CLASSIFICATION OF GENERIC DESIGN TASKS TO PROMOTE DESIGNER FLEXIBILITY AND INTEGRATION SKILLS IN CAPSTONE PROJECTS

S. Li*, R. Hugo, R. Brennan
Schulich School of Engineering, University of Calgary
*Corresponding Author E-mail Address: simoli@ucalgary.ca

Abstract – Designer flexibility is referred to as an ability to adopt design tools and engineering knowledge to solve design problems. As design methodology is intended to be general for different kinds of design problems, it would not be particularly helpful for designers to connect technical content to specific design applications, and students often face challenges with this connection. To address this issue, we propose five types of generic design tasks, which are applied as a platform for students to integrate their knowledge and skills for design work. These generic design tasks are background research, problem framing, idea generation, decision making and scientific analysis, which can take place in multiple design stages. After mapping design tasks and stages, we can provide common vocabulary for multidisciplinary design, define skill levels for design assessments, suggest a “reverse” learning path to train design skills from well-defined to open-ended problems.

Keywords: Engineering design education: generic design tasks: multidisciplinary design: design pedagogy

1. INTRODUCTION

Design methodology is important in engineering design education. In this paper, it is generally referred to as the content of some popular design textbooks such as Ulrich and Eppinger (2004) and Dym et al. (2014). Though these textbooks are not the same, they share some features to support design education. First, they provide a systematic treatment of the design process by clearly setting up a series of design stages (e.g., concept generation → concept selection → embodiment design). In each design stage, they provide reasonable and easy-to-follow procedures to address typical design tasks (e.g., quality function deployment to define engineering specifications and decision matrix for concept selection). Further, design methodology has defined a set of vocabulary (e.g., requirements, design concepts, performance metrics) for new learners to articulate and make sense of their design work.

At the same time, we would note that design methodology is a double-edged sword. After acquiring the techniques from design methodology, students may take what they have learned as the design method and have yet to acquire flexible skills to practically address design problems. This situation can become more apparent in capstone design projects, where design problems are less regulated with diverse design expectations from different stakeholders (e.g., professors for research-related design, community leaders for environmental improvements, company sponsors for tangible design outputs). While experienced engineers can flexibly adopt design tools for their work, students generally need some instructional support to gradually acquire such designer flexibility. This becomes the primary motivation of this paper.

When taking a course, some students could be sensitive to the grading scheme to recognize the types of work to receive a good grade. This “grading-scheme” mentality may turn a student’s focus to following a given set of design methods (maybe mistakenly perceived as a requirement from instructors). By designer flexibility, students should be relatively free from the concern of grading schemes and turn their focus to the design problem itself. However, such “freedom” could cause challenges because students often struggle with the choice and application of techniques given the context of a problem. Then, designer flexibility calls for an integration skill that allows designers to flexibly adopt different techniques for problem solving.

We classify this paper to the “Practice” stream because it represents a critical reflection of capstone teaching. We initially taught capstone classes with design methodology and gradually noticed some learning gaps (e.g., good execution of design methods did not necessarily imply good design results). As students in capstone classes should have learned design methodology, our main focus was not on advancing their knowledge in this direction. Instead, we turned the instructional focus to some generic design tasks such as decision making and scientific analysis to hone their integration skills in design work. Gradually, a framework of a design process coupled with
generic design tasks is formed, and it will be discussed in this paper.

We treat “generic design tasks” as a tangible platform for individual students to integrate their knowledge and design skills. With the focus of generic design tasks, in Section 2 we map them to different stages of a design process to propose a design task-stage framework. Then, Section 3 will show how this framework can be used to describe design work in different engineering disciplines. Section 4 will discuss how to apply the framework to support design education. Section 5 will provide closing remarks.

2. PROPOSED FRAMEWORK

2.1. Brief review of design methodology in textbooks

Engineering design textbooks have indicated one common framework for content organization. This framework typically starts with the structure of an engineering design process with various definitions of design stages (e.g., conceptual design, embodiment design). Then, various sets of vocabulary, methods and tools are taught to support the design work at each design stage. Table 1 lists some design textbooks with their design stages and relevant methods to demonstrate the content organization. Based on a similar organization, some educators make textbooks more student-friendly to support student design projects (e.g., Dym et al. 2014; McCahan et al. 2015; Hyman 2003; Cross 2000). Notably, we would comment that these textbooks tend to incline towards mechanical design or product design.

For comparison, we have briefly checked some textbooks on the design of software systems (Sommerville 2007), fluid thermal systems (Janna 2011) and pipeline systems (Mohitpour et al. 2007), which are discussed as follows.

- In the software textbook, a basic design process was articulated (i.e., specification \(\rightarrow\) design & implementation \(\rightarrow\) validation \(\rightarrow\) evolution) (p. 64), with a discussion of various process models (e.g., waterfall, spiral). The majority of the content covers domain-specific knowledge and techniques (e.g., requirements engineering, software architectures, object-oriented design and verification & validation).
- In contrast, the fluid-thermal textbook briefly discussed a design process with major contents to cover scientific fundamentals (e.g., fluid mechanics and heat transfer) and system components (e.g., piping, pumps and heat exchangers). Extensive discussions of physical equations can effectively support parametric design (e.g., sizing problem and evaluation of design properties).
- The pipeline textbook starts with a discussion of major considerations of pipeline design (from fluid properties, codes & standards to economics and operation). Then, it talks about pipeline route selection (a kind of system-level design) and then component details (e.g., fluid transmission, compressors / pumps and other mechanical components).

In this brief review, we notice that the basic design workflows described in design methodology textbooks and domain-specific textbooks are similar (e.g., from system-level design to component design, from general ideas to concrete details). While domain-specific textbooks are narrow to their design applications, they tend to provide more technical information that could be directly helpful for students who work on the same type of design. In contrast, to maintain the generality of design methodology, it is not possible to connect technical content to specific design applications, and students often encounter difficulty in closing this gap. This observation motivates the classification of generic design tasks for student design work.

2.2. Three-layer framework

Figure 1 shows the structure of the proposed framework, which has three layers. The top layer represents a design process, which consists of multiple design stages. In this paper, three design stages are set to highlight some key transitions: concept \(\rightarrow\) details \(\rightarrow\) testing, which should be generic to different engineering disciplines. These design stages are not intended to be unique and rigid, and they can be changed and refined for the specific needs of design projects.

The third (bottom) layer represents the inventory of knowledge and design tools / methods associated with engineering disciplines. Students develop their own inventory through multiple sources such as courses, projects and work experience. Notably, this inventory could represent the technical dimension of an engineering design project but proficiency of the knowledge in this inventory does not naturally imply good design work. This explains the need for the second (middle) layer: generic design tasks.

Generic design tasks are referred to as task types that are commonly involved in multiple design stages. For example, while we can see the need of the task “idea generation” in conceptual design (e.g., generate a new mechanism for snow removal), designers still need to get new ideas for design details (e.g., generate new shape of the connecting rod to support the new mechanism). Here, generic design tasks are intended to provide a perspective or platform that articulates the required information for the design project at different stages. With such a platform, students are asked to integrate the knowledge from their inventory to complete the tasks. The next sub-section will elaborate and discuss these generic design tasks.
Table 1: Sample textbooks with design stages and methods.

| Textbooks                  | Design stages          | Design methods                                                                 |
|----------------------------|------------------------|--------------------------------------------------------------------------------|
| Pahl and Beitz (1996)      | Conceptual design      | Abstraction → function structure → working structures                           |
|                            | Embodiment design      | Task decomposition, multiple design principles, design for X                   |
| Ullman (2003)              | Project definition and planning | Identify tasks → state objectives → estimate resources → develop sequence → estimate cost |
|                            | Specification definition | Quality function deployment (QFD)                                              |
|                            | Conceptual design      | Product decomposition, functional modelling, morphological method, decision matrix |
|                            | Product development    | Form generation, material / process selection, performance evaluation, tolerance analysis, robust design, design for X |
| Ulrich and Eppinger (2004) | Concept development    | Identifying customer needs → concept generation → concept selection → concept testing |
|                            | System-level design    | Quality function deployment (QFD), product architecture                        |
|                            | Detail design, testing & refinement, production ramp-up | Design for manufacturing, robust design, economic analysis                     |
| Dieter and Schmidt (2013)  | Conceptual design      | Define problem → gather information → concept generation → evaluation of concepts |
|                            | Embodiment design      | Product architecture → configuration design → parametric design                |

2.3. Discussion of generic design tasks

The first type of design task is background research (BR), which requires students to identify the relevant unknowns, check and learn new materials from valid sources. Besides the important library and Internet search skills, students also need to recognize the needs of the design project. For example, learning less-relevant information can make the project’s execution ineffective. Instructors or project advisors can help by providing the sources of information, guiding the learning of new materials and judging the relevance of information. Apparently, students can acquire new information continually through background research in different design stages.

The second type of design task is problem framing (PF), which asks students to define the problem’s scope and plan the workflow. While students are used to solving well-defined problems in their courses, framing sub-problems for design projects is not easy for them. For example, students can calculate the heat flux of a heat exchanger given the fluid’s properties and relevant coefficients. Yet, to design a space heating system, students may need to estimate the heat flux of a radiator by checking and assuming relevant parameters on their own. Taking a step back, they may even need to decide the level of accuracy of the results for their work. For example, data from manufacturers could be sufficient for conceptual design, while detailed models are needed for numerical simulation.

The third type of design task is idea generation (IG), which asks students to develop possible solutions for problem solving and creation of new design contents (e.g., concepts and design details). This task type represents the creative aspect of a design project. Besides coming up with new design concepts, this task can also call for new design details in design development and new testing scenarios in design verification.

The fourth type of design task is decision making (DM), which requires students to recognize the decision scenarios and formulate a decision problem. For example, some decision scenarios can be obvious and informal (e.g., which software tool to use in numerical analysis), while some can involve serious deliberations with stakeholders.
(e.g., choosing a design idea implies noticeable prototyping cost). When formulating a decision problem, typical work includes shortlisting comparable options, setting decision criteria and defining the numerical schemes for comparison. A decision matrix from design methodology can be used to organize and analyze the information. Notably, beyond conceptual design, decision making can be required in other design stages such as component selection for design development.

The fifth type of design task is scientific analysis (SA), which engages engineering and scientific knowledge for the design work. While this task type is obviously found in design development along with traditional technical content, it can also involve rough estimations for feasibility assessments in conceptual design and statistical analysis for design verification.

The “task” notion in design work is hardly new in terms of “what to do” in a design process. In design methodology, design tasks can be articulated as, for example, elicitation of requirements, problem decomposition, concept generation and selection. One difference of the proposed framework is that its design tasks implicate some kind of “general” design skills, while design tasks in design methodology are strongly tied to individual design stages. With this difference, we consider the proposed framework as a scaffold that helps students generalize their design skills. For example, students can first learn and practice the techniques from design methodology. Then, these design techniques can be part of their inventory (see Figure 1). When they work on a more open-ended project, generic design tasks help them to recognize the need to choose and adapt the learned techniques to a new situation. For example, they can adapt the concept generation techniques to explore and improvise prototyping options in design verification.

In summary, the essence of the layer of generic design tasks is to provide a tangible platform for students to integrate the knowledge and skills from their inventory to complete the design work. This kind of scaffold is aligned with a criterion of Canadian Engineering Accreditation Board (CEAB) that capstone design should be “based on the knowledge and skills acquired in earlier work” (CEAB 2018). Further, we can potentially map these generic design tasks to some CEAB’s graduate attributes as suggested in Table 2. On the other side, this kind of scaffold is not obvious to students because they could have various kinds of expectations with “design” (e.g., designing is just about making new things). While design methodology has been providing useful vocabulary and techniques for design work, the proposed “task layer” can allow students to adapt relevant knowledge to new situations more flexibly. In this aspect, instructors are free to change and add the generic design tasks to fit their pedagogical purposes.

### Table 2: Possible mapping of generic design tasks to CEAB’s graduate attributes.

| Generic design tasks | CEAB’s graduate attributes |
|----------------------|-----------------------------|
| Background research (BR) | Investigation |
| Problem framing (PF) | Life-long learning |
| Idea generation (IG) | Problem analysis |
| Decision making (DM) | Economics and project management |
| Scientific analysis (SA) | Use of engineering tools |
|                      | Impact of engineering on society and the environment |
|                      | A knowledge base for engineering problems |

### 2.4. Illustration

Let us consider the design of an autonomous snow removal device as an example. The proposed framework can be condensed to a simple matrix as shown in Table 3. The matrix captures the three design stages (vertical columns) and five types of design tasks (horizontal rows). Each matrix entry can note the design work associated with a task type in a design stage.

For example, check the column of conceptual design in Table 3. As autonomous snow removal is a relatively new concept, the design team can check any similar designs such as a robot vacuum for autonomous features and a snow blower for mechanical mechanisms. In problem framing, the design team considers what they can deliver in two semesters. After realizing their own skills and resources, they decide to only focus on the aspect of snow removal mechanisms (while autonomous features can be deferred to a future design project). In idea generation, they develop three possible ideas to remove snow from the ground, and they further analyze these ideas in the decision-making task. In scientific analysis, they estimate the required forces to remove a specific piece of snow using back-of-the-envelope calculations.

This discussion can be continued for the stages of design development and design verification, where Table 3 suggests some sample design tasks. While different design teams can fill in different contents in this matrix, we think that this framework can outline some reasonable work for the design of snow removal devices. Notably, each task type can be applied in different design stages. If one student learns some practical skills in background research (for example, use of libraries and filtering of information), these skills can be carried forward for multiple design stages and different design projects. This kind of flexibility and transferability of design skills can be demonstrated through this framework by integrating design stages and task types.
Table 3: Autonomous snow removal design in a design task-stage framework.

|   | Conceptual design | Design development | Design verification |
|---|------------------|--------------------|---------------------|
| #1: BR | Study a similar design: iRobot, snow blower | Safety code of snow blowers | Check standard performance of snow removal devices |
| #2: PF | Recognize the complexity and do the snow removal mechanism only | Define smaller problems: force balance, FEA, prototyping planning | Define functional testing and performance testing |
| #3: IG | Check three possible ways to remove snow from the ground | Design geometry, short-listing materials and prototyping techniques | Come up with testing ideas |
| #4: DM | Select one idea for further development | Component selection | Select the test plans and decide on the testing parameters |
| #5: SA | Estimate the required forces for snow removal with different ideas | Free-body diagram, FEA | Statistical analysis of testing results |

3. APPLICATION TO MULTIPLE ENGINEERING DISCIPLINES

Traditionally, the contents of design textbooks tend to target mechanical design (Pahl and Beitz 1996; Ullman 2003) and product development (Ulrich and Eppinger 2004; Otto and Wood 2000). However, the spectrum of engineering practice can be wider such as the design of chemical processes or software tools. To this context, the method-based teaching approach from design textbooks may not be applicable without major adjustments. This section tries to address this gap using the proposed framework.

As Table 3 has shown an example of mechanical design, this matrix format could be reused for other engineering disciplines. Table 4 tries to articulate some less discipline-specific design work in the mapping of generic tasks to three design stages. For example, “idea generation” can be applicable in the stage of “design development” by creating design details. This aspect could be applicable to multiple disciplines such as electrical designing a mount for motors, chemical pipe sizing, and civil construction details. Due to the space limit, we have demonstrated this matrix format for chemical and civil engineering only in Appendix. While we may not agree with the details of design work, we believe that the proposed framework can outline the design work from different engineering disciplines reasonably. From here, we have some common vocabulary in terms of design stages and tasks to describe design work from different disciplines.

Table 4: General design task-stage framework.

|   | Conceptual design | Design development | Design verification |
|---|------------------|--------------------|---------------------|
| #1: BR | Check similar designs. | Check codes and standards. | Check testing procedures and safety. |
| #2: PF | Define project scope. Outline project deliverables. | Define smaller tasks for individual work. | Classify test types and purposes. |
| #3: IG | Generate design ideas. Identify prototyping techniques / resources. | Create design details. Find off-the-shelf components. | Design test cases and scenarios. Design simulation approaches. |
| #4: DM | Set up the selection criteria. Select the design concept / prototyping techniques. | Determine parametric details. Decide on major design revisions. | Shortlist control parameters. Decide on fixes for test failures. |
| #5: SA | Estimate technical feasibility. Perform back-of-the-envelope calculations. | Analyze engineering properties of the design. Perform cost and benefit analysis. | Program simulation process for verification. Analyze test results with statistics. |

In view of the education of multidisciplinary design, common vocabulary is important. First, instructors of
different backgrounds can seek for their common expectations of project progress in terms of design stages. They can seek for some unified assessment through the quality of design tasks that they all recognize. Second, students from different disciplines can coordinate the boundary of their design works through the classification of design tasks. Also, they can collaborate with each other through their common understanding of design stages and tasks.

4. PEDAGOGICAL SUGGESTIONS

In engineering design education, one open question is how to inform the progression of student mastery of design skills. With the proposed framework, the development of design skills can be situated in the work of design tasks to the context of design stages. The next two sub-sections will discuss some suggestions.

4.1. Defining skill levels for design tasks

In Bloom’s Taxonomy (BT), six categories have been classified to inform the levels of student learning. In this work, we reference a revised version of BT (Krathwohl 2002) and then re-group these categories to classify three levels of design skills as follows:

- Junior level: Related to “Remember & Understand” in the revised BT, students do design tasks based on their intuitions and their prior experiences.
- Medium level: Related to “Apply & Analyze” in the revised BT, students start to apply new techniques for design tasks and analyze design work using scientific principles.
- Advanced level: Related to “Evaluate & Create” in the revised BT, students can evaluate their design work for the practical context of design problems. They can create new techniques and tools for design tasks.

Generic design tasks can provide an appropriate view to articulate the development of design skills. Table 5 shows an example of the mapping of design tasks to three skill levels. For example, consider the design task of idea generation, which can be strongly influenced by personal experiences. At the junior level, we acknowledge the efforts by students who can bring ideas that are close to their immediate experiences. Then, we encourage them to criticize their own ideas by checking how professionals work out similar problems (e.g., use of codes and standards to direct design ideas). While having one idea is good, the advanced level can check how effective students can explore different possibilities (rather than having an early buy-in).

While the detailed contents of Table 5 can change with the context of design work and the disciplinary nature, it suggests a general template to assess design skills, and it can be further developed for the rubrics of grading components.

| Table 5: Generic design tasks with three skill levels. |
|------------------------------------------------------|
| #1: BR | Identify and apply the materials learned before | Know how to search external information | Filter the relevant information for the design problem |
| #2: PF | Frame design problem as school exercises that were taken before | Recognize the scope of design work from the design problem | Able to convert vague information to design tasks with estimation of workloads |
| #3: IG | Ideas similar to common objects explored to students | More critical to the perceived ideas to solve the design problem | Able to check design functions and explore different kinds of possibilities |
| #4: DM | Show knowing how to use the decision matrix | Trade-off analysis between options | Recognize limitations and make authentic decisions |
| #5: SA | Able to cite the equations that were learned before | Able to anticipate the results from the analysis and check their relevance to the design work | Apply scientific principles for new modelling and analysis |

4.2. A “reverse” learning path

One difficulty of design problems is due to their open-ended nature. In contrast, students are more used to well-defined problems, which information is provided exactly to a right amount (no more, no less) and can be solved by prescribed procedures. In this context, one teaching approach is to let students initially work on specific design work and then let the design problems be more open-ended gradually in the curriculum. As the design process typically moves from vague problems to specific details, it implies that students may learn better from the end stages of the design process, and then they can have better skills to tackle the earlier stages. Then, a “reverse” learning path is suggested, and it can have three phases:

- Phase 1: Design verification
- Phase 2: Design development → design verification
- Phase 3: Conceptual design → design development → design verification
To illustrate, consider the design problem of snow removal. In Phase 1, we can provide several snow removal mechanisms. Then, we can ask questions such as how to compare the performance of different mechanisms. Ideally, these questions can engage students to do background research (e.g., check the benchmarks from the market) and scientific analysis (e.g., quantify performance). Semi-structured physical experiments are possible to train problem framing (e.g., how many experiments are needed), idea generation (e.g., design of test cases), and decision making (e.g., prioritize experimental work). The design work of this phase is specific because (partial) design solutions are already provided. Critical thinking with design quantification can be trained in this phase.

In Phase 2, we can first provide a single concept of a snow removal mechanism to initiate the stage of design development. Then, we can ask students for what design details are needed for further development. From the experiences of Phase 1, students can recall the performance measures of snow removal mechanisms and then identify important design parameters. Required design tasks may emerge naturally such as kinematic analysis and geometric modelling. If time allows, students can do design verification again to examine their own design work from the stage of design development.

Phase 3 represents a full design process, where students are expected to come up with a new design for snow removal. From the prior phases, students can recognize the required efforts of relevant design work from the prior phases of learning. Then, they can be more confident and realistic to manage the conceptual design (e.g., better chance to propose practical solutions with less guess work). Reworking on design development and verification can strengthen their design skills and help them recognize the importance of conceptual design.

In this learning path, generic design tasks can emerge repeatedly in different design stages. For example, the practice of quantifying performance in design verification can hone the skill in scientific analysis, and this experience can support the development of mathematical models in design development (another task in scientific analysis). Ideally, this repeated practice can help students develop better skills in scientific analysis for design work gradually.

5. CLOSING REMARKS

This paper classifies five types of generic design tasks, which are integrated with three design stages to articulate design work systematically. As this framework can provide common vocabulary for design education, it can potentially support multidisciplinary design work and inform the progression of design skills. As one future development, we can consider a three-dimensional matrix in Figure 2. The x and y dimensions represent the matrix between design tasks and stages, as used in Tables 3 and 4. The x and z dimensions form another matrix to inform the skill levels with design tasks, as shown in Table 5. This three-dimension matrix can present the connections of pedagogical information from “what to do” (in view of design tasks / stages) to “how to assess” (in view of skill levels). This three-dimensional matrix can be complex. Considering 5 task types, 3 design stages and 3 skill levels, there can be $5 \times 3 \times 3 = 45$ entries to this matrix. This may have implied the complexity of engineering design education that instructors need to address.

Currently, this design task-stage framework is applied to our mechanical capstone class. In future work, we plan to communicate this framework to other engineering disciplines for their feedback and possible adoption. As this framework aims to improve designer flexibility and integration skills, some instruments can be considered to measure its outcomes such as problem-solving efficacy (Heppner and Petersen 1982), transferable skills (Chan et al. 2017), and learner satisfaction (Joo et al. 2019).

![Three-dimension matrix of pedagogical information](image)

**Fig. 2. Three-dimension matrix of pedagogical information.**

**Acknowledgements**

This research is supported by the NSERC Chair in Design Engineering.

**References**

[1] Canadian Engineering Accreditation Board (CEAB), 2018 Accreditation Criteria and Procedures.
[2] Chan, C.K., Wong, G.C., Law, A.K., Zhang, T., and Au, F.T. “Evidence-based conclusions concerning practice, curriculum design and curriculum reform in a civil engineering capstone design course in Hong Kong,” Innovations in Education and Teaching International, vol. 54, no. 3, pp. 260-274, 2017.
[3] Cross, N., *Engineering Design Methods: Strategies for Product Design*, 3rd edition, Chichester, UK: Wiley, 2000.
[4] Dieter, G.E. and Schmidt, L.C., *Engineering Design*, 5th edition, New York, NY: McGraw-Hill, 2013.
[5] Dym, C.L., Little, P., and Orwin, E.J., *Engineering Design: A Project-based Introduction*, 4th edition, Hoboken, NJ: Wiley, 2014.
APPENDIX: DESIGN TASK-STAGE FRAMEWORK FOR OTHER DISCIPLINES

Table A1: Design task-stage framework applied for chemical engineering (Context: Design of a water treatment process to reuse waste water for a chemical process).

| Task Stage | Conceptual design | Design development | Design verification |
|------------|-------------------|--------------------|---------------------|
| #1: BR     | Study the technical requirements of the chemical process | Check the regulation of fresh water usage to benchmark the project’s performance | Check the convention of testing in industry practice |
| #2: PF     | Recognize the project’s scope | Decompose the detailed | Define three test cases for |

| Task Stage | Conceptual design | Design development | Design verification |
|------------|-------------------|--------------------|---------------------|
| #3: IG     | Develop approaches to separate large particles from waste water | Develop the design details such as pipe sizing | Use software tools to develop test cases |
| #4: DM     | Select one of three schematics that are developed in the team | Selection of pumps, valves and other components | Choose the software tools for simulation and testing |
| #5: SA     | Estimate the capacity to treat waste water under normal operations | Conduct the LCA for the overall process | Statistical analysis of testing results |

Table A2: Design task-stage framework applied for civil engineering (Context: Design of a greenhouse construction package for general customers).

| Task Stage | Conceptual design | Design development | Design verification |
|------------|-------------------|--------------------|---------------------|
| #1: BR     | Check similar greenhouse designs on the market | Check the building codes of greenhouse structures | Check existing designs for benchmarking information |
| #2: PF     | Define the project’s scope to structure and construction without piping | Define structural integrity and ease of construction as two technical focuses | Aim to verify if the new design is “safer” and “easier” compared to existing designs |
| #3: IG     | Explore ideas for scalable greenhouse structures for different households | Design a construction idea to minimize the use of specialized tools | Design test cases for normal and extreme conditions |
| #4: DM     | Choose the transparent materials for the structure’s envelope | Select the suppliers for construction materials | Choose between software testing or physical testing for some test cases |
| #5: SA     | Check similar greenhouse designs on the market | Check the building codes of greenhouse structures | Check existing designs for benchmarking information |