Capstone Design
CONFERENCE 2024
JUNE 3–5, 2024 ▶ KNOXVILLE, TENNESSEE

Panel 4A: Capstone Without [Physical] Prototypes

Facilitator: Kris Jaeger-Helton (Northeastern)

Panelists: Michelle Stallard (Iowa State), Jon Loftini (Mathworks), Aaron Rubin (Smith) Hadar Ziv (UC Irvine)

Description: Not all capstone classes result in physical prototypes. How are these projects scoped and assessed? How can instructors keep students engaged and focused? These panelists would love to share their knowledge about it!

Clarification: This panel is focused on projects without PHYSICAL prototypes. We are still advocating for iteration and nth generation designs as key aspects of the design process.

Q: What are the non-[physical]-prototype project intangibles that you as panelists are most familiar with? What's cool about them?

- Research projects on occasion - yield results and answers
- Facility floor plan reorganization - helps envision prior to investment
- Civil engineering - no immediate prototypes, but help with visualization
- Prototypes might be financially impossible, so produce digital model instead (office building, for example)
- Most Industrial Engineering projects - often process improvement and/or data analytics
- Value stream mapping and other characterizations of system flow
- Dashboarding with messy data - can facilitate quick inspection of reduced data
- Interdisciplinary projects with SYSTEMS might only have simulations - prior to implementation
- Proposal for which machine to purchase as deliverable - or any type of researched proposal
- Hospital transportation hub recommendations - or any logistics system
- Computer science/software projects - coding deliverables!

Q: Besides the fact that no one will start a fire, get electrocuted, or crushed and assuming everyone emerges with no casualties, what are some [additional] advantages of non-prototype projects? What are some features of successful non-prototype project deliverables?

Advantages:

- Cost is less when not building prototypes
- Get further along into the project without a prototype
- Easy to share and communicate across peers/faculty/sponsors
- Easy to build on across platforms as opposed to physical devices
- Simulations, computer programming

    - Still use the concept of “prototyping”. Prototyping is not necessarily physical, it’s iterations of versions
    - Virtual walkthroughs for example. Provides a ‘preview’ experience.
- Students can manage their timeline better/easier (no part sourcing/purchasing/lead time, etc.); more time designing & analyzing

Cool features:

- Many aspects listed above in first question
- Fewer constraints in terms of cost and time limitations
- Quick iteration by analysis and simulation to prove design works before manufacturing (Formula SAE example)
    - Potential of fun innovation opportunities because they know they have time to iterate
    - Prospects for creativity without needing to source or build (although those are still key considerations)
- Ability to have interesting challenges within the project scope (example: make this building Green - typically CE doesn’t have the budget to actually look into this)

Audience: Have a contrary statement to advantages discussion - everyone is used to Amazon one-day shipping, so they don’t get the experience of learning that lead times are a factor in design (this bolt will take 15 weeks, so redesign the system to find a new solution without the bolt).

Audience rebuttal: in our projects, the teams still have to source parts and determine lead times etc in the project, they just don’t actually purchase. So they still get some of that experience of redesigns if needed.

Q: Does anyone work with human subjects? This is common in process-driven projects and provides learning opportunities as well as challenges that can dramatically affect project lead times…

IRB lead times to consider per university - some take 2 weeks, some take 9 months

Q: How do you establish metrics? How do you evaluate teams/students and assess projects without prototypes?

- Outcome: have a clear summary from the beginning about what key goals are. Work with client to define good/achievable final deliverables. What percent complete did the team get with these deliverables? Clear vision from the beginning.
    - Audience follow up: how do you manifest that students are doing what they say they are, and how do you collect the information ABET needs? When deliverables are analysis/calculations/simulations it can start to feel more like research and less like design.
    - Panel rebuttal: really make sure the students are thinking about the assumptions going into it. Thinking about reality, no garbage in garbage out. Make sure students can **justify** they are considering practical reality. Simulation results must be very close to physical reality. Ensure

students understand the limitations of simulation/analysis and can communicate them clearly.

  - What’s left to be done? That is also a deliverable throughout the weeks.
- Agile has been mentioned several times already in this conference - it has built in metrics.
  - 1 week, 2 week sprints
  - End of sprints deliverable: product demo and retrospectives.
    - Retrospective is about the process as a team
    - Product demo is about the product
- Designing an experiment can count as “design”, then execute the experiment and be able to interpret the results - all help meet ABET requirements. Not all ABET outcomes are related to a physical prototype, which is helpful for non-prototype capstone projects.
- Some project final results are that the goal of the sponsored project was unachievable! This is an acceptable outcome.
- Panelist went through ABET accreditation and was not asked about prototypes
- ABET requirements for Aerospace Engineering are so complicated that prototypes often aren’t practical (orbital science)
- Focus on sponsor requested deliverables, grade on if students followed the **process** instead of the functionality of the final prototype/device
- Consider requiring each team to calculate/present an ROI for customer savings/benefits; use that as a gradable deliverable
- What are programming milestones?
  - Student defined metrics
  - Use Agile

Q: Do your students have standard or unique projects?

- Projects are unique and on a variety of different industries - this makes it important to focus on process
  - Medical
  - Local manufacturing (plant layout, fixtures, shadow boards, process flow, quality improvement)
- Some standard and some unique
  - Requirements come from the outside, and are related to what topic is popular/current in industry
    - AI
    - LLM,
    - Others …
- All unique projects, starting with company sponsor proposals with capstone program bounding/scope management
- Audience - all unique
- Audience - ME unique, AE canned/repetitive
- Audience - all unique (ME, IE, MfgE)

Q from Attendee: How do you manage courses where there are teams with different deliverables in the same course in your Learning Management System?

Do you have to have two assignments for each deliverable (prototype and non-prototype)? This is onerous! Registrar won't let them make two different sections.

Panelist ideas:

- Make more generic assignments (1st is scoping, 2nd is concept generation, 3rd let student teams propose what they want the next design review to cover w instructor approval, 4th, students/teams must determine what to cover). Help students come up with their own metrics then OWN them.
    - Grading moves to something more generic, meets expectations, exceeds expectations, not meets…
    - Grade design reviews on both presentation and ability to answer questions (two different grades)
        - Sections of design review = pre-meeting reading materials, 5 minute presentation, Q&A
- How to measure success
    - Incremental success - first iteration won't work, second interaction works very slow but works (is that success?). Evaluate every phase of the process to determine what the metric of success is at each stage

Q: Examples of why projects (scope) might change:

- Software is perceived as infinitely changeable/modifiable so it is inevitable that changes will be requested
- Uncertainty in requirements can cause scope change (plan for it instead of avoiding it)
- Window manufacturer requested a new plant layout. Students found it wasn't a layout problem but an issue with picker personnel notf following the process. Students proposed different training/picker SOP solutions and still gave the company an efficiency gain without changing the layout
- Poor early scope definition can lead to scope change
- Some projects are just unique and might have no option but to change part way through the AY
- Embrace scope change because it reflects industry real life. Teach students to renegotiate scope with sponsor. You can frame it to be a learning opportunity; help students manage the change (don't let industry sponsor bully them into something unrealistic)
- Set sponsor expectations early that not all goals will be met
- Tolerance for scope change becomes less as semester progresses

Q: Is there anything you do to engage the sponsors and students to maintain interest/enthusiasm without physical prototypes?

- Have students make short videos (2 minutes) for sponsor, so that sponsor can send to boss to prove this is a worthwhile effort in a short period of time
- Use CAD. SketchUp (overlay design over photo) is a powerful communication tool that easily conveys ideas early on

- End of spring video recap
- Encourage early sponsor visits then visit again mid-semester with faculty. End of semester invite all sponsors to campus for final presentations and poster session
- Frequent instructor check ins w students (stand up meetings in the lab, working sessions in class time, etc.).
- Two weeks into spring semester have students do a “show & tell” of something that moves to ensure they are making sufficient progress towards a final product before spring break (turns into slightly competitive exercise)
- Bring mentors in at ⅓, ⅔, and final. Students do presentations at each of those points. When mentors come in person they might take team out to dinner… this can re-energize students.
- 6 minute demo of final system during final presentation cannot be Powerpoint - you have to use something else (videos, click through simulation, etc.) - quite successful for both in-person and online presentations

Q: Final advice for advising, developing, managing non-prototype projects?

- Focus on the process, it always leads the team in the right direction
- Open communication with sponsors, maintain good relationships
- Frequent meetings with sponsors
- Have clear deliverables set at the beginning,with agility for project shift
- Documentation for non-prototype projects becomes more important (assumptions & constraints clearly defined, comments in code, etc.)
- For instructors, self reflection, continuous improvement
- Normalize that not all capstone projects need physical prototypes

Mathworks has a github repository of different ideas for no-prototype projects! Check it out!

https://github.com/mathworks/MATLAB-Simulink-Challenge-Project-Hub