

Using Student Projects to Compare Model-Based Systems Engineering and Human-Centered Design

Luke Schroeder¹, Adam Hanna¹, Diya Amat¹, and Mohamed Elhawabri¹

¹*Department of Systems and Information Engineering
University of Virginia
Charlottesville, United States of America*

Abstract—This paper compares two approaches of systems design: Model-Based Systems Engineering (MBSE) and an iterative, human-centered design (HCD) approach. They are compared in a case study where both MBSE and HCD are applied to develop a makerspace information system. MBSE has become a common industry tool for architectural design of large-scale, complex, safety-critical systems that involve hardware/software integration and have long life cycles. It relies on rigorous upfront requirements and system modeling using SysML. In contrast, HCD is prototype-driven and requires continuous iterations given client feedback, incorporating users at every stage of development. This paper will discuss the team’s experience with both approaches in a case study centered around highlighting the perceived differences, strengths, and limitations of each method. Research outcomes stemmed from case study observations, comparison to existing research, and result validation via interviews with subject experts. The team’s findings agree with observed industry trends of MBSE being better suited to large, complex systems with strict safety requirements, long development timelines, and lower error tolerance. On the contrary, they perceived HCD to be better suited to smaller projects with accessible users and projects with uncertainty in requirements.

Keywords—Model-Based Systems Engineering, Human-Centered Design, Makerspace

I. INTRODUCTION

The motivation for this research stems from a project undertaken by the authors, which involved diagnosing operational deficiencies within an industrial makerspace and designing a system to address them. The primary objectives of the resulting system included tracking consumables, managing device usage, and monitoring capacity to make the makerspace easier to use and easier to manage. The authors, who engaged with this project as a culminating design experience of their undergraduate studies, applied Model-Based Systems Engineering (MBSE), using the SysML modelling language, on this project during the first semester. Once the problem was defined through external lab research and literature review, the

general workflow followed a weekly cycle of MBSE diagram deliverables, which progressed the solution architecture on a linear basis. This approach is utilized within the industrial contracting realm due to its fit towards detail oriented, large scale complex projects that need to fit established specifications. During the second semester, the team shifted towards designing a new solution to improve the operational efficiency of a different makerspace. The accessibility of these clients, as well as a condensed timeline, led to the choice of Human Centered Design (HCD) as the development methodology. Core tenets of this process include engaging early with the client and a range of users to diagnose needs and identify root problems, developing solutions iteratively with frequent user feedback, and being flexible to the voices of many stakeholders. The iterative, user-focused, and emergent nature of HCD offered an insightful contrast to the highly structured, client-isolated MBSE makerspace approach. In this paper, these projects are collectively used as a case study to compare student experiences with HCD and MBSE to existing research and the experiences of industry professionals.

II. LITERATURE REVIEW

A. Subject of Review

MBSE and HCD are well-established design methodologies that are used in industry. Existing scholarly work often focuses on their individual strengths or how they can be integrated into practice. However, much of this work looks at professional or real-world applications rather than how these approaches are learned and experienced in a classroom setting. Each specific method introduces different challenges for students, from understanding system modeling while using MBSE to navigating user-focused processes with the help of HCD. The existing focus on professional application highlights the importance of examining MBSE and HCD from a student viewpoint, which is the core theme of this research.

B. Exploring Model-Based Systems Engineering

MBSE has emerged as a response to limitations of document-based systems engineering. It replaces documents as the primary artifact to address the inconsistencies that occur when multiple authors produce independent documents that may disagree as the design goes through various changes [1]. In comparison, MBSE translates design information into

interconnected models that serve to improve tracking between requirements, testing, implementation, and maintenance [2]. When one part of the system is updated, those changes carry through to the rest of the model, which can lead to less manual labor and fewer errors. Interconnected models help prevent costly redesigns, minimize human error, and improve overall system reliability. Small inconsistencies can eventually lead to significant failures. Prior research supports this, emphasizing that model-based approaches enhance consistency and communication across system components, ultimately improving design quality and reducing lifecycle costs [3].

MBSE also helps improve testing and validation, which supports consistency across engineering work and enhances communication between teams [4]. Prior literature also highlights that MBSE is rarely adopted due to the steep learning curve. It can be hard to learn, but its benefits usually outweigh the challenges given the scale of what is possible to design through effective usage and improved understanding of systems [5]. Many MBSE tools are effective at analyzing systems after solution concepts have been developed, but they are less useful during the idea-generation phase [6].

C. Exploring Human Centered Design

HCD is an approach that looks at designing systems around the people who use them. It emphasizes the importance of understanding users and their needs. Users should be involved throughout the entire design process. HCD is flexible, its process is iterative, and designs should be tested using user feedback with teams from multiple perspectives and backgrounds [7]. It is different from the other engineering approaches which usually have user requirements as something that is fixed at the start of the process instead of something that changes with time. Effective design in this space comes from a willingness to experiment through prototyping, learn from prototypes that fail to meet user’s needs, and incorporate diverse perspectives to better address human needs [8]. HCD differs from traditional engineering approaches, which often treat user requirements as fixed at the beginning of the process rather than evolving over time. Its iterative nature highlights learning through experiences with users, building prototypes, and testing. In contrast, MBSE emphasizes structure and rigidity while HCD supports design creativity by encouraging teams to think about user needs instead of limiting ideas to what can already be modeled [9].

D. Integration and Comparison of Methodologies in Prior Literature

While there is not a great volume of sources linking HCD to MBSE directly, prior work by the Air Force Institute of Technology looked at the integration of Systems Modelling Language (SysML), and HCD. They pursued this combined methodology by first creating separate SysML and human centered diagrams, analyzing them to fully understand how humans and architectural elements interact with the system, and finally creating new diagrams which combined both viewpoints and highlighted the human-system intersection [10]. This mirrors the work done during the MBSE portion of the technical project. Activity diagrams were created to highlight user workflows, although without users validating the assumptions, it fell short of truly incorporating HCD principles.

III. METHODS

This paper treats both projects as case studies. They are complementary data sources, as previously mentioned, one applying MBSE and the other applying HCD. This project was conducted with the same team across both semesters, enabling a direct comparison of student experiences with each methodology. During both projects, the team collected data weekly on the tasks they engaged in, and hours spent on each type of task. Progress was visualized as a design timeline [12], and is shown in Fig 1. The MBSE case took place from September to December, with HCD directly following. Additionally, as the end of the second semester project neared, the team developed and answered a set of questions about their experiences with MBSE and HCD.

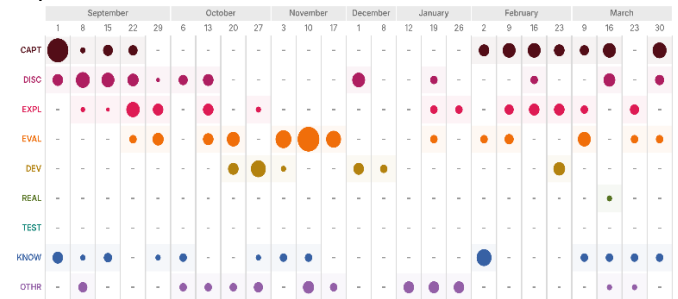


Fig. 1. Weekly Team Activity Type Log

Questions for the MBSE portion focused on the utility of specific modeling deliverables, the impact of limited client access, and the tradeoffs of a front-loaded structured process. HCD questions focused on how iterative prototyping and user feedback shaped design decisions, and where that flexibility becomes a limitation. Each member of the team completed the questionnaire individually. Then, all members of the team discussed their answers as a group to both embrace the diverse experiences of different team members and compare takeaways.

The team used expert-debriefing to increase the credibility of the results relative to each methodology. Questions were developed to interview a practitioner of MBSE from industry and two academics with expertise in human-centered approaches to design. One academic expert focuses on HCD integration and behavioral design ;and the other focuses on UI/UX design and human factors. The team decided to conduct these interviews using the same questions they answered individually to pinpoint where their conceptual shortcomings came from, clarify previous assumptions, and expand on areas where the team discussion showed uncertainty or disagreement. The interviews were recorded and transcribed. The transcribed interviews, along with notes taken during the interviews, were reviewed by the authors to identify themes.

IV. MAKER LAB CASE STUDY – MBSE VS HCD

A. Semester One: MBSE at an Industrial Makerspace

The first project was initiated in response to operational deficiencies identified within an industrial makerspace. The proposed system aimed to address pain points; these included tracking consumable inventory, monitoring device usage, and managing lab capacity. Once the problem space was scoped

through the project description and preliminary research, the team followed a structured weekly deliverable approach for defining solution architecture through SysML diagrams.

The highest-level view of the system was the Block Definition Diagram (BDD), which situated the Resource Management System (RMS) within the broader domain. The BDD identifies the system's primary stakeholders: lab managers, lab users, and the client company itself. It also showed how the RMS existed on the same domain level as the makerspace, as well as the tools within it. This top-level view establishes the operational context and stakeholder relationships from which all additional models are built.

From there, a structural diagram, which is a subtype of BDD, was created to decompose the RMS into its major structural components. This was divided across two high-level subsystems: a back end and a front end that split into other components contributing to the system (see Fig 2 and Fig 3). While this is not an MBSE task, all highlighted components within the diagram were selected to perform a trade study on, which was a client requirement needed to support design choices.

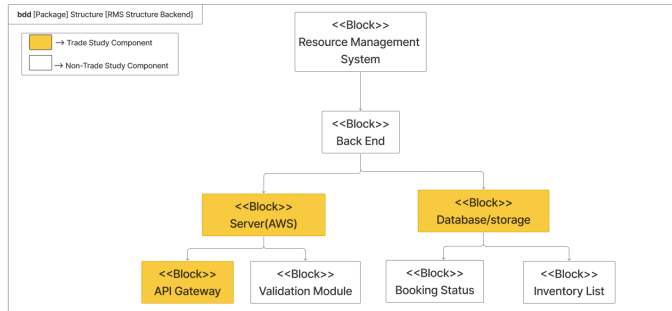


Fig. 2. Block Definition Diagram Back End example

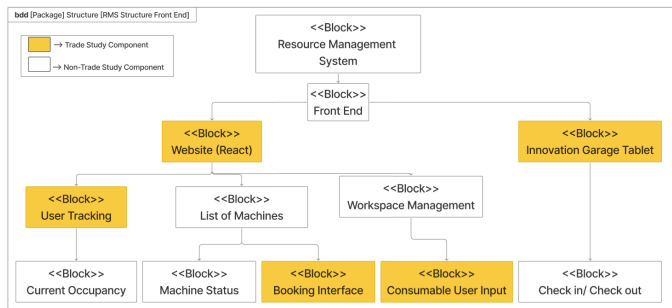


Fig. 3. Block Definition Diagram Front End example

Internal Block Diagrams (IBDs) specified how those components communicated at runtime. The internal context IBD (see Fig 4) traced data flows between core system components. It also listed specific ports where this data would be passed and received. External context IBDs captured how the system interfaced with outside stakeholders.

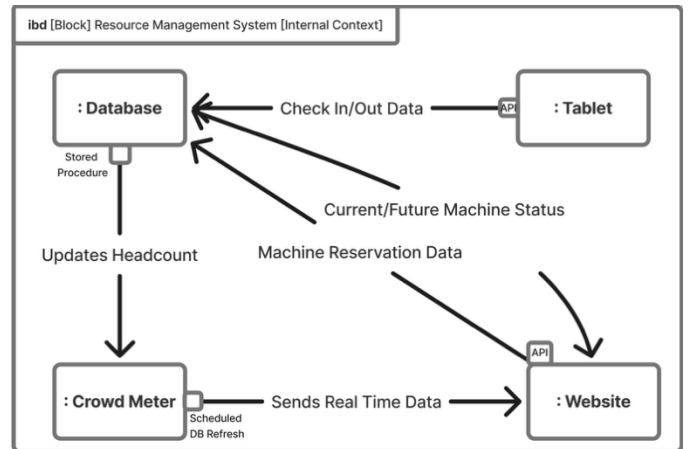


Fig. 4. Internal Block Diagrams, Internal and External Contexts

Activity diagrams were then created to guide workflows from a user's point of view through the system under key operational scenarios (see Fig 5). Activity diagrams serve to model the sequence of actions and decisions within a system, making implicit workflows explicit and traceable within the model. They are particularly useful for identifying where system components hand off responsibility to one another and where potential weak points may exist.

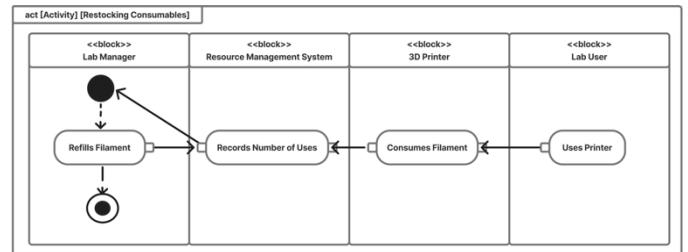


Fig. 5. Activity Diagrams, Restocking Consumables, Booking Hand Tool, Reserving Machine

Taken together, these diagrams produced a fully specified system architecture. This architecture was translated into a Papyrus model, which is a free online MBSE tool used for model creation. With respect to fig. 1, MBSE diagram creation falls under the evaluate/develop categories, where the team spent a considerable amount of time during the first case study. The MBSE process concluded at this architectural design stage, as the project was discontinued before the system could be built. This outcome, while limiting, is the expectation when conducting MBSE processes; the methodology is designed to produce traceable design artifacts that precede and inform implementation, not replace it.

B. Semester Two: Human Centered Design at an Academic Makerspace

The second project addressed booking and scheduling inefficiencies at the University of Virginia's architecture makerspace, where 3D printers were under high demand (especially during midterm and final seasons). Unlike the prior semester, the team had direct and continuous access to both lab operators and student users throughout the project. This access

made HCD a natural methodological fit given its dependence on ongoing stakeholder engagement.

The process began with contextual questioning: the team observed lab operations firsthand and conducted structured interviews with staff and users to find pain points before any design work began. Key findings included confusion over printer availability, the absence of a centralized reservation mechanism, and friction in the handoff between booking and actual equipment pickup. What came of these findings were not just static design requirements but a starting point that changed and evolved as the design continued to progress. Low-fidelity wireframes were produced early to show design concepts with the makerspace staff before starting functional system development. These were brought back to users in iterative testing sessions, where feedback drove several meaningful changes to the design, including revisions to the reservation flow, the addition of a booking confirmation notification, and simplification of the primary user interface. These sessions revealed that user uncertainty about system status and whether actions had been completed, which highlighted the need for improved system transparency. Higher-fidelity prototypes were then developed (see Fig 7), enabling users to interact with a near-realistic version of the system and surface usability issues that earlier sketches had not exposed.

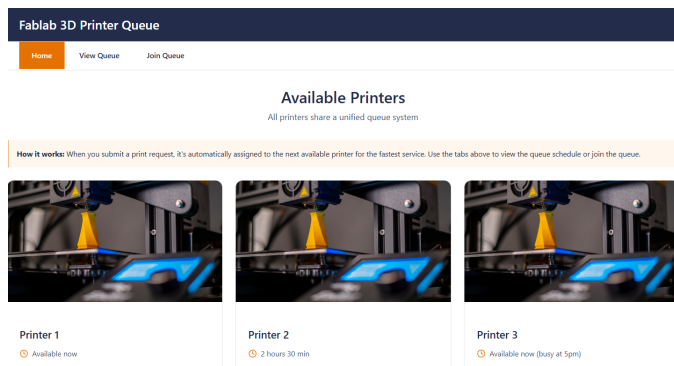


Fig. 6. Home Page Wireframe 1

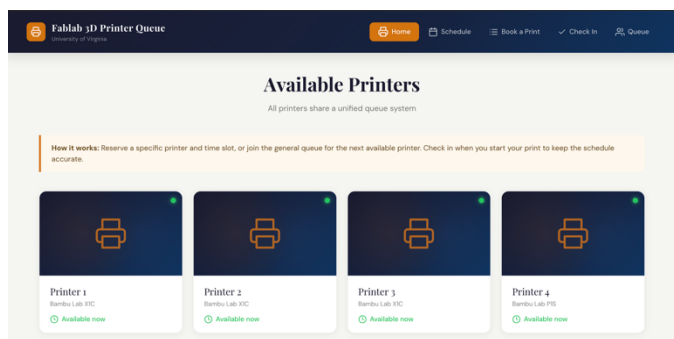


Fig. 7. Higher-fidelity prototype

C. Structural Comparison

The two projects showed how the differing methodologies shape almost every aspect of the design process. In MBSE, weekly progress was measured by diagram deliverables. This had reviewable artifacts that pushed the architecture through a linear flow. The end delivery of the term was a Papyrus model which contained all the section IV-A diagrams. In the HCD

semester, weekly progress was measured by the application of user feedback and the fidelity of the next prototype iteration. The end delivery of the term is a website that satisfies the discovered user needs. These structural differences in how requirements are determined, how artifacts are created, and the form of the final deliverables, form the foundation for comparative analysis.

V. RESULTS

A. Scale and Complexity

Formal systems architecture and requirement documentation is central to MBSE and effectively absent in the team's usage of HCD. This was acceptable on the scale of these projects, but would become a limitation in larger, more complex systems. Without formalized architecture, there is no uniformed record of how design decisions were made, how components interact, or how requirements map to the final system. As seen in the HCD case, the lack of this is manageable because the system is "simple" enough that the team can manage its structure mentally or write it out in a traditional document-based manner. As a system grows in complexity, involves more engineering teams to work on subsections, or requires regulatory approval, the absence of that documentation can create coordination issues that would be used to catch integration errors, or prove to a client or regulator that the system does what it claims to do.

B. Risk Distribution

Risk is managed differently as well. MBSE concentrates risk reduction at the front of the process, using modeling to catch design problems before the solution is built. HCD distributes risk across the entire development timeline, with each iteration surfacing and correcting issues incrementally. This was experienced during the second semester, as client responses to each prototype helped in shaping the next one. Early mistakes are expected and inexpensive in HCD, but the absence of an architectural plan can create integration problems in more complex systems. This sentiment was mirrored in MBSE fits projects with high complexity, strict safety requirements, and long timelines. HCD is better suited to smaller, more flexible projects with accessible users and changing requirements. A key structural difference underlying this distinction is the role of prototyping.

C. Problem Definition and Timeline

A theme emerging from the team's questionnaire responses was the difference in how each methodology approaches problem definition. In MBSE, boundaries, data flows, stakeholder relationships, and requirements are established upfront, whereas in HCD, the problem definition emerges gradually through iterative prototyping and direct user interaction. The team identified this distinction in responses to the individual questionnaire, though responses varied in how strongly each member felt the distinction was held. One student stated that "it was as if someone else had identified the problem and MBSE was being used to model the problem" showing that MBSE had a more linear trend in our own timeline through months August to December (see fig.1). When taking an HCD approach, the team noted that they spent multiple weeks interacting and observing stakeholders before they established what specific problems within a makerspace, they were going to

address implying no linear trend. This is also seen in fig. 1 as the team remained in the capture phase almost every week, which also corresponded to the need of client feedback to drive iteration.

D. Classroom Context vs Industry Practice

In the discussion, the team noted that the MBSE approach is highly dependent on the quality of the initial problem definition; with HCD, incorrect assumptions can be identified and corrected throughout development. During our expert debriefing, it was clarified that in industry, MBSE does involve iterative problem refinement through client-facing gates like the preliminary design review (PDR), and customer design review (CDR), suggesting the team's experience of a fixed and incomplete problem definition was more a product of the classroom context than the methodology itself.

E. Nature of Iteration and Deliverables

Iteration and client interaction are present in both methodologies but structured differently. MBSE, as explained by the industry professional, organizes iteration through formal review gates following the architecture definition, and produces a complete system architecture traced across requirements and structure. This makes the approach well-suited for large engineering teams and regulatory contexts. On the other hand, HCD incorporates continuous feedback starting from the first prototype. It also produces working prototypes that evolve into a final product through user validation, demonstrating value earlier but creating challenges for scalability and formal handoffs. Using HCD the creation of low and high-fidelity prototypes served as the main source for detecting user needs and validating design decisions throughout the semester.

F. Uncertainty and Subjectivity

Two methodologies also handle uncertainty and subjectivity differently. MBSE attempts to limit uncertainty before construction through requirements and modeling. HCD treats uncertainty as a starting condition, resolving it progressively through user feedback and iteration. HCD experts revealed that reliance on human facilitation introduces a risk of subjectivity, if the person running user feedback sessions unconsciously favors certain users, the resulting design can be affected in ways that are difficult to detect. MBSE, by contrast, moves toward objectivity through its structured modeling and formal requirements documentation, though our MBSE expert noted that even MBSE involves human judgment in how requirements are interpreted and how client feedback is incorporated during design reviews.

VI. DISCUSSION

Through the process of conducting the case study, reviewing existing related work, and discussing findings with industry professionals in both MBSE and HCD, the team uncovered insights on the student design experience. One of which is a fundamental misunderstanding regarding how they perceived client-oriented differences in the two methodologies. MBSE is intended to involve the client more so than how they performed it, and its intersections to HCD manifest not only through detailed up-front user specifications, but also later iteration during the next phase once the complete MBSE model is

delivered to the client. These next steps, revealed during the MBSE expert interview, are the PDR, and CDR, which mirror the iterative prototyping process in HCD, as the model is modified with respect to client feedback until a product is constructed. The key differences, as noted during the interview, are a result of which stages in the process the feedback is taken. MBSE leads with specifications and returns after the model is passed off for review, while HCD is more consistent, generally correlating to prototypes or model iterations. During the expert debriefing, it was confirmed that the gate-driven structure of MBSE, moving from architecture definition through requirements specification, design review, and verification reflects standard industry practice. Furthermore, that the team's process was recognizable as MBSE. The MBSE case, however, did not follow this described path, as the trial ended after the first model was completed in Papyrus, before the PDR and CDR could be conducted.

A. Method Viability Indicators

These methodologies are complementary in that they can be used to accomplish similar goals, although the case study uncovered that varying project conditions determine the viability of each. The experimental process detailed in Section IV sheds light on possible inflection points that determine when one development method is preferential to the other. This was visible through the rate at which different milestones were achieved on both sides of the case study, and group reflection on the process before expert validation. Variables that factor into this decision include project scope, user availability, and error tolerance. MBSE's architectural frontloading, detail-oriented modelling, and ability to display traceable schematics on a large scale make it the dominant option under the following conditions: Large complex systems with steep consequences on product error, and situations where a client organization can provide detailed requirements on what the solution should accomplish. Government work and larger corporate contracts often fall into these categories due to the need for well-documented, error-averse operations. An interesting counterexample to this usage trend is SpaceX, which does not use MBSE and favors a rapid prototyping methodology. This variance is expected in the real world and occurs due to SpaceX's dedication to quickly iterating and pushing out prototypes [11]. Failures occur frequently in this type of process, although SpaceX expressed a perceived value in error occurrence via the lessons learned. HCD mirrors this in that it involves pushing out quick changes to the system, taking in feedback, iterating the solution, and encountering errors along the way. This type of rapid, user personalized work was found by the team to favor smaller projects with fewer stakeholders, often aligning with the development work found in startups or other small organizations. It conforms with limited budgets, as viable products are created throughout the development timeline, with the final stopping point being client satisfaction or resource exhaustion. These findings are consistent with literature reviewed in section III. [1] confirms that MBSE's interconnected nature and model traceability reduced inconsistencies that occur in document-based systems engineering. Similarly, [4] found that MBSE improves cross team communication, which aligns the needs of large organizations with widespread corporate structures. Regarding

HCD, [7] agreed that users should be involved across the design process, although it differs from MBSE in that requirements should change throughout development, rather than be fixed up front. Contrary to the team's findings, the experts in HCD consultation revealed that any user-oriented work is fit for HCD, with large systems employing different tools. Both interviewees did agree that MBSE was a good fit for larger projects as it provided useful visibility on different levels of system abstraction.

B. Limitations

This work, despite indicating valuable trends in methodology viability, has several important limitations. With only two projects total that display the facets of each methodology, it is difficult to separate methodology effects from the clear differences in project scope, client availability, and team experience with each method. Further constraints stem from an incomplete MBSE, which contrary to the HCD case, was ended before solution building began. This implicitly biased how HCD was perceived as a faster, more error tolerant methodological foil to MBSE. An attempt to counteract this was to validate use case findings through the experience of industry professionals in their respective subjects, although this was also done with only one MBSE professional, and two HCD academics, thus limiting industry wide consensus to the views of a few. A further hindrance to research applicability was pointed out by the MBSE expert, stating that comparing a tool made for designing architecture and reporting progress, next to a method for how to build something was like comparing "Apples to Oranges". While this is a reasonable concern, the application of these in an academic setting accentuated case specific details in method viability.

VII. CONCLUSION

This research compared MBSE and HCD through two-one semester makerspace projects which showed how each method works with respect to varying project circumstances. The team favored MBSE to help organize large, complex systems, define requirements early, and create detailed solution architecture. Contrarily, HCD was preferred in situations where the end goal was less certain, upfront requirements were yet to be defined, and user interaction was a core pillar in solution viability.

An important note is that some of the resulting opinions about each method came from how the team experienced them in practice, which is not directly aligned with how they are used in industry. For example, the MBSE project didn't include the full feedback process, while the HCD project involved a lot of direct interaction with users. Instead of seeing MBSE and HCD as completely different or competing methods, they function better as complementary ideologies. HCD principles help diagnose what users need, and MBSE provides a method to turn those ideas into a well-structured and reliable system. In the future, it would be interesting to explore combining both approaches, especially employing HCD early on and MBSE later in the process. Overall, this project showed that choosing the right method really depends on the project, and being able to adapt is an important skill in systems design.

ACKNOWLEDGEMENTS

Reid Bailey – Systems Engineering Department at University of Virginia; Oliver Olsen – MBSE Subject Matter Expert; Gregory Gerling – Systems Engineering Department at University of Virginia; Leidy Klotz – School of Engineering and Applied Sciences at University of Virginia

REFERENCES

- [1] INCOSE, *Systems Engineering Vision 2020*, INCOSE-TP-2004-004-02, International Council on Systems Engineering, 2007.
- [2] SEBoK Editorial Board, "Model-based systems engineering (MBSE)," in *Guide to the Systems Engineering Body of Knowledge (SEBoK)*, v. 2.10. Hoboken, NJ: The Trustees of the Stevens Institute of Technology, 2025. [Online]. Available: [https://sebokwiki.org/wiki/Model-Based_Systems_Engineering_\(MBSE\)](https://sebokwiki.org/wiki/Model-Based_Systems_Engineering_(MBSE))
- [3] A. M. Madni and M. Sievers, "System of systems integration: Key considerations and challenges," *Systems Engineering*, vol. 17, no. 3, pp. 330–347, 2014.
- [4] S. Campo *et al.*, "Human-centric co-design of model-based system architecture," *Procedia CIRP*, vol. 119, pp. 434-439, 2023.
- [5] J. Pandolf, "Investigation of model-based systems engineering integration," M.S. thesis, Dept. Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, 2023.
- [6] S. Shoshany-Tavory, E. Peleg, A. Zonnenshain, and G. Yudilevitch, "Model-based-systems-engineering for conceptual design: An integrative approach," *Systems Engineering*, vol. 26, no. 6, pp. 783-799, 2023.
- [7] International Organization for Standardization, "Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems," ISO 9241-210:2019, Geneva, Switzerland, 2019.
- [8] T. Kelley and D. Kelley, *Creative Confidence: Unleashing the Creative Potential Within Us All*. New York: Crown Business, 2013.
- [9] G. A. Boy, "Human-centered design," in *Design for Flexibility: A Human Systems Integration Approach*, G. A. Boy, Ed. Cham, Switzerland: Springer, 2021, pp. 329–345.
- [10] M. E. Watson, C. F. Rusnock, J. M. Colombi, and M. E. Miller, "Human-centered design using system modeling language," *Journal of Cognitive Engineering and Decision Making*, vol. 11, no. 3, pp. 252-269, 2017.
- [11] E. Berger, "Elon Musk says SpaceX driving toward orbital Starship flight in 2020," *Ars Technica*, Feb. 28, 2020. [Online]. Available: <https://arstechnica.com/science/2020/02/elon-musk-says-spacex-driving-toward-orbital-starship-flight-in-2020/>
- [12] J. Turns *et al.*, "Using Design Signatures to Make the Invisible Visible: Designing Activities for Design Process Learning," *International Journal Of Engineering Education*, vol. 41, no. 4, pp. 928–946, 2025.