FACILITATING CROSS & BEYOND COURSE PROJECT-BASED SOFTWARE ENGINEERING LEARNING EXPERIENCES

Tim Maciag
University of Regina, Faculty of Engineering and Applied Science
tim.maciag@uregina.ca

Abstract –*Throughout each academic semester software engineering students are often provided with opportunities to explore open-ended project-based activities. Within the confines of specific courses, many of these explorations have resulted in interesting and impactful, partially or fully engineered software solutions. However, after student-developed solutions are explored, tested, and delivered within a classroom setting it has been the author's experience that they often don't progress beyond the course in which students explored and created them in. The results of this are missed opportunities for innovation as well as missed opportunities for further creative and collaborative explorations. This work-in-progress explores the following question: what could a model, process, and/or framework look like that would enable software engineering educators to create a learning environment that facilitates continued exploration, collaboration, and iteration of project-based student work beyond individual courses? This paper will describe an exploratory hybrid framework called ORhiDeCy that the author has designed and has been exploring in his courses over the last several years. This paper describes ORhiDeCy, an example of its successful use in the author's software engineering teaching practice, collaborator and student feedback, and the author's reflections and ideas for continued explorations.*

Keywords: active learning, project course, design project, student experiments, project-based learning, open systems, collaboration, software development management

1. INTRODUCTION

Throughout each academic semester software engineering students are provided with opportunities to participate in several courses (core, elective, and capstone) that provide them with opportunities to design and create iterative, open-ended software systems as part of learning [1, 2]. From the author's experience as a software engineering educator, many project-based learning student explorations have resulted in interesting and potentially impactful software engineered solutions. However, one of the unfortunate perceived deficiencies here is that after student-created software solutions are explored, tested, and delivered within the bounds of a course/semester they often don't progress beyond. Consequently, this has resulted in missed opportunities for the continued individual and collaborative explorations of developed software. Which may directly impact interesting local and global innovation possibilities and/or societal advancements.

This exploration-in-progress revolves around the following question: what could a model, process, or framework look like that would enable software engineering educators to facilitate a learning environment to enable continued exploration, collaboration, and iteration of project-based student work beyond the individual courses that they were explored and created in? Indeed, there exists individual models, processes, and frameworks that can aid iterative and collaborative activity accordingly. For example, open licensing of student works can enable opportunities for continued exploration of created software [3]. As well, iterative software engineering process such as agile/design thinking can promote collaboration and software iteration [4]. Further, traditional project management activities can help documentation and process flow of project deliverables [5]. However, often when these models, processes, guides, and/or frameworks are used on their own, they can lack key *ingredients* to successfully facilitate the potentially desired experience described.

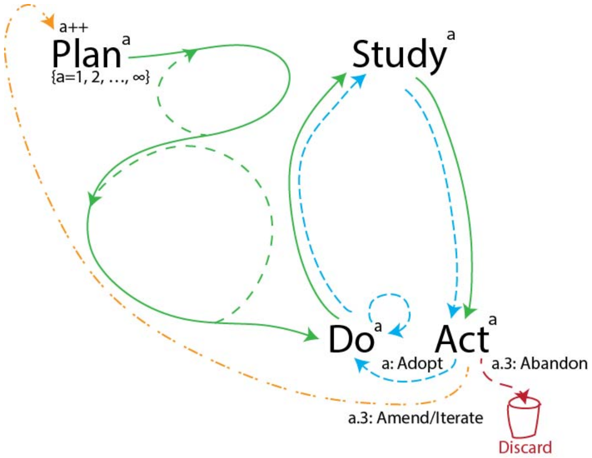
This exploration is inspired by the author's work in industry leading iterative knowledge management efforts in support of provincial healthcare technology systems in Saskatchewan. Here models, process, guides, and frameworks in lean management/the Toyota Production System, total quality management, adult education and learning, and systems and complexity thinking, were widely used by the author, and adopted with good success [6, 7, 8, 9, 10]. Since becoming an educator in the Faculty of Engineering & Applied Science at the University of Regina in 2017, the author has been continuing to explore, analyze, review, and *frankenstein* a framework in support of facilitating experiential and iterative learning experience so that opportunities for local and global innovation, societal advancements, and further creative and collaborative explorations of student-developed software can continue beyond individual courses.

The result of the author's action-based exploration-in-progress is the design, and continued iteration of a hybrid framework called ORhiDeCy. This paper will discuss the make-up of ORhiDeCy (pronounced oh-rye-deck-eye). As well, an experiential example of its use in facilitating cross and beyond course project-based learning experiences will be provided. The author's reflections on its success, collaborator and student feedback, and where future iterations might improve its use will be provided.

2. Framework Design

ORhiDeCy is an acronym representing the inspirations, guides, models, processes, frameworks, representative of its hybrid (*frankensteined*) framework (made up of ideas around Free/Libre **O**pen Source Software, **Rhi**zomatic Learning, **De**ming's PDSA model, Snowden's **Cy**nefin Framework). All inspirations, guides, models, processes, frameworks, representative of the hybrid framework are of equal importance to the overall educational experience desired. Beginning first with a rigorous planning phase. The plan phase is conducted by the author and with any outside/industry collaborators, if any. Represented by the "De" letters in ORhiDeCy, the planning phase commences by incorporating Deming's Plan-Do-Study-Act (PDSA) cycle [6], illustrated in Figure 1. The PDSA cycle is a well-known iterative business improvement technique where explorers *plan* a series of events/actions for some improvement activity, *do* the series of events/actions for some prespecified period, pause to *study* the results of the events and actions as related to continual improvement, and *act* on the outcomes of the improvement events and actions (adopt/study over time, abandon, or amend/iterate).

Figure 1. Deming's PDSA model for iterative exploration



To help the author, outside/industry collaborators, and engineering students understand, experience, and manage the complexity of each PDSA phase, Snowden's cynefin framework [7] is underlaid the PDSA model (The "Cy" in ORhiDeCy). Snowden's Cynefin Framework, illustrated in Figure 2, is a popular framework for visualizing complex knowledge management activities and flow. This, from the perspective of five knowledge domains: Complex, where knowledge activities include probe-sense-respond; Chaotic, where knowledge activities include act-sense-respond; Complicated, where knowledge activities include sense-analyze-respond; Obvious/simple, where knowledge activities include sense-categorize-respond; And disorder. An adapted Cynefin Framework is illustrated in Figure 3. At any given time during knowledge work, knowledge workers may be fluidly moving between knowledge domains (Snowden often uses a chef metaphor for knowledge activities and movement between knowledge domains. https://bit.ly/cynefin-chef (March 2022).

Figure 2. Snowden's Cynefin Framework (adapted)

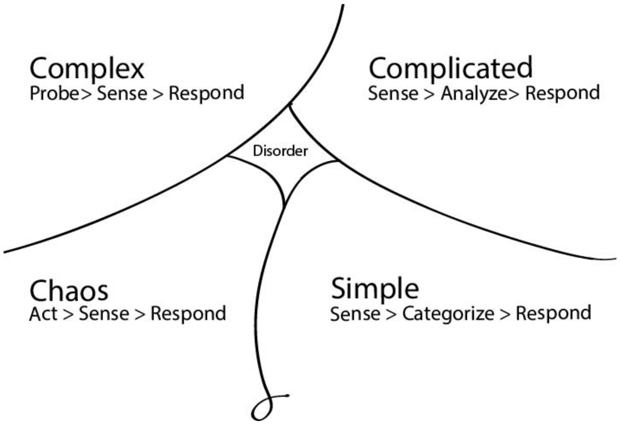
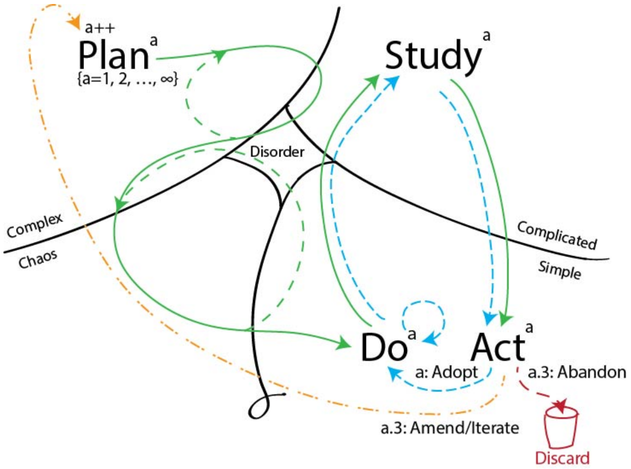


Figure 3. Snowden's Cynefin Framework (adapted) combined with Deming's PDSA model for iterative exploration



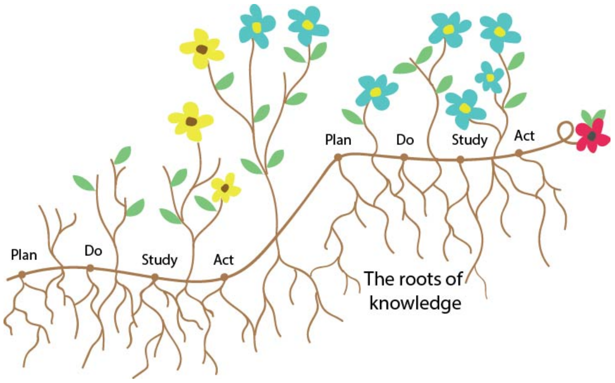
As illustrated in Figure 3, the planning phase is a *complex* knowledge activity where an educator and their collaborators draft and agree upon a course's project-based learning strategy and experience (and its theory if applicable). This is considered a complex-complicated-chaotic activity as educators and collaborators may experience varying degrees of ebb-and-flow as discussions, changes, and rigorous course, project, and stakeholder details are fleshed out. This variance depends on how well-formed the course and project-based experience aims to be. For example, whether the project idea is to be of the educator's choosing, or if outside/industry collaborators are involved and have unique requirements requiring trade-offs and new curriculum ideas, or whether the educator expects students to come up with their own project ideas. All of these increase the complexity of the plan. The complexity of drafting the project plan is visualized by the iterative, back-and-forth oscillations between the complex, complicated, chaotic, and disorder knowledge domains. This is representative of the give and take of the various stakeholders (educators, any outside/industry collaborators, and/or the students).

The project plan provides the foundations for the cross and beyond course project-based learning experiences. The plan details: A description of the project; A description of the challenge, opportunity, and/or problem area; Information about the collaborating outside/industry partners (if required); The software engineering design and development approach to be used (typically the process engineering learners were exploring as part of the course topics); The technology and programming environment to be used (as defined by the author, collaborators, or learners depending on the project); And any assumptions and constraints (such as project scope, time/commitment, quality of creative works, and software licensing, etc.).

As a course commences and continues throughout a semester, the "do-study-act" phases commence and iterate. As illustrated in Figure 3, the knowledge domains here include simple, disorder, and complicated. As the author provides course-based knowledge as per its curriculum, students act on their learning by sensing-categorizing-responding to newly acquired knowledge creating opportunities. This can be described as *simple* knowledge activities as students are taking-in, reflecting, and applying (doing/studying) knowledge provided by an educator (like reading a recipe and acting on baking steps). This can also be described as *complicated* knowledge activity as students seek to make sense by applying current and past understanding and knowledge to their learning as well. As students explore and apply their newly gained knowledge, they iteratively experience *simple, disorder, and complicated* knowledge as they *messily* put the pieces together. These iterations end with the final "act" phase at the end of a course. Here is when conversations between an educator, outside/industry collaborators, and, at times, engineering students discuss the potential to iterate and continue exploration of project work (adopt or amend/iterate) or abandon it.

For cross and beyond course project-based learning explorations to work as described, the licensing model used for the creative project outputs of students is a key aspect. Representative of the "O" in ORhiDeCy, and stemming from the complex "planning" phase, the exploration and use of free/libre open-source software (FLOSS) licensing models [3] is discussed. Here, an educator may require the addition of such knowledge within the course curriculum so that students (and collaborators) can understand the trade-offs between licensing models (open and proprietary). Before any project-based work commences an educator, outside/industry collaborators, and students must have agreement upon a license model. Although use of a culturally open licenses, such as many FLOSS licenses, for student creative works is encouraged (the license of the author's preference is a Creative Commons attribution ShareAlike license, BY-SA 4.0. Online (March 2022) at: https://creativecommons.org/licenses/by-sa/4.0/), other licensing models can be used, including proprietary ones. However, use of FLOSS licenses can provide more flexibility for continued iteration of creative works.

Figure 4. Side view of ORhiDeCy illustrating cross and beyond course-based learning knowledge connections (roots) and interesting outcomes of student works (blooms/shoots)



The final aspect of facilitating cross and beyond course project-based learning with ORhiDeCy is use of the idea of rhizomatic learning (the "Rhi" in ORhiDeCy). Rhizomatic learning is a learning pedagogy inspired by Deleuze and Guattari and their metaphorical description of knowledge formation and growth as a rhizome [11, 12]. "Botanically, rhizomes are modified subterranean stems that […] orient their growth perpendicular to the force of gravity but retain the ability to spawn geotropic shoots that can become independent," that which is "made up of a number of semi-independent nodes, each of which is capable of growing and spreading on its own, bounded only by the limits of its habitat." ORhiDeCy integrates the idea of rhizomatic learning to help visualize, realize, and document collaborations (cross and beyond-course), documenting the personnel, knowledge, and contributions of everyone involved in the collaborative and iterative learning experiences. As well, public repositories containing explored work via team GitHubs are made accessible for continued (decentralized) explorations by a global citizenry. A sideways view of the ORhiDeCy model is provided in Figure 4. As illustrated, as per PDSA/cynefin knowledge interactions, *plant shoots*/*blooms* of knowledge emerge that are representative of the interesting and impactful engineering ideas that result because of project-based learning and collaborative experiences. As well, interconnected "*roots* of knowledge" form with each iteration of a course experience, illustrating the documented history of collaborator expertise and/or individual and social project contribution(s).

3. Exploration of ORhiDeCy

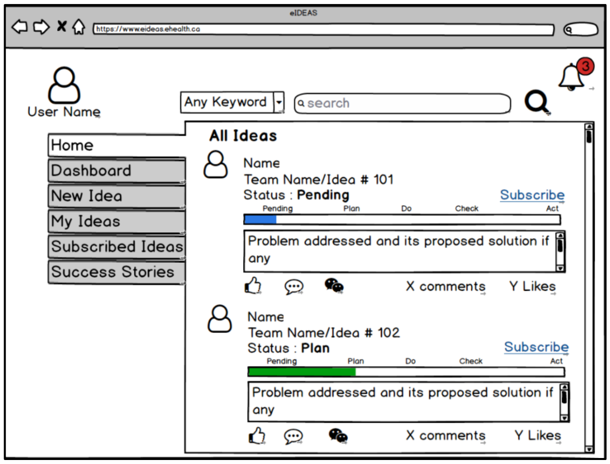
The author first evaluated ORhiDeCy in two software engineering courses at the University of Regina during the 2018-2019 academic years. The first course was a graduate-level software systems engineering (SSE) course titled "Research Topics in Computer Supported Collaborative Work," numbered 805, that occurred in the spring of 2018 (May-June). The second course was a fourth-year undergraduate SSE course titled "Software Systems for Computer Supported Collaborative Work", numbered 405, that occurred in the fall of 2018 (September-December). Although the learning activities slightly differed in the stated courses, the learning theory and topics in each course were similar. Both courses were also designed to have a cross and beyond course project-based learning activity that was facilitated by ORhiDeCy.

The project-based learning activities in both courses involved a collaboration with eHealth Saskatchewan (eHS, as of March 2022: https://www.ehealthsask.ca/), a local healthcare technology support organization. The author's eHS point of contact was Mrs. Janice Reeves, Continuous Improvement Specialist. The project description as collaboratively defined by eHS and the author was to explore the engineering of a software solution that aimed at helping eHS more effectively foster an internal culture of innovation through collaborative idea sharing. The underlying goals of the software system was to improve staff engagement and collaboration, support idea exploration and the advancement of shared ideas, and to improve employee morale, mental well-being.

3.1. Beginning with 805

The first course where ORhiDeCy was used was a graduate-level SSE course numbered 805. Two student teams were created for project-based learning activities, each with three students. Early in the course, student teams were provided with the eHS-author collaboratively drafted project plan. The project plan included: A description about the collaborative cross and beyond course project; A brief introduction regarding the challenge and opportunity; A primer on eHS; An overview of the software engineering design and development approach (collaboratively decided by eHS and the author as an Agile/Scrum Framework [13] approach); The licensing model to be applied (collaboratively decided by eHS and the author to be an MIT license (https://opensource.org/licenses/MIT (March 2022); And assumptions and constraints which included that the scope of the project in 805 would be solely to explore design ideas (not any development), students would utilize the Balsamiq prototyping software to explore and iterate design ideas (https://balsamiq.com/, (March 2022). A high-fidelity prototype is representative of a *blueprint* for future software development/coding), and that students would set up and use a publicly accessible GitHub for all creative outputs of their project-based work (https://github.com/ENSE-805-Spring2018-eHealth (March 2022). GitHub (https://github.com/ (March 2022) is an industry-class version control system).

Figure 5. One of the 805 student team design proposals



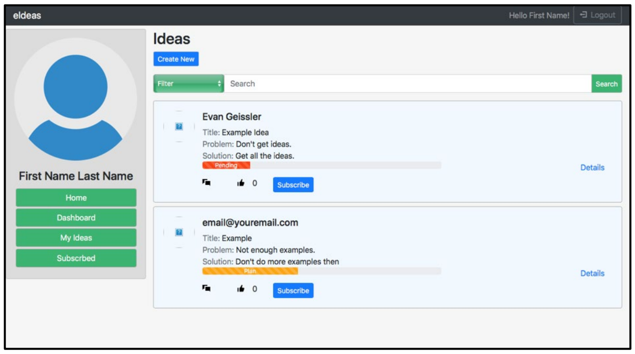
Along with exposure to learning theory in community-centred design, over two "do-study-act" ORhiDeCy cycles (each cycle called a "sprint" as per the Agile/Scrum Framework), 805 students explored and iterated design ideas in collaboration with eHS throughout the duration of the course. Figure 5 illustrates the output of a partial view of one of team's high-fidelity prototype design. As part of the engineering learning experience, both groups conducted a user evaluation with eHS employees at eHS main headquarters. 805 students "studied" (collected, analyzed, and documented) comments from eHS employees and provided summaries of results obtained. Project-based learning creations by 805 students were stored in publicly accessible team GitHubs.

3.2. Cross-course iteration with 405

After the completion of the 805 SSE course, the author and eHS completed a final "do-study-act" ORhiDeCy cycle by reviewing and reflecting on next steps regarding the iteration of design works in a future SSE course. The review and retrospection resulted in a positive outlook for continued exploration. With the next steps approved by eHS and the author, both parties reviewed the initial project plan drafted for the 805 course and proposed slight revisions for the next iteration in the author's 405 course. Given that the educational and project-based goals in 805 revolved around design, and that both designs delivered by 805 students were deemed by eHS and the author as good for iteration and continued exploration, it was decided that the 405 course would focus on development of both proposed designs for project-based learning.

The revisions in the project plan included: An evolved description about the collaborative cross and beyond course project-based learning experience, providing 405 students with access to 805 student contacts and their creative works (illustrating the "roots of knowledge" connections); Defining a development/coding environment (collaboratively decided by eHS and the author to be C# programming language (https://docs.microsoft.com/en-us/dotnet/csharp/ (March 2022)) and MySQL database environment (https://www.mysql.com/ March 2022)); And revisions to the assumptions and constraints which included a revised scope by focusing on development/coding and, like in 805, students in 405 would set up and use a publicly accessible GitHub for all creative outputs of their project-based work (https://github.com/ENSE-405-Fall2018-eHealth, (March 2022)).

Figure 6. One of the student team design proposals from 405 (based on the design in Figure 5)



Evan Geissler
Title: Example Idea
Problem: Don't get ideas.
Solution: Get all the ideas.

email@youremail.com
Title: Example
Problem: Not enough examples.
Solution: Don't do more examples then

Four student teams were formed, each with four to six students. Each team was randomly selected to iterate one of the two designs delivered by student teams in 805. 405 student teams set out to develop/code their design over four "do-study-act" ORhiDeCy sprints. With two later sprints that included a second user evaluation with eHS employees and a final sprint for product delivery. Figure 6 illustrates the output of a partial view of one of team's developed designs (as based on the design illustrated in Figure 5).

3.3. Beyond-course iteration with eHS

After completion of the 405 course, the author and eHS completed a final "do-study-act" ORhiDeCy cycle by reviewing and reflecting on next steps regarding the integration of engineering works in the eHS environment. Of the four software systems developed by 405 student teams and based on data and analysis from the user evaluation between 405 students and eHS employees, one of the software systems was chosen for integration within the eHS environment. It was later integrated by eHS employees into their internal technology ecosystem.

4. DISCUSSION

There were several interesting insights when discussing cross and beyond course project-based learning experiences facilitated by ORhiDeCy with students in both courses and with eHS. Firstly, most SSE students in both 805 and 405 commented favourably on collaborating cross and beyond courses with an outside/industry partner and in tackling a real-world problem collaboratively. eHS also commented favourably on the experience. Although it is important to note that in recent conversations in March 2022 with eHS that the software system is no longer in active use. It was indicated that priorities had shifted. The author is still encouraged that the student creative works are still publicly available via class/team GitHubs for anyone interested in continuing the exploration, as in the decentralized spirit of rhizomatic learning.

Project licensing also provided an interesting discussion in both courses. Although the author thought there would be more student push back in the requirement of an open license, as it is the right of students to apply their own licensing on their creative works, it was clear some students didn't fully understand the topic of licensing. The author has since been adding knowledge content on this topic to the curriculum of all courses that utilize ORhiDeCy for project-based learning. The requirement of a publicly accessible version control system to store student-created project knowledge was also envisioned to be an issue as some students may not want their learning works to be publicly available. Currently this is a rigid requirement of ORhiDeCy (although it can still work with a closed system. Iterative capacity is limited). It should be noted that the author has been exploring ORhiDeCy in several SSE course offerings since the initial experience described here without issue.

One of the more interesting and delightful experiences enabled by ORhiDeCy was in how many SSE students in both courses commented favourably on cross and beyond course collaboration and knowledge sharing. The author witnessed students in both courses forging strong knowledge bonds which only could have occurred by documenting and maintaining said connections (the "roots of knowledge"). The author witnessed 805 students asking to participate in 405 project-based learning activities, and students in both courses wanting to help eHS integrate the selected system in their environment and to learn more about the organization and opportunities for work. The only real issue that arose was that some 405 students struggled with iterating a design proposal not of their own creation. Students cited a perceived lack of creative control and freedom. The author is currently exploring how best to update curriculum as he believes this an important skill and experience as often in industry software engineers are required to iterate on existing works.

5. Conclusion & Future Work

This exploration-in-progress revolved around the following question: what could a model, process, or framework look like that would enable software engineering educators to facilitate a learning environment for the continued exploration, collaboration, and iteration of project-based student work beyond individual courses? In support of exploring this question the author has been *frankensteining* and exploring a framework called ORhiDeCy. This paper discussed the initial use of ORhiDeCy over two software engineering courses, and in collaboration with an outside/industry partner – eHealth Saskatchewan. Knowledge on the make-up of ORhiDeCy and insights into its initial use were provided. As the initial use of ORhiDeCy proved interesting, the author has continued to explore its use in other software engineering courses. With each exploration, the author has gained further insight and knowledge in support of it providing an effective framework for facilitating cross and beyond course project-based learning experiences.

Other courses where the author has explored ORhiDeCy with successful results include undergraduate and graduate-level courses on people-centred and community-centred design. All courses have had (and currently have) an outside/industry collaborator where student engineered works are made accessible on the web. Learning experiences have included a collaboration with Regina Fire and Protective Services (RFPS), where students worked with RFPS on the engineering of a mobile app in support of fire safety (the app is currently live on the iOS store (https://apple.co/3JJtNrw (as of March 2022) and Google Play store (https://bit.ly/ReginaFireSafetyAndroid (as of March 2022)), collaborations with RCE Saskatchewan, and currently, a collaboration with the Regina Food bank. Although all cross and beyond course project-based learning explorations have proven interesting and thus far, there remains more to explore as the author continues to collaborate, iterate, and evaluate.

Acknowledgements

The author wishes to thank and acknowledge the support and contribution of industry collaborators from eHealth Saskatchewan (eHS). Specifically, Mrs. Janice Reeves, eHS Continuous Improvement Specialist, who took on the role of the product owner and gave significant time in working with the author and students during all project phases in both courses. The author would also like to thank and acknowledge all students in both courses for the hard work and professionalism.

References

[1] D.L. Dekker, "Difference between open-ended projects and design projects," *Technology Based Re-Engineering Engineering Education*, Vol. 3, pp. 1257-1259, 1996.

[2] C.L. Dym, A.M. Agogino, O. Eris, D.D. Frey, and L.J. Leifer. "Engineering design thinking, teaching, and learning," Journal of Engineering Education. Vol.94. no.1. pp. 103-120, 2005.

[3] E.S. Raymond. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly. 2001 (Revised Edition). 241 pp. {ISBN: 0-596-00108-8}.

[4] T. Lindberg, C. Meindel, and R. Wagner. *Design Thinking: A Fruitful Concept for IT Development?* In: Meinel, C., Leifer, L., Plattner, H. (eds) Design Thinking. Understanding Innovation. Springer, 2011.

[5] D.C. Barrett. *Understanding Project Management (2nd ed)*. Canadian Scholars. 2021.

[6] W.E. Deming. The New Economics (2nd ed.). The MIT Press. 2000. 247 pp. {ISBN-13: 978-0262541169}.

[7] D.J. Snowden and M.E. Boone. "A Leader's Framework for Decision Making." Harvard Business Review. 2007.

[8] T. Ohno. *Toyota Production System: Beyond Large Scale Production*. Productivity Press. 1988.

[9] C. Rose, T. Nichols, D. Hackner, J. Chang, S. Straube, W. Jooste, H. Sawe, and A. Tenner. *Utilizing Lean Software Methods to Improve Acceptance of Global eHealth Initiatives: Results from the Implementation of the Basic Emergency Care App*. JMIR, vol. 5, no. 5, 2021.

[10] D. Duncan. *Patient First Review Update: The Journey So Far and the Path Forward*. 2015. Available as of March 2022 from: https://bit.ly/eHealthSk

[11] D. Cormier. "Rhizomatic Education: Community as Curriculum." Journal of Online Education. Vol. 4, No. 5. 2008.

[12] G. Deleuze and F. Guattari. *A Thousand Plateaus: Capitalism and Schizophrenia*. University of Minnesota Press. Pp. 632. 1987 {ISBN-13: 978-0816614028}

[13] K. Schwaber and J. Sutherland. *The Scrum Guide: The Definitive Guide to Scrum, The Rules of the Game*. 2020. Available as of March 2022, from: https://scrumguides.org/scrum-guide.html