

Course Context (Summary)

This is a **hybrid** learning situation that:

- focuses on **practical web development skills**
- promotes **self-learning, autonomy, creativity, and digital identity**
- integrates **AI tools, GitHub, VS Code**, and community interaction
- culminates in a **personalized, published portfolio**
- requires **continuous formative assessment**, self-evaluation, and peer review
- targets **adult learners with technical motivation**

Given this structure, the ideal model must support:

- ✓ hands-on creation
- ✓ iterative improvement
- ✓ personalization
- ✓ autonomy
- ✓ collaborative learning
- ✓ digital tool integration

1 ADDIE Model (Analysis-Design-Development-Implementation-Evaluation)

✓ Strengths for this course

- **Structured and systematic**, ideal for complex course planning
- Supports clear alignment of objectives, tasks, assessment, and tools
- Strong emphasis on **iterative improvement**
- Helps document the instructional design process
- Useful for multi-phase projects like building portfolios

✗ Limitations

- Less explicit about learner-centered differentiation
- Less focus on motivation and engagement factors
- Evaluation tends to be project-based rather than ongoing

Fit for this course

ADDIE would work well as a **macro-framework** for designing the course syllabus, resources, progression, and assessments, but does not fully address learner personalization or motivational dimensions.

② ASSURE Model (Analyze learners, State objectives, Select methods/media, Utilize technology, Require participation, Evaluate/Revise)

✓ Strengths for this course

- Tailored to **technology-rich learning environments**
- Emphasizes **active student involvement**
- Strong alignment with multimedia integration
- Encourages iterative revision at classroom level
- Focuses on learner analysis and instructional personalization

✗ Limitations

- Less deep in motivational theory
- Primarily lesson-level rather than full-program instructional design

Fit for this course

ASSURE aligns **extremely well** with a course that requires:

- diverse learner backgrounds
- active experimentation
- use of GitHub, AI tools, VS Code
- continuous hands-on participation

It excels where teaching and technology meet.

3 ARCS Model (Attention, Relevance, Confidence, Satisfaction)

✓ Strengths for this course

- Prioritizes **motivation**, crucial in adult learning
- Helps increase learner autonomy and persistence
- Supports meaningful creation of identity-based portfolios
- Strengthens engagement through feedback cycles

✗ Limitations

- ARCS is NOT a complete instructional design model
- Requires pairing with another structural model (e.g., ASSURE or ADDIE)
- Does not cover content sequencing or assessment design

Fit for this course

ARCS would **enhance** the course by helping learners stay motivated through:

- real-world outcomes (publish portfolio)
- peer discussion and GitHub feedback
- mastery-based progress

But it cannot stand alone.

4 CD/CI – Content Delivery / Constructivist Instruction

✓ Strengths for this course

- Perfect for practical, project-based learning
- Students learn through doing, building, experimenting, debugging
- Encourages metacognition, autonomy, and identity creation
- Aligns with the course's psychological paradigm:
 - constructivism
 - connectivism
 - experiential learning
 - humanistic approach

✗ Limitations

- Less structured
- Requires strong self-regulation
- Depends on facilitator expertise in open guidance

Fit for this course

CI is extremely compatible, because the objective is not to memorize content but to:

- build something personal
- solve problems
- ask questions
- interact with networks and AI tools

Synthesis and Recommendation

Best Overall Model: ASSURE + CD/CI hybrid approach

This instructional situation benefits most from combining:

✓ ASSURE → Technopedagogical backbone

Supports:

- digital tool integration
- accessibility + diversity
- active participation
- learner analysis
- formative evaluation

✓ CD/CI → Cognitive + experiential learning process

Supports:

- project-based knowledge construction
- personalization
- iterative practice
- identity development
- autonomy

Complementary Model: ARCS

Adds motivational depth and supports persistence — highly relevant in programming courses where frustration risk is high.

Useful but secondary model: ADDIE

Valuable for structuring the course at the design level, but not the strongest match for the learners' active creation needs.

Final Conclusion

The best instructional tecnopedagogical fit for this course is a combination of:

✓ ASSURE

as the main instructional design framework, because the course depends on:

- technology use
- active student participation
- multimedia integration
- diverse learner profiles
- continuous reflection and improvement

✓ CD/CI (Constructivist Instruction)

as the core learning paradigm, because learners:

- build real products
- solve problems
- experiment with tools
- construct professional identity

+ ARCS (recommended addition)

to enhance motivation, self-confidence, and engagement.

+ ADDIE (optional structural layer)

to provide design and evaluation coherence at program level.