

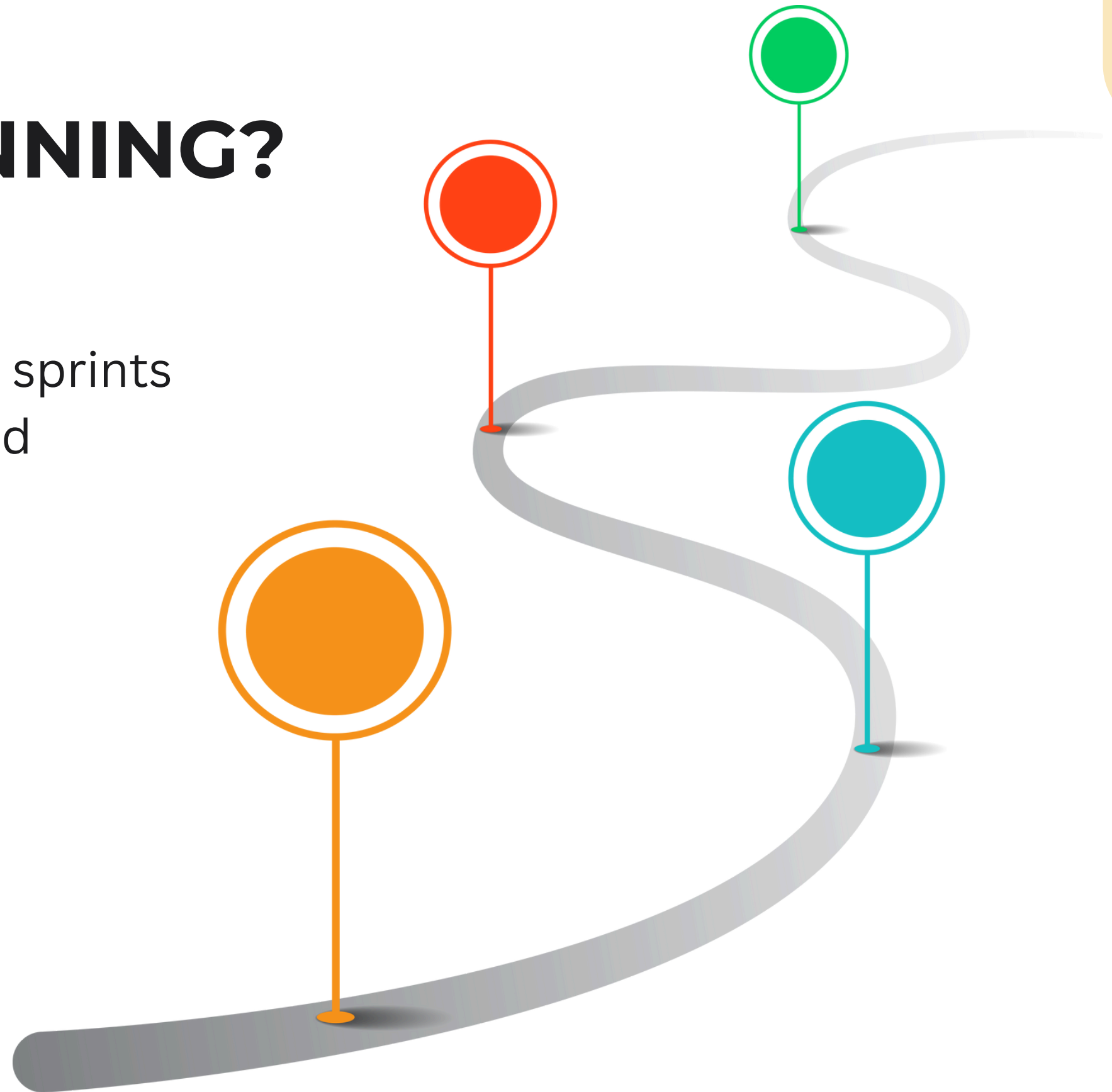
COMP 3550

2.4 — CREATING A RELEASE PLAN

Week 2: Planning Artifacts & Project
Architecture

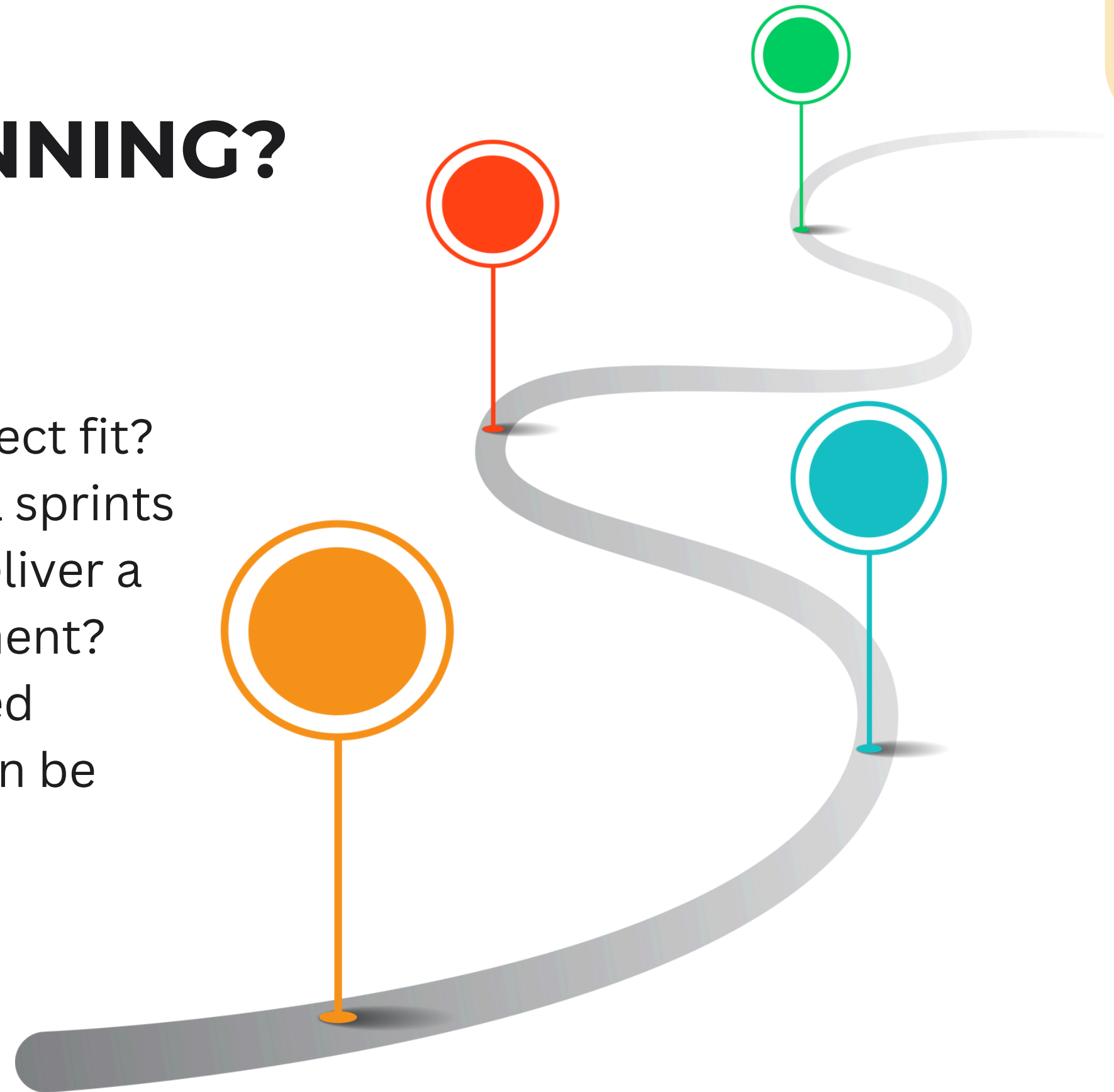
WHY RELEASE PLANNING?

- Roadmap for your team
- Break down a big vision into small sprints
- Adjust scope while staying focused



WHY RELEASE PLANNING?

- Roadmap for your team
 - When will you do what?
 - Does your vision for your project fit?
- Break down a big vision into small sprints
 - What should be shuffled to deliver a working project each deployment?
- Adjust scope while staying focused
 - If you have too much, what can be removed?
 - What needs to shift?
 - What needs to be reframed?



ITERATION BREAKDOWN

- Iteration 1 = skeleton + core feature
 - MVP (Minimum Viable Product)
 - Fakey/Stub Database for Ease
 - Unit Tests
- Iteration 2 = expansion + polish
 - Features Round 2 (with Unit Tests)
 - Payback Tech Debt
 - Real Database (In Memory)
 - Integration Tests for all existing code
- Iteration 3 = Stretch + cleanup
 - Minimum Possible New Features (not Zero new functionality but still some)
 - More Tech Debt inevitable
 - System/E2E/Acceptance Testing

ESTIMATING STORY SIZE

T-shirt Sizes



Story Points



Hours

Working hours?
What about meetings?
Hand to keyboard?
Research?
Debugging?

***Sorry not sorry, we use
hours in this project***

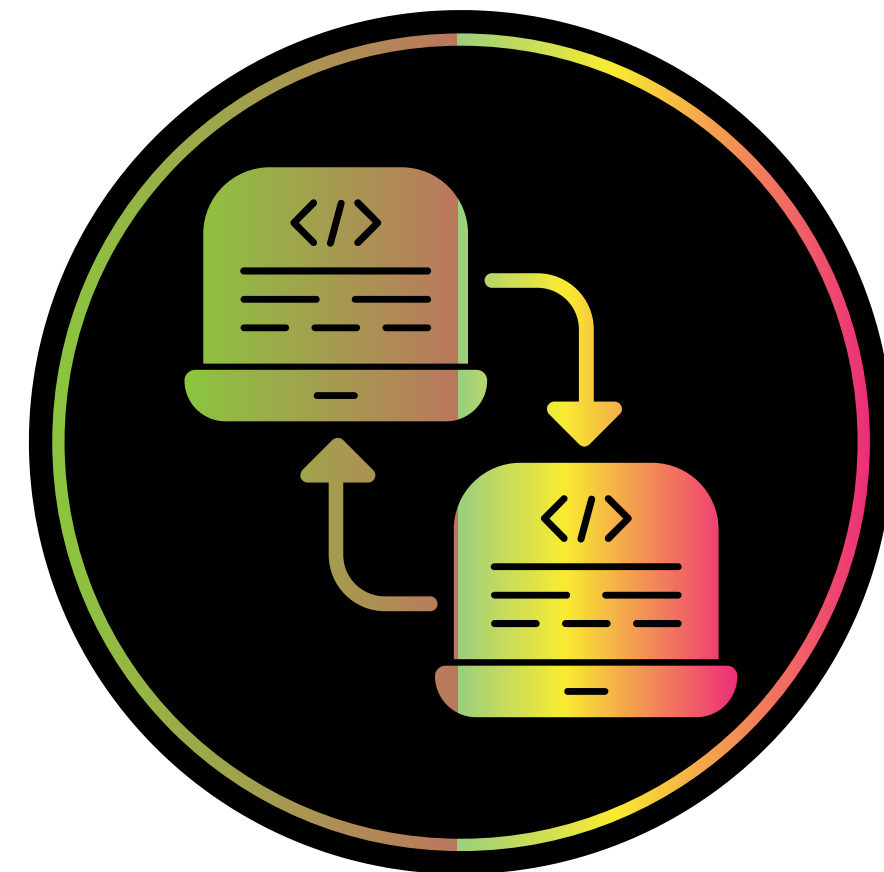
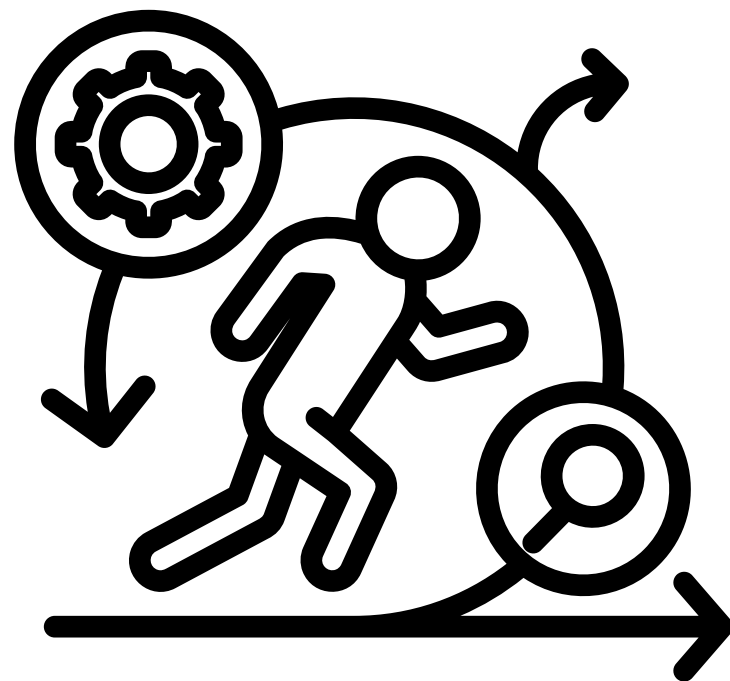
WHAT TO INCLUDE WHEN

- Iteration 1:
 - login, setup, data model, viewing of existing data
- Iteration 2:
 - core workflows: sign up, CRUD, etc
- Iteration 3:
 - error and exception handling
 - polish
 - extras

AGILE REALITY: YOU WILL BE WRONG

The goal is to plan for change, not to predict perfectly

Reserve time for debugging, reworking, refactoring and more...



PAUSE & REFLECT

What would you consider as “done” for your project?

How will you map out your project over the 3 iterations?

Draft a release plan by assigning 3–4 features to each of the 3 iterations. Don't worry — you'll revise it later.