Teacher’s Implementation Guide: The Agile Dev Studio

Course: Programming & Software Development | Duration: 4–6 Weeks

I. Instructional Philosophy

This project follows the **Gradual Release of Responsibility (I Do, We Do, You Do)** model. It shifts students from being "tutorial-followers" to "problem-solvers" by simulating a professional software sprint.

II. The 6-Week Sprint Calendar

Week	Phase	Focus	Teacher’s Role
1	**Discovery**	Research & Logic	**Coach:** Help students narrow "World Hunger" ideas into small, buildable MVPs.
2	**Design**	Wireframes & UI	**Art Director:** Ensure UI logic makes sense before a single line of code is written.
3	**Sprint 1**	Scaffolding & Setup	**Lead Dev:** Help with Git repos, environment setup, and boilerplate code.
4	**Sprint 2**	Core Build	**Scrum Master:** Daily stand-ups. Unblock students stuck on specific logic bugs.
5	**QA / Test**	Bug Hunting	**QA Manager:** Facilitate peer-testing. Ensure READMEs are actually readable.
6	**Launch**	Demos & Post-Mortem	**Project Owner:** Final evaluations and professional "Demo Day" presentations.

III. Weekly "Stand-Up" Routine

To simulate a real studio, start every Tuesday/Thursday with a **3-minute team huddle**. Students must answer:

1. What did I finish since the last huddle?
2. What am I working on today?
3. What is my "blocker" (what am I stuck on)?

IV. Implementation of Artifacts

How to use the Project Packet:

- **Distribution:** Hand this out on Day 1. It acts as the "Syllabus" for the unit.
- **The Contract:** Have students sign the AI Use Policy on the back page to establish academic integrity early.

How to use the MVP Checklist:

- **Peer Review (Week 5):** Have Team A use the checklist to audit Team B. This forces students to read other people's code—a vital industry skill.
- **Formative Assessment:** Use this as a "Ticket to Demo." A student cannot present their project until the checklist is 100% "Present."

How to use the Rubric:

- **Summative Grade:** Use this for the final grade. Because you used the Checklist in Week 5, there should be **no surprises** on the final rubric score.

V. Differentiation & Scaffolding Strategies

- **For Advanced Students:** Challenge them to implement an **External API** (e.g., Weather or Maps) or a **Persistent Database** (e.g., Firebase).
- **For Developing Students:** Provide a **Starter Template**. Do not make them start from a blank index.html. Provide the CSS layout so they can focus on the JavaScript logic.
- **For English Language Learners (ELLs):** Utilize the **Technical Writer** role. They can focus on documenting the process using visual diagrams or translated comments before writing complex README prose.

VI. Evidence for your Portfolio (Danielson Aligned)

Keep copies of the following for your own professional review:

1. **The Kanban Boards:** Evidence of student planning (Domain 1e).
2. **The Bug Logs:** Evidence of students responding to feedback (Domain 3d).
3. **Student Post-Mortems:** Evidence of student reflection (Domain 4a).