



Team-Based Software Projects: Preventing Free- Riders

CSC324 Software Design and
Development Course

Fernanda Elliott, Ph.D.

Computer Science, Grinnell College

Team Project: Prevent Free-Riders



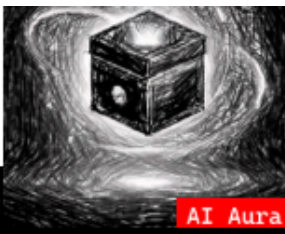
1. Inspiration: the CRediT table from the Software Impacts journal (Which was inspired by: [Publishing: Credit where credit is due](#)).
2. I adapted it to the context of a Software Design and Engineering course:
 - Students must report weekly on their contributions using the [Individual Contributions Log Report](#) (I compare what I observe in class with the entries and weekly demonstrations).
3. Helps with accountability (peers see each other's reports).
4. Unexperienced students find concrete, actionable ways to contribute.
5. I also use a [Professional and Scientific Conduct document](#) to ground teams' practices.

	Week # here	Student_Name_here	Individual Contributions
#	Topic	Description	Student Example What did you do? Your Contributions
1	Project administration Coordination or management	Running retrospective, meetings, organizing meetings or deadlines, checking project scope, updating project scope, helping to group solve conflicts	Planned group meetings
2	Team encouragement and support	Making sure to motivate team members and offer help. Being in touch with the instructor and course mentor and ask for tips/help.	Met and kept in touch with stakeholders
3	Conception, ideas	Providing ideas or checking if the group's activities make sense to the project's purpose	
4	Methodology Development or design of methodology; creation of models	Creating methods or models, e.g., diagrams, visual information, methods for gathering, storing, and retrieving resources	
5	Resources	Provision of study materials, identifying tools, packages and resources	
6	Collection of data or evidence to support the project	Using the group's methods and models to collect data or evidence	Collected data
7	Formal Analysis	Application of statistical, mathematical or other formal techniques to analyze study data	
8	Data visualization	Study and provision of data graphics suitable for the goals and questions	
9	Software Design	Screen mockups, wireframes, concepts	
10	Human in the loop advocate/audit	User interview, accessibility, user performance	
11a	Requirements - client	Checking with the client and making sure the client feedback is <u>taken into account</u> and properly addressed	
11b	Requirements	Creating requirements or checking if the group and project fall under the course's policy and requirements (coding policy, copyrights, proper conduct...)	
12	Software development	programming	Coded the data visualization part of tab 2
13	Software testing	testing and helping to solve bugs	



More details on my Page:
[Instructional Frameworks – Software Design and Development Course](#)

Individual Contributions Log Report



Ignition, Shelf #2, November 25, 2025 (Previous ignitions afterwards):

1. **The secret sauce hypothesis.** What if a tech company poses itself as having developed an artificial general intelligence (AGI) system but actually hides a secret “human sauce”? A human ventriloquist, a conductor, in symbiosis with a machine in a fashion somewhat similar to [this](#) and [those](#) cases? Doesn't that trigger bells similar to [The Mechanical Turk](#) or the Automaton Chess Player? (May be of interest: the [“Maelzel's Chess Player”](#) essay.) **What would it mean if the first “AGI” was part-human (is that AGI)?** What would that mean for the AGI race? Is a *human-in-the-loop* key to achieving AGI, or would you feel more comfortable if it were? If the “winner” of the AGI race offers a human-AI combo... did they really win?

Visit AI Aura:
An experimental space that applies a liberal arts *ethos* for reflecting on GenAI tools' societal impacts