# [Lecture 11] Demo - The Planter and the Pruner

An improvisational short skit demonstrating how pull requests can benefit software and software gardener cultivation.

### Roles:

- **Planter**: an eager developer who wishes to add many changes to a repository.
- Pruner: a principal software gardener who seeks to strengthen the sustainability of the work.

## Props:

- **Y Planter:** garden hat (wide brim)
- **Pruner**: pruning scissors
- Shared: readied code changes for a pull request which will be visible to the audience.

# Outline (paraphrased screenplay - feel free to improvise)

Planter excitedly prepares pull request with large number of diverse changes (showing the audience).



Wow, I can't wait to add these changes to the project. It's an overhaul of our previous work and I bet my team will be super impressed by how awesome it is!

Check out this super huge notebook...

And this cool new dependency I've added...

Oh and I also "fixed" quite a number of files other people created...

Okay, I'm ready to submit this for review. I'll bet Pruner will see the genius of this work and approve it right away.

Planter submits the PR and marks Pruner for the review.

### **YPRUNER**

Okay it looks like Planter just submitted a pull request review for me! I'm excited to get this merged, we've been trying to get this added for months.

... (looks pull request over)

Oh wow, this is a huge pull request (highlights and discusses changed lines).

And the types of changes are all over the place (highlights different files and foci of the changes within the pull request).

(frustrated) I can't believe they did this! There's no way I can review all of this in time before my next meeting. This means I can't possibly review this until sometime tomorrow. I'm going to reject this; they clearly didn't even try here.

Pruner requests changes on the pull request with a message like: "Please simplify and resubmit for review."

Planter opens up the pull request review with a grim face...

### **PLANTER**

What?! I can't believe it. I don't know what's wrong with this pull request. Why would I simplify this - what's simpler than having everything in one place? This is so aggravating.

I'm going to go talk with Pruner to figure out where I went wrong. Talking over the phone or in person is almost always the best way to figure out what's really happening if I'm lost.

Planter calls Pruner over the phone (using "hand phone") and Pruner picks up.

### **PLANTER**

Hey - I just wanted to chat with you about the pull request from the other day. I feel confused about why you rejected the work - I don't know how to fix it.

### **YPRUNER**

This was a really big pull request! It shows a ton of effort on your part but it has some challenges that you should be aware of.

Pull requests are ideally small in the number of changes involved, both by lines changed and conceptually. We should seek to create small, modular changes which focus on (ideally) one aspect.

Your pull request is both large (the number of lines changed is high) and also conceptually unfocused. This is hard for reviewers to look through. It's also more likely to introduce bugs because there's a lot happening and it will drastically change any shared context about the software.

I also didn't do a good job of telling you this within the review! I should have done a better job at telling you this, or mentioning that we needed to talk before submitting the review. After all, pull requests are a big opportunity to learn something new (for both the developer and reviewer).

#### **YPLANTER**

Oh I completely get it now! I'm still a little upset that we can't just merge this work without changes, but I feel confident we can pare things down a bit and keep it focused on the goal at hand.

I'll get to work on changes here and follow up for another review to make sure this is on the right track.

Planter goes through the work, talking through the focus of the code changes, and reduces the number of changes to a more minimal set. After the changes are pushed, they open the pull request and re-request a review from Pruner.

Pruner opens up the renewed pull request for another review.

#### **PRUNER**

Wow, this is a huge improvement! I now can leave very specific comments, including code suggestions which can be included in the final work.

... Talks about how the code is now much smaller and more focused...

Pruner includes at least one code suggestion within the GitHub pull request to show how easy it is to implement a fix along with a review.

Planter receives the review with delight.

#### **PLANTER**

Ah great, I received the review and there are more specific suggestions I can follow. There's even a code suggestion I can implement right away (and which will include proper commit attribution to Pruner).

Planter finishes up the review and merges the change.

Planter / Pruner discuss what happened, summarizing how important it is to keep PR's small and focused, to communicate properly in the course of a PR, and how the PR becomes a joint work between the author and reviewer (we're not working alone!).