### Logistics

**WEEK 5**

<table>
<thead>
<tr>
<th>Date</th>
<th>Task</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>04/24</td>
<td>L: Build Systems</td>
<td></td>
</tr>
<tr>
<td>04/25</td>
<td>T:</td>
<td>DUE: DnA!!!</td>
</tr>
<tr>
<td>04/26</td>
<td>L: Testing</td>
<td>Testing &amp; CI/CD (TCC)</td>
</tr>
<tr>
<td>04/27</td>
<td>P:</td>
<td></td>
</tr>
<tr>
<td>04/28</td>
<td>L: CI/CD</td>
<td></td>
</tr>
</tbody>
</table>

**WEEK 6**

<table>
<thead>
<tr>
<th>Date</th>
<th>Task</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>05/01</td>
<td>L: Test Coverage</td>
<td></td>
</tr>
<tr>
<td>05/02</td>
<td>T:</td>
<td>DUE: TCC!!!</td>
</tr>
<tr>
<td>05/03</td>
<td>L: Mutation Testing</td>
<td>Alpha_Release (R1)</td>
</tr>
<tr>
<td>05/04</td>
<td>P:</td>
<td></td>
</tr>
<tr>
<td>05/05</td>
<td>LX: Code Defenders</td>
<td></td>
</tr>
</tbody>
</table>
Continuous Integration & Continuous Delivery
CI/CD: What’s the difference?

**Continuous Integration (CI)**
- Integrates code into a shared repository.
- Builds/tests each change automatically.
- Complements local developer workflows (subset of tests vs. all tests).

**Continuous Deployment (CD)**
- Builds on top of CI.
- Software can be deployed at any time.
- Automatically pushes changes to production.
CI/CD: What’s the difference?

**Continuous Integration (CI)**
- Integrates code into a shared repository.
- Builds/tests each change automatically.
- Complements local developer workflows (subset of tests vs. all tests).

**Continuous Deployment (CD)**
- Builds on top of CI.
- Software can be deployed at any time.
- Automatically pushes changes to production.

403 focuses on establishing good CI practices.
Continuous Integration & Delivery Tools

- Jenkins
- GitHub Actions
- AWS CodePipeline
- GitLab
- Travis CI
- Azure Pipelines
- circleci
CI (w/ GitHub Actions) Basics

Terminology: Workflow (pipeline), Event, Job, Action, Runner

https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions
CI (w/ GitHub Actions) Basics

Terminology: Workflow (pipeline), Event, Job, Action, Runner

BTW: What SW Architecture do you think CI/CD uses?
The pipe-and-filter architecture doesn’t specify the design or implementation details of the individual components (the filters)!
Live example: CI in action
Live example: CI in action

```yaml
name: CI - UnitTesting
on: [push]
jobs:
  test:
    runs-on: ubuntu-latest
    strategy: <2 keys>
    steps:
      - uses: actions/checkout@v3
      - name: Set up Python ${{ matrix.python-version }}
        uses: actions/setup-python@v3
        with: <1 key>
      - name: Set up MongoDB ${{ matrix.mongodb-version }}
        uses: supercharge/mongodb-github-action@1.8.0
        with: <1 key>
      - name: Install dependencies
        run: python3 -m venv .venv
      - name: Pre-fly setup
        run: cp $GITHUB_WORKSPACE/GITHUB_ENV
      - name: Test with hatch
        run: |
          hatch run test:test
```

Reuse other people’s work!

Remember L13? One command!
Live example: CI in action

CI Tests run only on push for now. PL + Push was duplicating runs. #15

- test (3.11, 6.0)
  - succeeded 1 minute ago in 1m 16s
  - Set up job
  - Build supercharge/mongodb-github-action@1.8.0
  - Run actions/checkout@v3
  - Set up Python 3.11
  - Set up MongoDB 6.0
  - Install dependencies
  - Pre-fly setup
  - Test with hatch
  - Post Set up Python 3.11
  - Post Run actions/checkout@v3
  - Complete job
Live example: CI in action

(litw-api) nigini@librarian-xps:~/WORKSPACE/LITW/litw-api$ git push
Example: CD in action

GitHub Pages

Websites for you and your projects.
Hosted directly from your GitHub repository. Just edit, push, and your changes are live.
Example: CD in action

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from

Your site is live at https://nigini.github.io/SWEng/
Last deployed by nigini 2 days ago

Build and deployment

Source

Branch

Your GitHub Pages site is currently being built from the main branch. Learn more.

Learn how to add a Jekyll theme to your site.
Example: CD in action
Example: CD in action
Example: CD in action
Continuous Integration & Continuous Delivery

Questions, please!!!