

# ICE 1: We seriously must have a Silliness Detector.

---

## ICE 2: The CI/CR "Fire Drill" (Process Validation)

- **Objective:** To (1) validate the team's entire CI/CR process with an in-class, TA-supported "fire drill" and (2) assign an individual "competency drill" for homework.
  - **Time Limit:** 45 minutes
  - **Context:** Our "Silliness Detector" (CI) needs to be tested. This is a "fire drill": a single team member (the **Dev Crew**) will intentionally break the build, and the *entire team* will practice the "Code Review" (CR) process to reject and approve the fix.
  - **Team-Building Goal (Not a Race!):** The goal is **not speed**, but **clarity and accuracy**. This is a *validation sprint* to prove your team's process works. The team that wins is the one that communicates most clearly at each handoff. The goal is to **de-risk the individual homework** for everyone.
- 

## Role Kit Selection (Strategy 1: Parallel Processing ⚡)

For this ICE, we will use the **DevOps Kit**. Assign these three roles immediately. *Remember the course policy: You cannot hold the same role for more than two consecutive weeks.*

- **Repo Admin:** (GitHub & Code Review) Creates the feature branch, scaffolds the directories, and acts as the **primary gatekeeper** for the Code Review.
  - **Dev Crew:** (Code, YAML & Failure) **Drives the in-class demo:** Writes the **main.yml**, writes the **failing test**, and finally writes the **fix**.
  - **Process Lead:** (Verification & Documentation) Monitors the "Actions" tab, opens the PR, assigns *all* reviewers, and logs the complete saga in **CONTRIBUTIONS.md**.
- 

## Task Description: In-Class Team "Process Validation"

**Critical Prerequisite:** This workshop is **100% dependent** on the "**Week 9 ICE Prep**" assignment. The **Dev Crew** member (and ideally everyone) must have their self-hosted runner active *right now*.

- **To start your runner:**
  1. **cd** into your **actions-runner** directory.
  2. Run the script: **./run.sh** (macOS/Linux) or **run.cmd** (Windows).
  3. You should see it output: **✓ Connected to GitHub and ...Listening for Jobs**.

### Part 1: Branch & Directory Scaffolding

- **([Repo Admin]) NOTE:** This depends on the PR from the ICEX05 has been merged to main.
  1. **git checkout main && git pull**.
  2. Create the feature branch: **git checkout -b ice2-ci-workshop**
  3. Create the required directory structure: **mkdir -p .github/workflows**

4. Create an empty file: `touch .github/workflows/main.yml`

5. Commit and push this structure:

```
git add .
git commit -m "chore: add ci workflow directory"
git push -u origin ice2-ci-workshop
```

6. **Announce to your team:** "Branch is scaffolded! `git pull` to get the new directory."

## Part 2: Establish the Safety Net (Green

- **([Dev Crew]) ONE MEMBER ONLY**

1. `git checkout ice2-ci-workshop` and `git pull`.
2. Open `.github/workflows/main.yml` and paste in the *exact* contents below.

```
name: MoJ CI Pipeline

on:
  push:
  pull_request:
    branches: [ "main" ]

jobs:
  build-and-test:
    runs-on: self-hosted

    steps:
      - name: Check out repository code
        uses: actions/checkout@v4

      - name: Set up Python 3.10
        uses: actions/setup-python@v5
        with:
          python-version: '3.10'

      - name: Install dependencies
        run: |
          pip install -r requirements.txt

      - name: Run tests with pytest
        run: |
          pytest
```

3. Commit and push this new pipeline:

```
git add .
git commit -m "feat: add initial pytest ci pipeline"
git push
```

- ([Process Lead])
  1. Go to the repo's "**Actions**" tab. Wait for the workflow to run.
  2. **Verification:** The team's goal is to see a **green checkmark** . This proves the pipeline is configured correctly.
  3. **Announce to the Dev Crew:** "We have a green check! Process is validated. Proceed with the fire drill!"

## Part 3: The "Fire Drill" (Get to Red )

- ([Dev Crew]) ONE MEMBER ONLY (DIIFERENT FROM PART 2)
  1. Open `tests/test_app.py`. Add this new *intentionally failing test*:

```
def test_intentional_failure():
    """
    This test is *meant* to fail to test our CI/CR loop.
    """
    assert "The Ministry" == "A silly place"
```

2. Commit and push this failing test:

```
git add tests/test_app.py
git commit -m "test: add failing test to demo ci"
git push
```

- ([Process Lead])
  1. Go back to the "**Actions**" tab. Wait for the new run.
  2. **Verification:** The team's goal is to see a **Red X** .
  3. **Announce to the team:** "Fire! Fire! The build is red!"

## Part 4: The Code Review (Rejecting the PR )

- ([Process Lead])
  1. Go to the repo's "**Pull Requests**" tab.
  2. Open a new PR to merge `ice2-ci-workshop` into `main`.
  3. **Title:** `ICE 2: In-Class CI/CR Workshop Demo`
  4. **Reviewers:** Assign the **Repo Admin AND all other team members**.
  5. Create the PR. Notice the **Red X** on the PR.
- ([All Members (except Repo Admin)])
  1. Go to the PR that the **Process Lead** just created.
  2. Go to the "Files Changed" tab. Click "**Review changes**".
  3. Add a diagnostic comment (e.g., "I see the `test_intentional_failure` is causing the Red X.") and "**Comment**".
- ([Repo Admin])

1. **Wait** for all other team members to comment.
2. After they have, open the PR. Go to "**Review changes**".
3. Add the formal gatekeeper comment (e.g., "This PR is failing CI, as the team has noted. Please fix the broken test (`test_intentional_failure`) before I will approve it.")
4. Select the "**Request Changes**" 🤖 option and submit your review.
5. **Announce to the Dev Crew:** "PR has been formally 'Rejected'. Fix the build."

## Part 5: The Fix (Back to Green ✅)

- ([**Dev Crew**])

1. Open `tests/test_app.py` and **delete** the entire `test_intentional_failure` function.
2. Commit and push the fix:

```
git add tests/test_app.py  
git commit -m "fix: remove failing test to pass ci"  
git push
```

- ([**Process Lead**])

1. Stay on the open Pull Request page.
2. Watch as the **Dev Crew**'s push *automatically* triggers a new CI run, which updates to a **Green Check** ✅ .
3. **Announce to the Repo Admin:** "The build is green! Re-review, please."

## Part 6: The Approval (Approving the PR 👍)

- ([**Repo Admin**])

1. Go back to the PR. You will now see a **Green Check** ✅ .
2. Click "**Review changes**" again. (You may need to "Dismiss" your old review first).
3. Select the "**Approve**" 👍 option and submit your review. **DO NOT MERGE**.
4. **Announce to the Process Lead:** "In-class demo is complete and approved!"

## Part 7: Process Logging & Final Sync

- ([**Process Lead**])

1. Pull all the changes: `git checkout ice2-ci-workshop && git pull`
2. Open `CONTRIBUTIONS.md` and add the log entry (see template below).
3. Commit and push the log:

```
git add CONTRIBUTIONS.md  
git commit -m "docs: log ice 2 in-class demo"  
git push
```

- ([**All Team Members**])

1. Run `git pull`.
2. **Goal:** Everyone on the team must have the final `CONTRIBUTIONS.md` file locally.

## CONTRIBUTIONS.md Log Entry

One team member (Process Lead) share their screen. Open **CONTRIBUTIONS.md** on your feature branch (**ice2-ci-workshop**) and add the following entry **using this exact format**:

```
#### ICE 2: The CI/CR "Fire Drill"
* **Date:** 2025-10-22
* **Team Members Present:** `@github-user1`, `@github-user2`, ...
* **Roles:**
  * Repo Admin: `@github-userX`
  * Dev Crew: `@github-userY`
  * Process Lead: `@github-userZ`, ...
* **Summary of In-Class Work:** We successfully validated our team's CI/CR process.
  1. `Dev Crew` pushed `main.yml` and got a **Green Check **.
  2. `Dev Crew` pushed a failing test, getting a **Red X **.
  3. `Process Lead` opened a PR and assigned all members.
  4. The entire team commented, and the `Repo Admin` formally
    **"Requested Changes" **.
  5. `Dev Crew` pushed the fix, getting a new **Green Check **.
  6. `Repo Admin` **"Approved" ** the PR.
* **Evidence & Reflection:** We just proved our *team's process* works. How does this in-class "validation sprint" help de-risk the *individual* homework, where every member has to do this loop themselves?
* **Homework Evidence Links (To be filled out by 11:59 PM):**
  * `@member1`: [Link to member1's individual demo PR]
  * `@member2`: [Link to member2's individual demo PR]
  * `@member3`: [Link to member3's individual demo PR]
  * *... (add all team members)*
```

After logging, commit and push this file. All other members must **git pull** to get the change.

## Definition of Done (DoD)

### In-Class DoD (Goal: 45 Minutes)

Your team's in-class workshop is "Done" when you can show a TA all of the following:

- **The "In-Class Demo PR"** that is **"Approved"**  and has a **"Green Check"** .
- The demo PR's history *also* shows a **"Red X"**  and the **"Requested Changes"**  event.
- The PR's comment history shows comments from **all** team members.
- **Synchronization: All team members** have successfully run **git pull** and have the latest **CONTRIBUTIONS.md** file locally.

### Final DoD (Due: 11:59 PM Tonight)

This is now an **individual competency drill** for every team member.

#### 1. Individual Task (Everyone):

- Pair up with a teammate.
- **Author:** Create a *new, personal* branch (e.g., `jsmith-ci-demo`) from `main`.
- **Author:** Push a *failing test* and open a PR to `main`.
- **Author:** Request a review from your partner.
- **Reviewer:** Go to your partner's PR and "Request Changes" 🧑.
- **Author:** Push the *fix* to your branch (gets a **Green Check** ✅).
- **Reviewer:** Go back and "Approve" 👍 the PR.
- **Leave the PR open and unmerged!**

## 2. Team Task (Process Lead):

- `git pull` the latest `ice2-ci-workshop` branch.
- Edit `CONTRIBUTIONS.md` and fill in the "Homework Evidence Links" section with the URLs to everyone's individual demo PR.
- `git push` this final log.

## 3. Submission (Process Lead):

- Submit the URL of the *main team PR* (`ice2-ci-workshop`) to Canvas. This one PR will contain all the evidence (in-class and homework links) for grading.
- 

## TA Grading Rubric (10 Points)

The final `ice2-ci-workshop` PR (due 11:59 PM) will be graded using this rubric.

| Criteria               | Points | Description   |
|------------------------|--------|---|
| In-Class Demo Evidence |        | The PR's history clearly shows the team's <b>Green</b> ✅ $\rightarrow$ <b>Red</b> ❌ $\rightarrow$ <b>Green</b> ✅ loop, the <b>CR-Approve</b> loop, and <b>comments from all members</b> .           |
| Process Log & Links    | 3 pts  | <code>CONTRIBUTIONS.md</code> is complete and—most importantly—contains <b>working links</b> to every team member's individual demo PR.   |
| Individual Homework    | 3 pts  | Each linked individual PR <i>also</i> correctly shows the <b>Green</b> $\rightarrow$ <b>Red</b> $\rightarrow$ <b>Green</b> loop and a <b>CR-Approve</b> loop. (Partial credit if some are missing). |
| Total                  | 10 pts |   |