Computer Science: Software Engineering Design vs EGEN 310

	Discover	Define	Ideate	Prototype	Test	Communicate
EGEN 310	• ***Background Research*** • ***Observation*** • ***Interaction with Stakeholders***	***Discoveries distilled into***: • Functions (Verbs)  • Objectives (Adverbs/Adjectives)  • Constraints = standards, non-functional design boundaries	***Generate ideas for design concepts***: • **Functions to Morph Charts** ***Diverging Design Sp***  ***Converging Design Space*** • **Objectives to Pugh Charts**	***Build a prototype to begin to fill in gaps in design knowledge*** • Prototype … • To test most uncertain parts • If fundamental physical modeling is not possible • If statistical information is required • To learn about fit and tolerance of components • Long List of System Interfaces • FMEA • System Integration	***Set up experiments and test to learn more about uncertain design aspects*** • Clearly identify variables in question • Define what constitutes an "effective" result • Summarize next steps	***Utilize MANY forms of communication to inform your team, client and stakeholders about your design process*** • *Face to Face Meetings*: Team meetings (with/without instructor), with other instructors, with Makerspace and Innovation Alley staff • *Written Communication*: Texts, emails, memos, A3, Readme Files, Smartsheet • *Visual Communication*: Fabrication drawings, terrain maps, sketches and schematics, photos, videos, charts to document design process, spreadsheets for numerical models
Computer Science – Software Engineering Design	• ***Very Human Centered!*** • ***User/Client Empathy*** • ***Interviews***	• ***Initial User Story Requirements*** • Functional vs Non-functional requirements • UX Requirements • Accessibility • Level of User Knowledge • ***UML Use case diagram*** • ***Done = ?***	• ***UML*** • ***Interface*** • ***Update User Story as Necessary*** • ***Estimation of Tasks*** • ***Database Design***	• ***Design and Develop Code*** • ***Possibly a UX prototype***	• ***Release and Test with Users to Obtain Feedback*** • ***Continuous Integration*** • Automated testing as each new part of code is added/attached • After each addition, run rests to ensure functionality • **Tools: github, JIRA, Kanban, Travis CI**	• ***From "Day 1" Written Documentation Needs to Include…*** • ***Source Code from Day 1 that Exists*** • ***Tests*** • ***User Documentation*** • ***Developer Documentation*** • ***Attribution of any open source code*** • ***Marketing Writeups*** • ***Licensing***