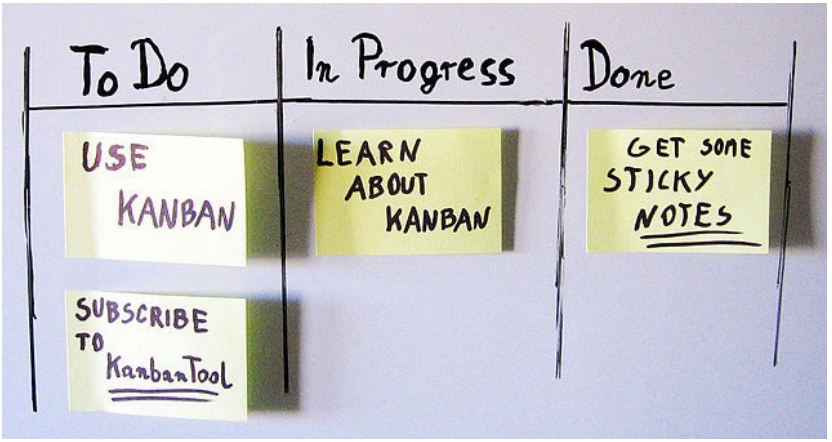
Video modified based on CA Agile.

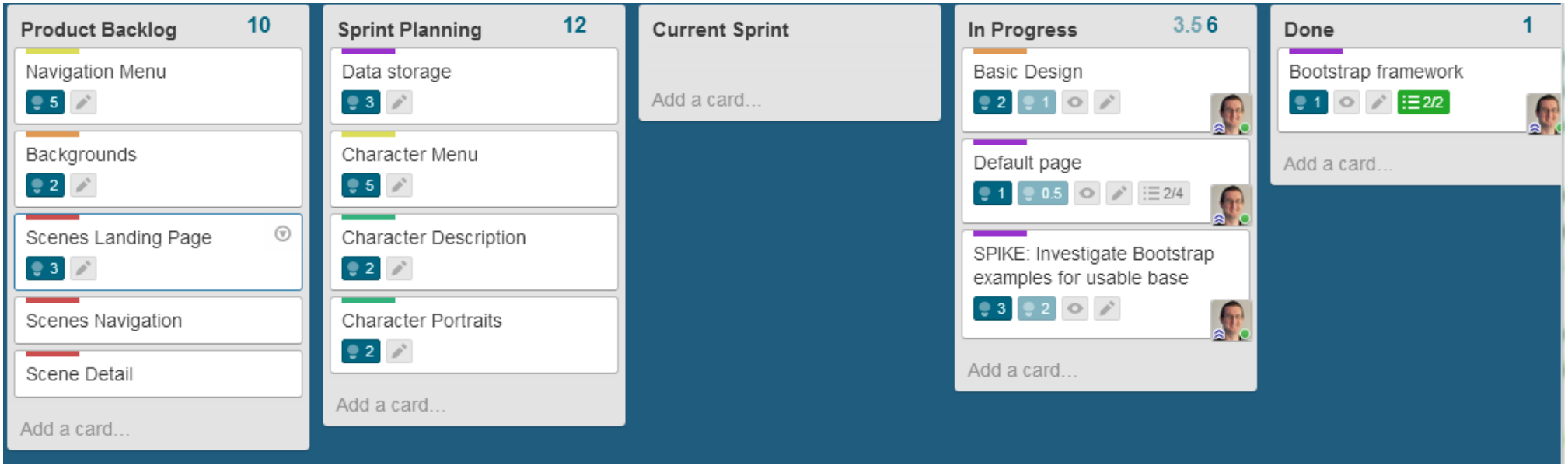
Agile Project Management



Kanban

- Recommended tool: **Trello** – http://www.trello.com
  (Free)
- Kanban can be used to plan anything, even a birthday party!

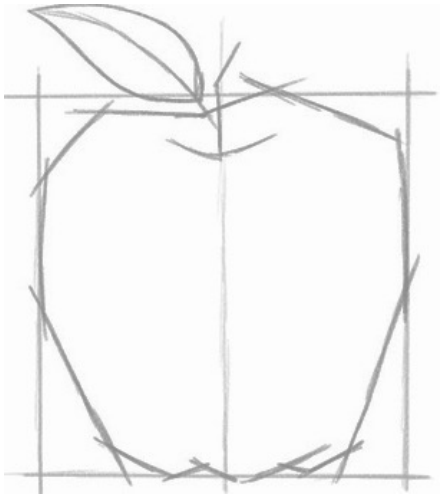




Queen Mary
University of London

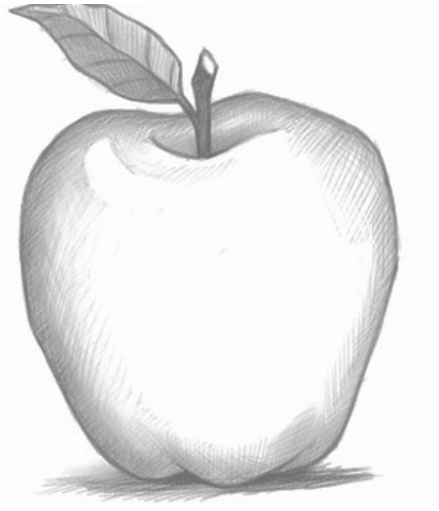
Don’t be too ambitious!

- The obvious advantage of Agile is you can have a functional product in the shortest period.
- Always start from something simple.



Sprint 1

It’s rough, but it works!



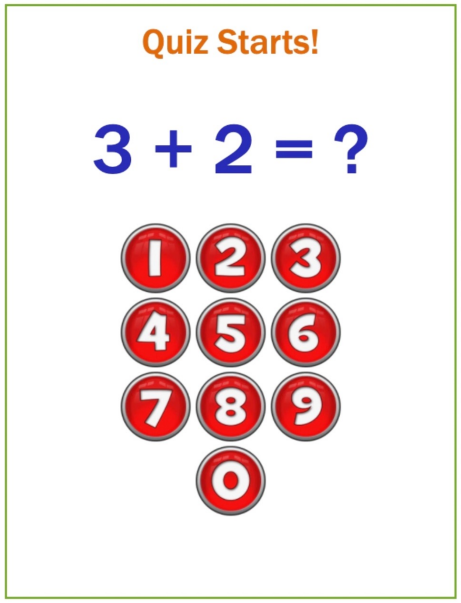
Sprint N

After many iterations, it has become a fine product.

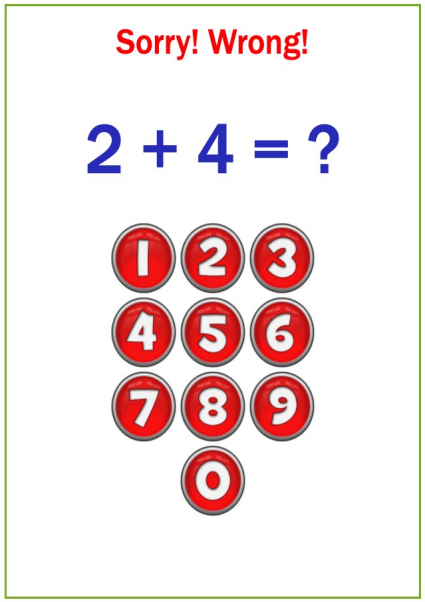
Benefits of Agile

- Agile can transfer uncertainty to certainty in a very short time period.
- Microscopically, Agile dynamically allocates and updates resources. Tasks are reviewed daily (or as frequent as it should be).
- Macroscopically, productions are usually completed sooner than the deadline that is initially set.

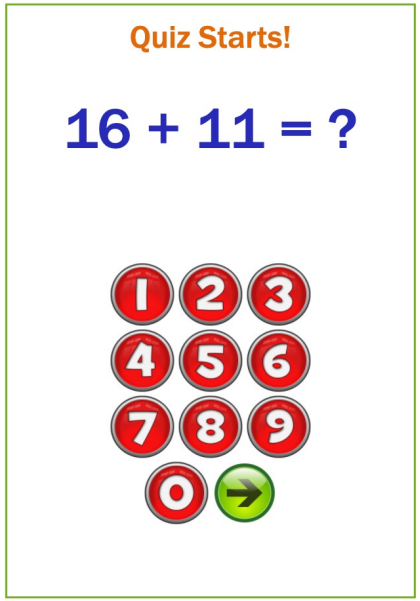
Your Agile Practice in Labs



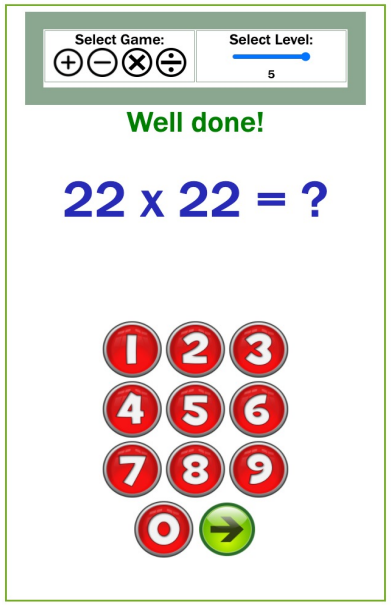
Lab 1
1st Sprint
Version 1



Lab 2
2nd Sprint
Version 2



Lab 3
3rd Sprint
Version 3



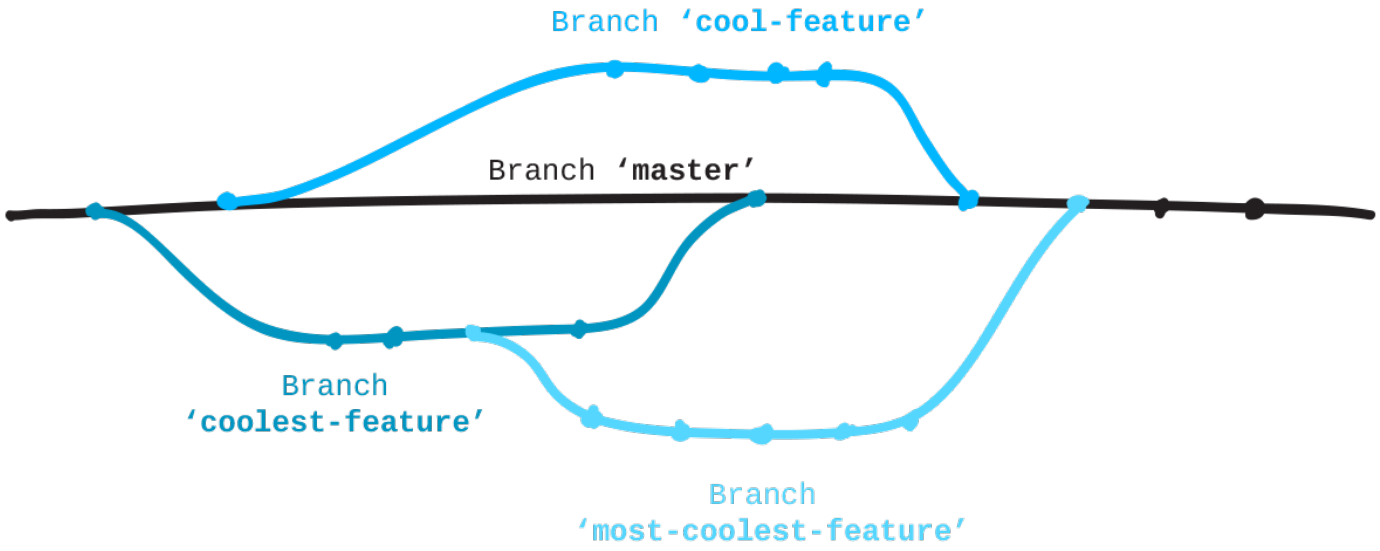
Lab 4
4th Sprint
Version 4

Topics

- Agile Management
- Version Control

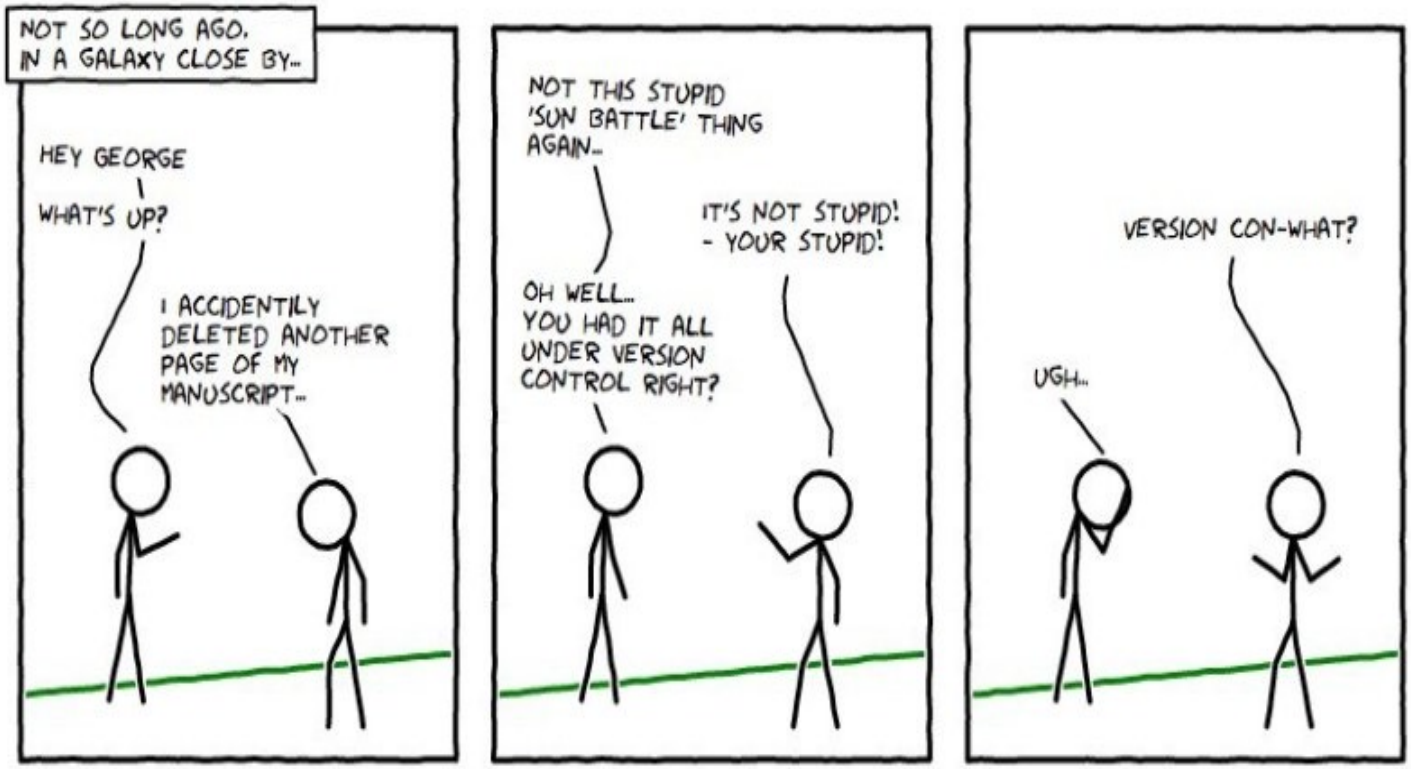
Version Control

- Recommended tool: **GitHub** – http://www.github.com
  - It is one of the most established version control platforms nowadays.



Benefits of Version Control

- Tracks code history
- Allows for human error

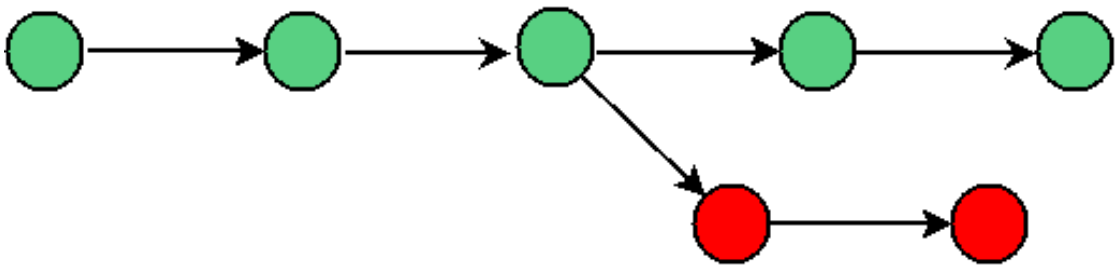


More Reasons to Use GitHub

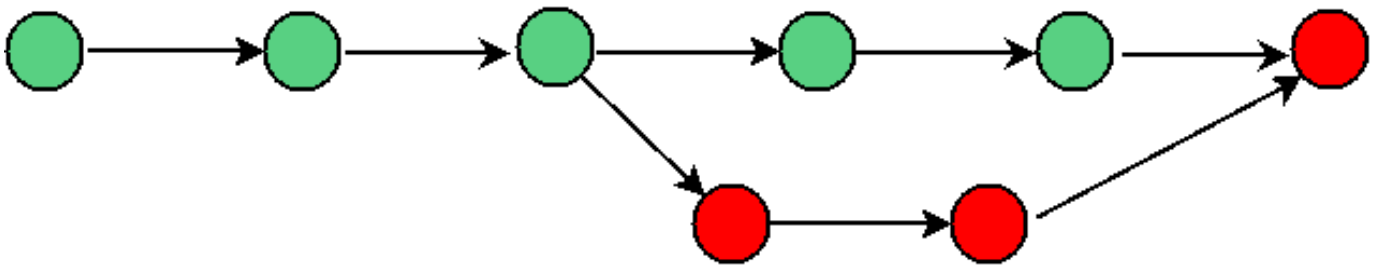
- Collaborative/social coding
- Code storage in the cloud
- Guaranteed to retrieve what people put in (GitHub and Trello do not lie!)
- Optimized network transfers

Project Development

Project Development



Project Development

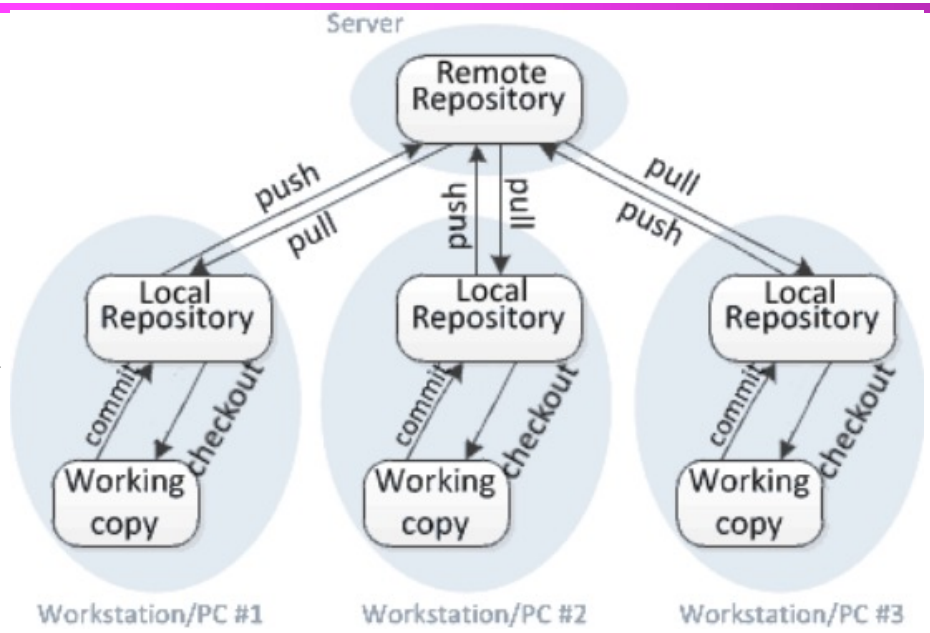


Git/GitHub Terminologies

- **Repository:** is the most basic element of Git. They are easiest to imagine as a project folder. A repository contains all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.
- **Clone:** is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. It is, however, connected to the remote version so that changes can be synced between the two. You can push your local changes to the remote to keep them synced when you're online.
- **Remote:** is the version of something that is hosted on a server, e.g. GitHub. It can be connected to local clones so that changes can be synced.
- More GitHub glossary can can be found at https://help.github.com/articles/github-glossary/

Git/GitHub Terminologies

- **Commit:** is an individual change to a file (or set of files). Every new change is associated with a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by who. Commits usually contain a commit message which is a brief description of what changes were made.

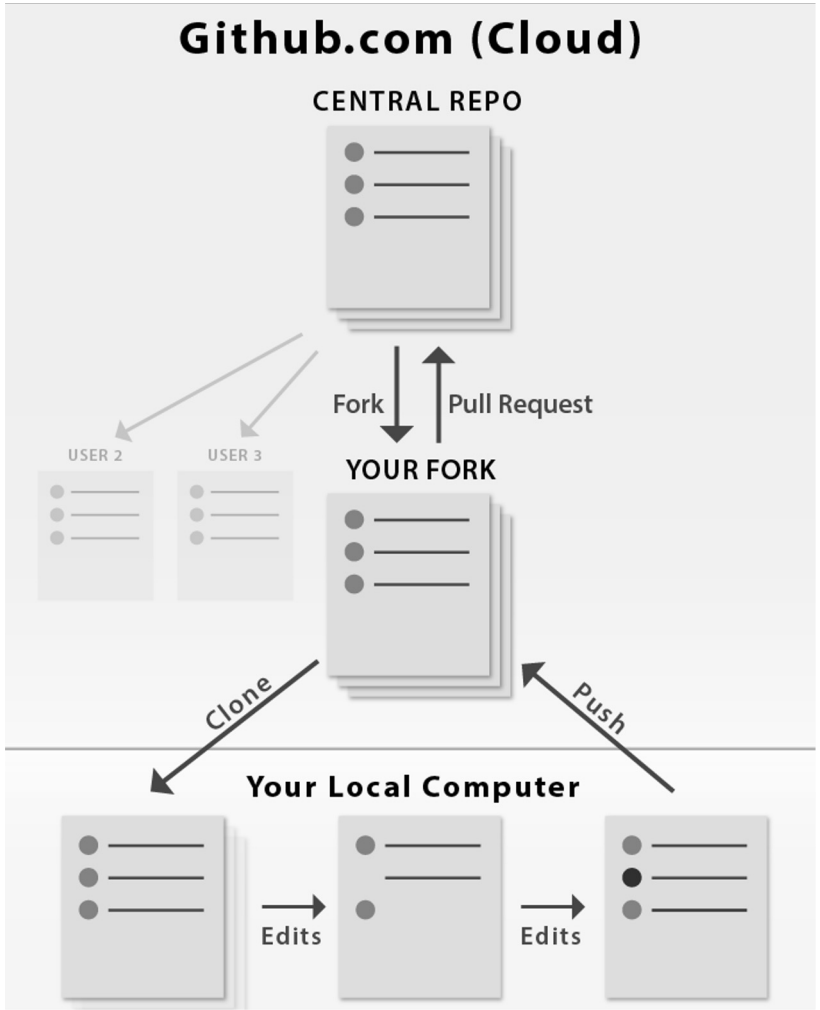


- **Pull:** refers to when you are fetching in changes and merging them. For instance, if someone has edited the remote file you are both working on, you will want to pull in those changes to your local copy so that it's up to date.

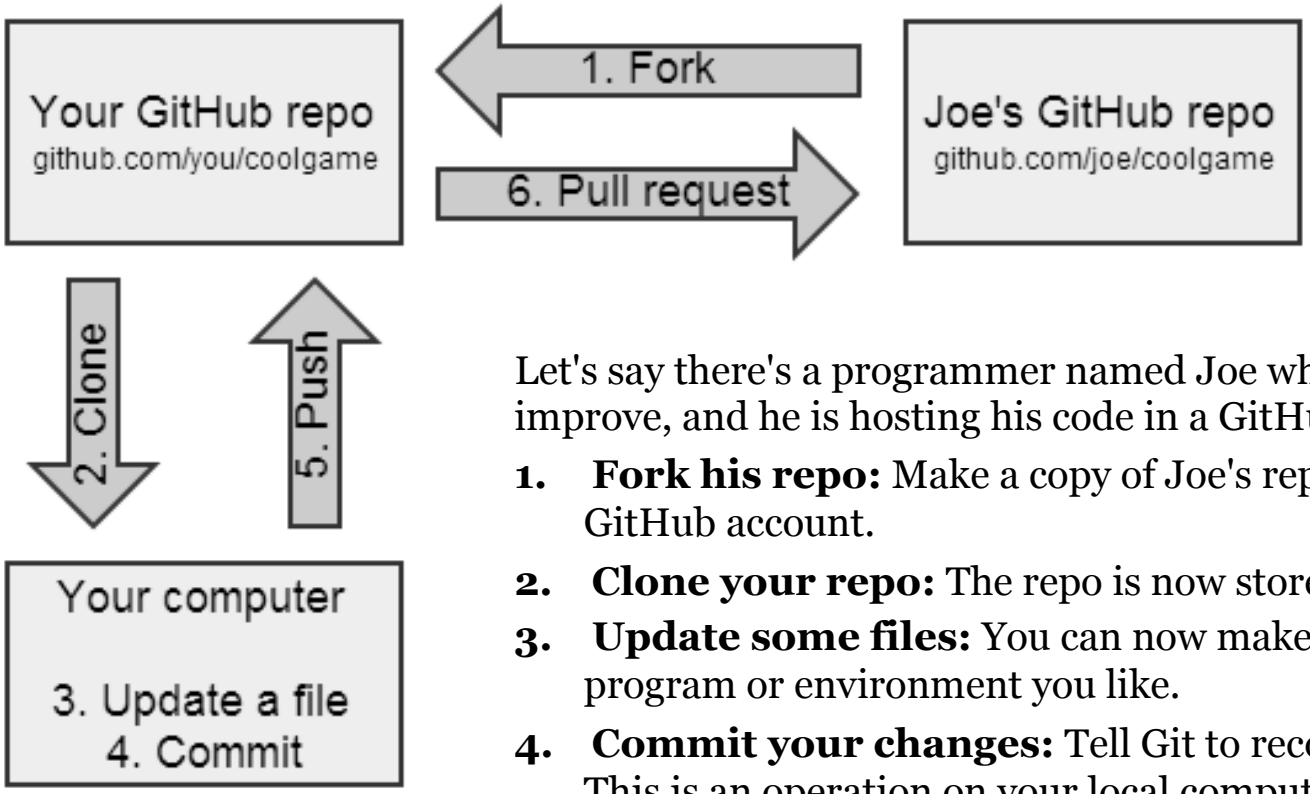
- **Push:** refers to sending your committed changes to a remote repository. For instance, if you change something locally, you would want to then push those changes so that others may access them.

Git/GitHub Terminologies

- **Fork:** is a personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original.
- **Fetch:** refers to getting the latest changes from an online repository without merging them in. Once these changes are fetched you can compare them to your local branches (the code residing on your local machine).
- **Pull request:** requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators. Pull requests each have their own discussion forum in GitHub.



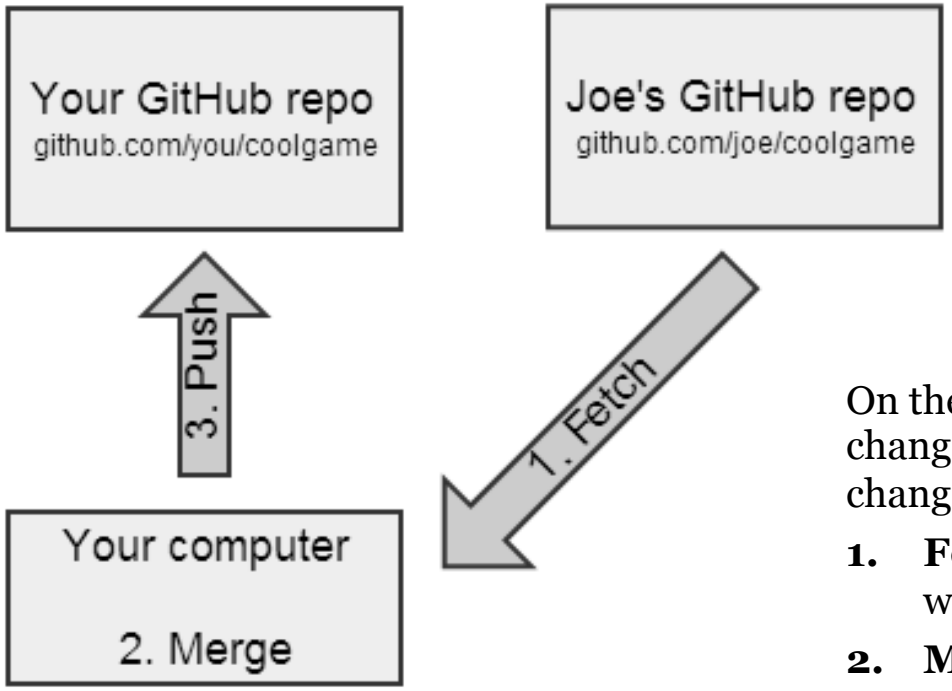
Git/GitHub Terminologies – Example 1



Let's say there's a programmer named Joe who built a game you would like to improve, and he is hosting his code in a GitHub repository. Here is what you do:

1. **Fork his repo:** Make a copy of Joe's repository, which now lives in your GitHub account.
2. **Clone your repo:** The repo is now stored on your local computer.
3. **Update some files:** You can now make updates to the files in whatever program or environment you like.
4. **Commit your changes:** Tell Git to record the file changes you have made. This is an operation on your local computer only.
5. **Push your changes to your GitHub repo:** The GitHub repo now has the changes.
6. **Send a pull request to Joe:** If you think that Joe might like to incorporate your changes, you send him a pull request. It is up to him whether he pulls from you or not. If Joe accepts your pull request, he will pull your changes into his repo.

Git/GitHub Terminologies – Example 2

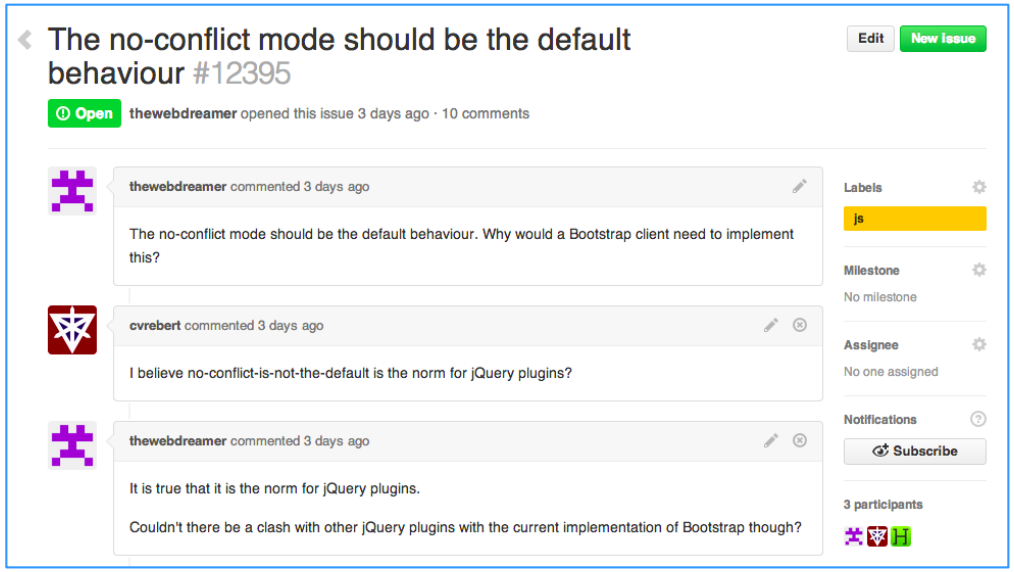


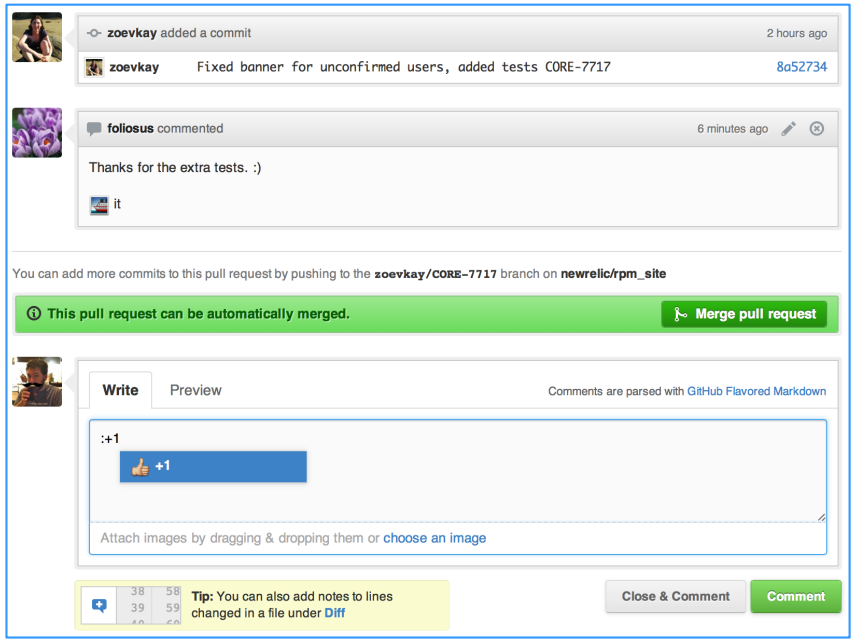
On the other hand, if Joe is the one that has made an extra changes to your software, and you would like to incorporate the changes, then do this:

1. **Fetch changes from Joe's repo:** Tell GitHub that you would like to retrieve the latest files from Joe's repo.
2. **Merge those changes into your repo:** Update the repo on your local computer with those changes (which have been temporarily stored in a "branch"). Note: Steps 1 and 2 are often combined into a single Git operation called a "pull."
3. **Push the updates to your GitHub repo**: Make the final changes available in the GitHub repo.

Git/GitHub Terminologies

- **Issues:** are suggested improvements, tasks or questions related to the repository in GitHub. Issues can be created by anyone (for public repositories), and are moderated by repository collaborators. Each issue contains its own discussion forum, can be labelled and assigned to a user.
- Pull requests also have their own discussion forum in GitHub.



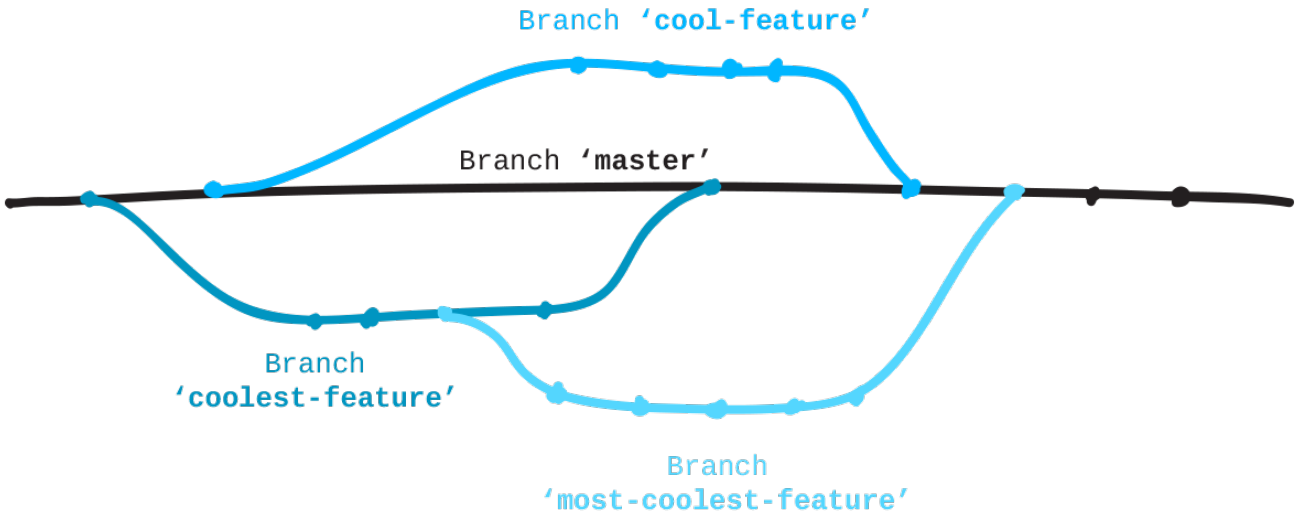


Social Coding

Queen Mary
University of London

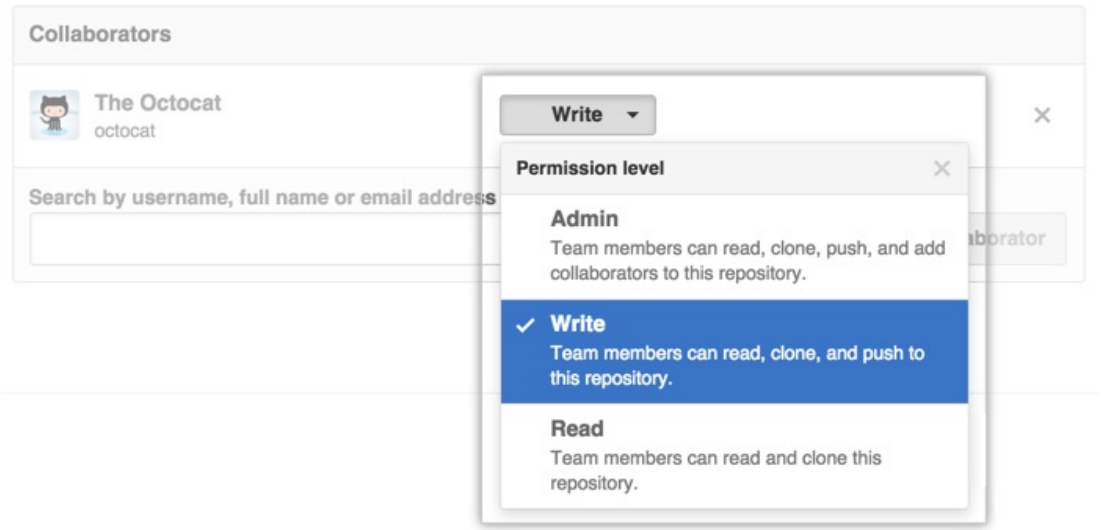
Git/GitHub Terminologies

- **Branch:** is a parallel version of a repository, which allows people to work freely without disrupting the "live" version. When you have made the changes, you can merge your branch back into the master branch to publish your changes.
- **Merge:** takes the changes from one branch and applies them into another. This often happens as a pull request (which can be thought of as a request to merge), or via the command line.



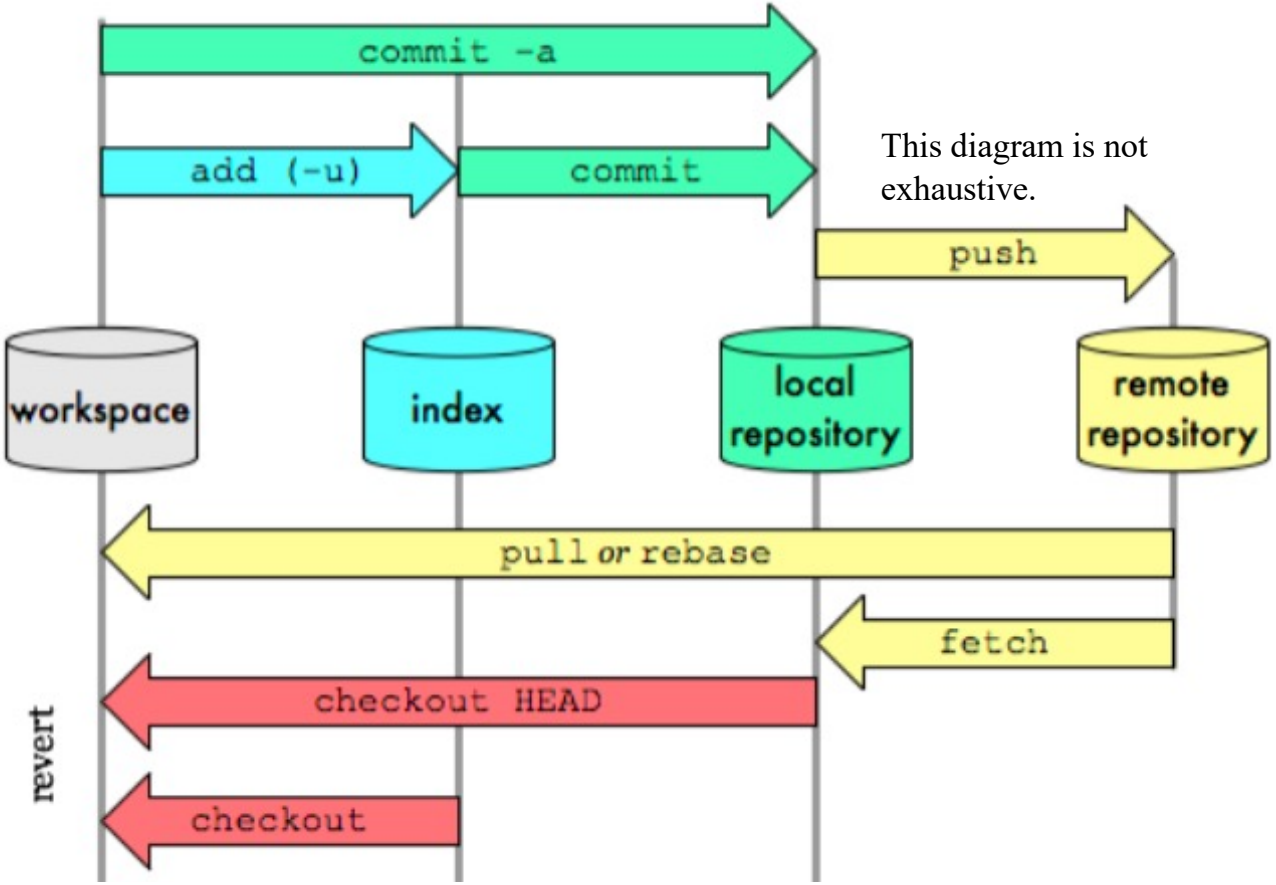
Git/GitHub Terminologies

- **Collaborator:** is a person with read and write access to a repository who has been invited to contribute by the repository owner.
  - Collaborators are in the core development team for the project.
- **Contributor:** is someone who has contributed to a project by having a pull request merged but does not have collaborator access.
  - Contributors are normally not in the core development team but from outside the project who wish to contribute to the project.



Git Commands Diagram

- The web-based GitHub interface may have slight variation of terms.
- If you are working on your own, you probably will not need most of commands.



This diagram is not exhaustive.

The git **index** is where you place files you want committed to the git repository. The index is also known as cache, directory cache, current directory cache, staging area, staged files. Before you "commit" (check-in) files to the git repository, you need to first place the files in the git index.

Understand:

- commit vs. push vs. pull request
- fetch vs. pull vs. fork

Queen Mary University of London

GitHub – Compare Code

- Split view of one commit compared to the one before:

10 Cherry_comics copy old/Cherry_comics/Cherry_comicsViewController.m View

```
@@ -116,10 +116,12 @@ -(void)handleSwipeRight:(UISwipeGestureRecognizer *)recognizer {
 - (IBAction)newImage:(id)sender {
     Cherry_comics_imageView *imageView = [[Cherry_comics_imageView alloc] initWithFrame:CGRectMake(100, 100, 300, 300)];
     [[pages objectAtIndex:(pageNumber)] addSubview:imageView];
+    [[pages objectAtIndex:(pageNumber)] bringSubviewToFront:imageView];

     imageView.layer.borderColor = [UIColor blackColor].CGColor;
     imageView.layer.borderWidth = 1;
-    imageView.backgroundColor = [UIColor whiteColor];
+    imageView.backgroundColor = [UIColor clearColor];
+    imageView.userInteractionEnabled = YES;
     //imageView.image = [UIImage imageNamed:@"bin.jpg"];

     /*
```

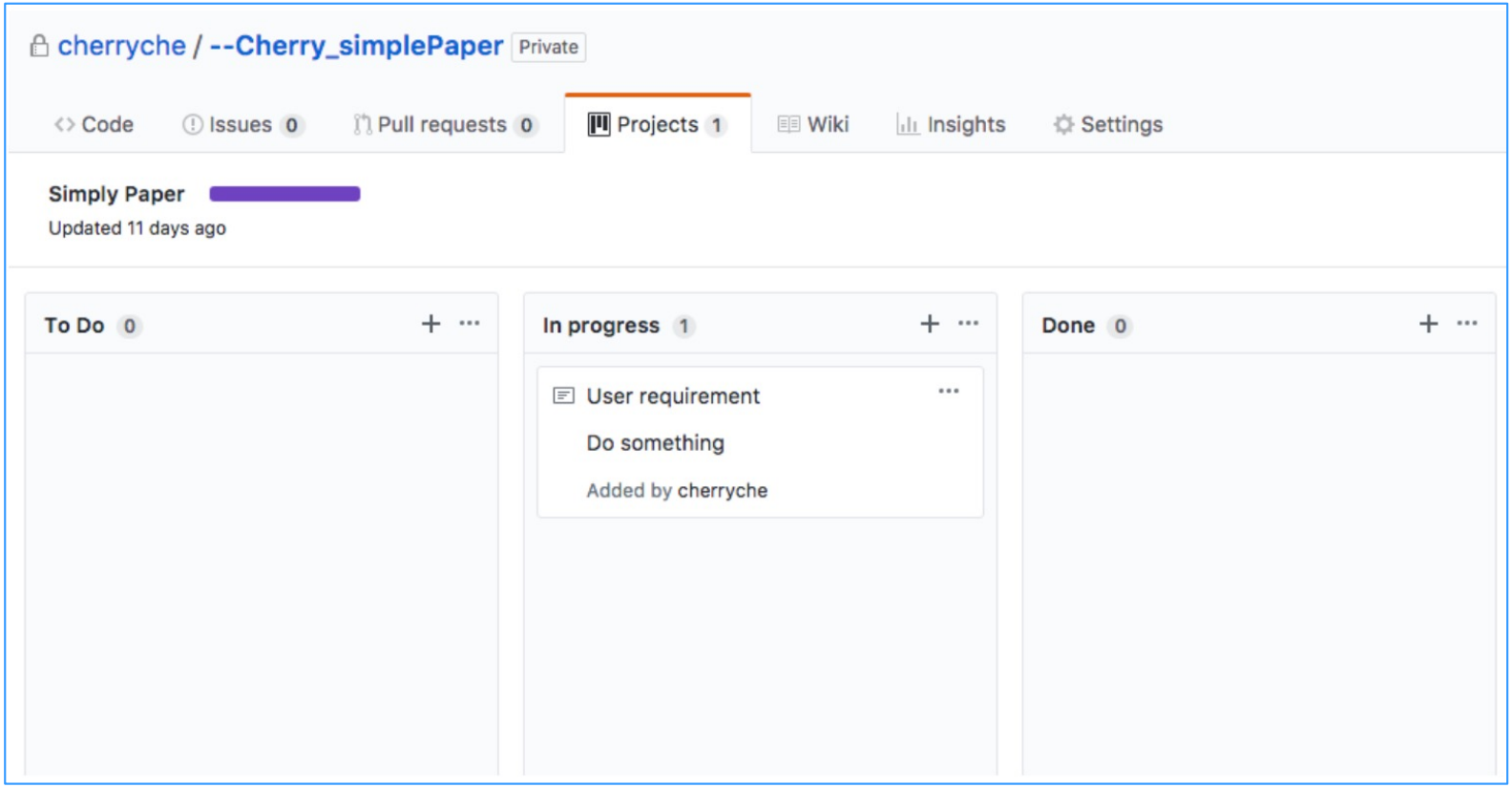
Can be useful for plagiarism checking

GitHub Demo

April 26, 2016

Kanban in GitHub

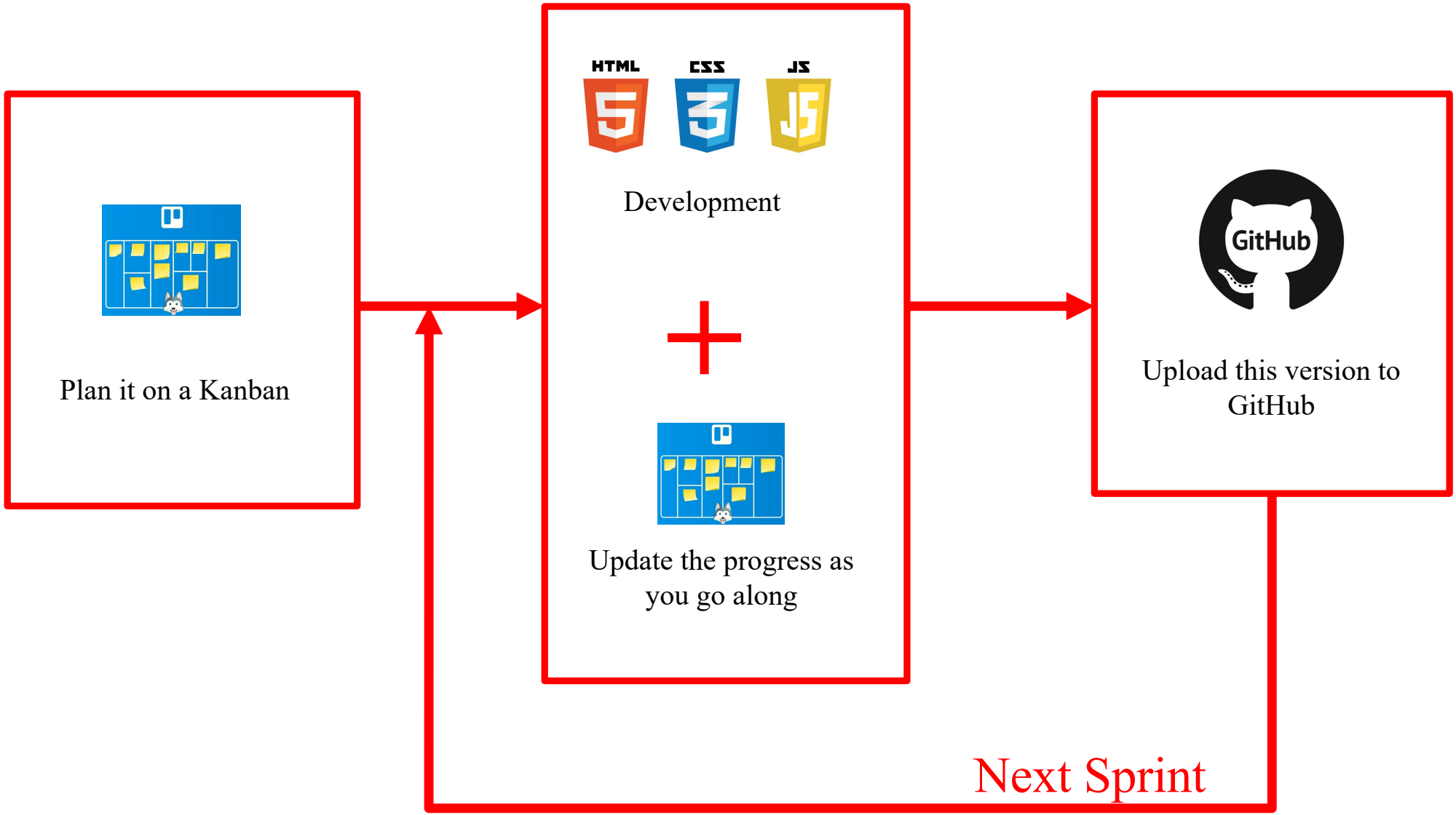
- GitHub also has a built-in Kanban management tool. Within your repository, simply click the “Project” tab and click a new project. Note that the project is not created by default within the repository – you have to manually create one.



GitHub Student Developer Pack

- By default, the repositories you create in GitHub is public, which means everybody in the world can access them. It normally costs a subscription fee to create private repositories. The good news is it is free for students.
- Please visit https://education.github.com/pack, and register for a Student Developer Pack using your **university** email address.
- From now on when creating a new repository, make sure you choose the "**Private**" option instead of "Public".

Manage Your Coursework



Queen Mary
University of London

Questions?

x.che@qmul.ac.uk