

# Welcome to "Little Bits to Big Ideas"

Lab 9: Software development with AI

TAs: Ashish & Shivani

CAs: Abhinav, Harry, & Sherry



# Two software development philosophies

1. Plan out the entire project
2. Create the software

The **waterfall** model describes how to do this

1. Make something tiny
2. Add one more feature
3. Repeat step 2

Agile methods like **scrum** describe how to do this

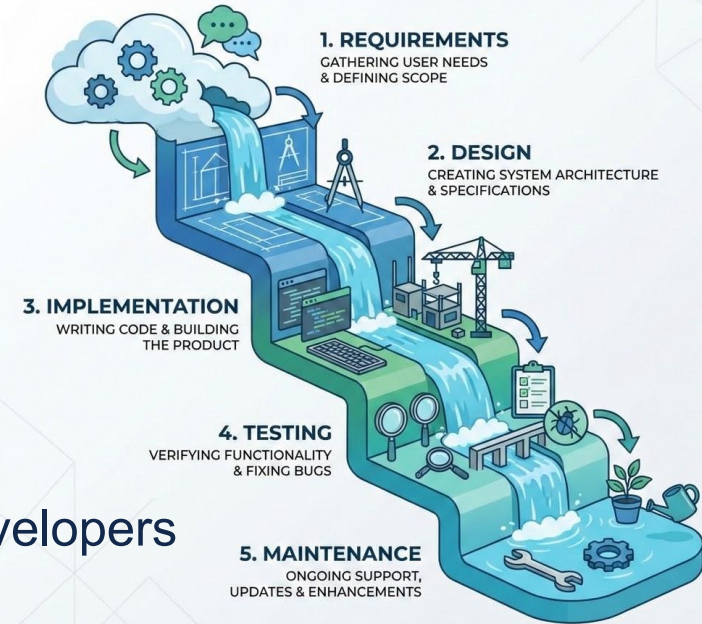
**Spiral** is somewhere in between



# Waterfall

1. Requirements
  - a. Functional = code does
  - b. Nonfunctional = anything else
2. Design
3. Implementation
4. Testing
  - a. Unit, integration, alpha – all done by developers
  - b. Beta – done by clients
5. Maintenance

## The 5-Stage Waterfall Model



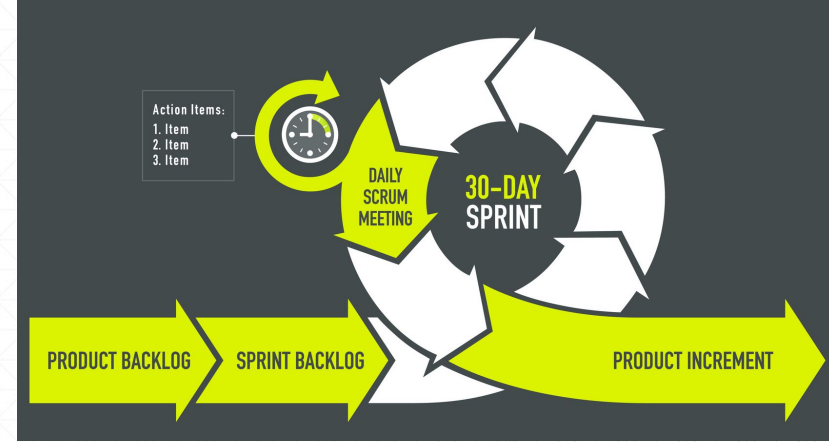
# Maintenance

- Fix bugs
  - ... because testing is not perfect
  - ... and cyber criminals find ways to turn obscure bugs into hacks
- Add features
  - ... because requirements elicitation missed some things
  - ... and some were known but missed the release deadline
- React to upstream changes
  - ... because most code uses **libraries** written by others
  - ... and those get their own maintenance



# Scrum

- Requirements written as individual **stories**
- Stories are placed in a list called a **backlog**
- Development goes in **sprints** of ~4 weeks each:
  - Pick a set of stories to add
  - Design, then refactor, to make room for them to be added
  - Check in daily to see about progress
  - Deliver a working, improved version every sprint



# Build a 2D game

- See [Basic AI-assisted programming](#) text page for tools and tips
- Recommended first prompt:

I want to create an offline web app that I can run from a `file:// URL`

- Start simple, add as you go



Gemini's version of "a gallery of 2D game icons"



# Prompting strategies (for no-login LLM tools)

## Short single-idea prompts

- + Responses become part of future prompt's context
- + Easier to control focus
- May need more prompts
- Might flesh out unmentioned parts in undesired ways

## Complete descriptive prompts

- + Defines the full project, enabling better design
- + Better utilizes LLM input size
- More effort needed to write
- Can overwhelm AI and increase bugs

