Implementing No/Low Code Development in a Systems Analysis and Design Course

Jack Crumbly
Tuskegee University

Daniela Field
Mendix | MIT

No/Low Code is the next phase in the code evolution. Enterprise companies build complex solutions using platforms that are primarily no/low code. According to Gartner - "By 2024, low-code application development will be responsible for more than 65% of application development activity." Thus, adding a no/low code tool at Tuskegee University's System Analysis & Design class prepares students for the future. During class, students select projects, analyze, design, build, and implement full systems using the latest expertise tools and processes such as the agile The paper highlights why tools such as Mendix are important to incorporate in the classroom.

Keywords: innovative classroom activities, no code, low code, Mendix platform, curriculum refresh, student hands-on activities, agile process, building solutions, system analysis and design

INTRODUCTION

This paper showcases how a traditional System Analysis and Design course where students do not necessarily build a solution will benefit from actually building a system within a semester. In addition, the paper will highlight how the Mendix platform can be implemented and used throughout the semester to enhance the student's learning experience. Organizations request for early career talent to be ready to engage in the marketplace and they request for skills that encourages critical thinking while providing solutions at a rapid pace. Software and other service-based firms rely on critical thinking tools such as design thinking to develop new applications and services (Abdalhamid and Mishra, 2017). Students are expected to have hard and soft skills. They need to both understand the business needs and implement the solutions with technical understanding. In 2020, the Gartner research firm reported that at least fifty percent of all new business applications will be created using high-productivity toolsets (Knupffer et al., 2019). Implementing these tools in courses is critical to the success of students seeking successful early career placement within organizations. Courses such as Systems Analysis and Design discuss both the Systems Development Lifecycle (SDLC) or Waterfall method and Agile methods such as Design Thinking and SCRUM to give Business, Information Technology and Computer Science students an understanding of how applications are developed within organizations (Shelly and Rosenblatt, 2012). In addition, the System Analysis and Design course at Tuskegee provides students with all the technical competencies they need to build systems. For example, the data management and structure, user flow and experience, business logic and workflows and user roles and access. In the next sections we will focus on

the development process and provide knowledge in the technical skills that the students gain and learn in the classroom.

DEVELOPMENT PROCESS – WATERFALL VS AGILE METHODOLOGY

It is imperative to have the right tools and knowledge to build solutions. However, the process of how the solutions and software is built is just as – if not more – important then the tools used. Without the right process in place to do the planning, development and deployment, and define communication cadences, projects are bound to fail. Thus, we spend time to define the various industry standard processes in place. In addition, we do a design thinking workshop for the students to gain deeper insight and problem solving skills applicable to the real world. Students are introduced to waterfall, agile and design thinking concepts in the beginning of the classroom.

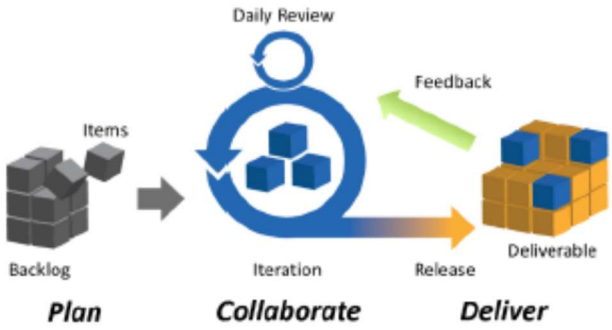
Classroom Methodology

The Waterfall/SDLC method is a method that requires the following: plan, analyze, design, implement and support of an information system (Shelly and Rosenblatt, 2012). Figure 1 describes the waterfall development method. The SDLC/Waterfall method uses a sequential process of collecting user requirements, development, testing and deployment of an application (Frydenberg et al., 2017). The waterfall methodology is thoroughly covered in most System Analysis & Design books and classrooms. This process can take up long periods of time. In addition, the level of success is unknown without user feedback. Even though, companies have been developing software solutions since the 1960s, project still fail at a high rate and are usually over budget and late (Cerpa and Verner, 2009). https://dl.acm.org/doi/pdf/10.1145/1610252.1610286.

FIGURE 1
THE SDLC/WATERFALL METHODOLOGY FOR APPLICATION DEVELOPMENT

Most companies in today's world, are focused on implementing the Agile method across their organizations. Figure 2 describes the agile method. An agile methodology process stresses a team-based effort focused on team members working in cycles rather than individual development processes (Shelly and Rosenblatt, 2012). This provides the team with the ability to design, build, test and reflection at each cycle. Although the process reduces time of development, it can increase risks in the development process. Risks are reduced by incremental steps in short time intervals (Shelly and Rosenblatt, 2012).

FIGURE 2
THE AGILE METHODOLOGY FOR APPLICATION DEVELOPMENT



The course is being taught using project-based learning method in a simulated environment (Levi. 2011). The identified project-based learning methods using simulations provide students with hands-on experience in project management. It provides students with the ability to learn through applications such as Mendix No/Low Code solutions.

After the course starts, the students are introduced to the two methods of SDLC/Waterfall method and Agile method. In addition to the two methodologies, at Tuskegee's System Analysis & Design course, the students are introduced to design thinking process of development. Students experience an initial design thinking exercise related to a common experience based on the Stanford Design School (Appendix 1). During the class, the students are given a situation such as improving the gift-giving experience. For example, how would a student go about improving their gift-giving. They think about the process that goes into gift-giving and are asked to think how to improve the experience. The class structure is as follows:

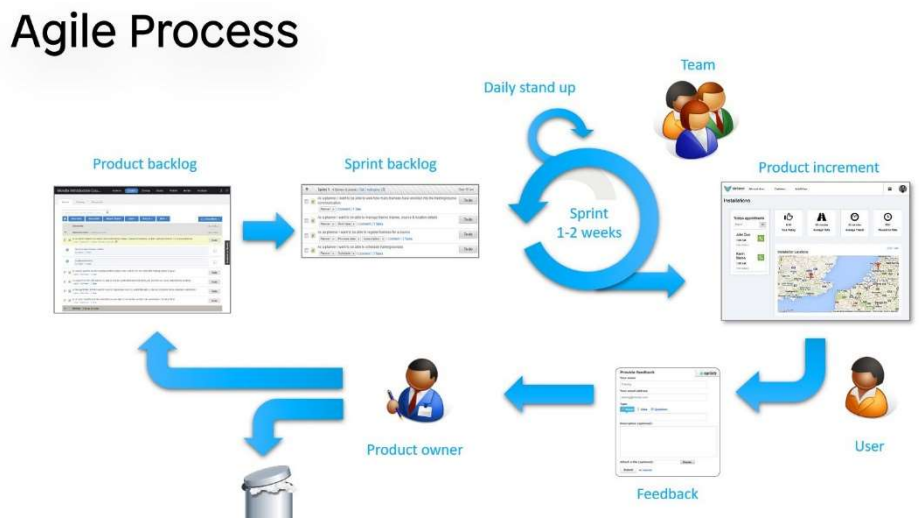
1. 2 sessions x 4 minutes – Students gain empathy and interview each other on their gift giving experience
2. 2 sessions x 4 minutes – Students dig deeper to better understand the problem
3. 3 minutes – Student spend time capture their findings
4. 3 minutes – Students define the problem statement
5. 4 minutes – Students sketch 5 ways of meeting the user's needs
6. 2sessions x 4 minutes– Students will share their solutions and capture feedback
7. 3 minutes – Student reflect and generate a new solution
8. 10 minutes – Students Build a Solution
9. 2 sessions x 4 minutes each – students will share their solution and what they learned during the experience

After the students take one class period to gain exposure to the whole design thinking process. They take time outside of the classroom to start learning the Mendix platform. Design thinking and the agile methodology is one of the first items that the Mendix academy focuses on, as the students are learning the tool. The students bring their design thinking experience into their projects. Mendix Academy offers a free Rapid Developer Course (Mendix 1) with a certification at the end. The students complete the rapid developer course at the beginning of the semester, so they have the knowledge to continue building in their projects. In addition, once they pass the Rapid Developer course, they gain a certification they can use in their curriculum vitae and showcase as experience. After students pass the Mendix Rapid Application development process, teams are selected, and the project selected during the beginning of the semester starts to be implemented. Students choose roles in the application development. The roles range from Project Manager to developer to customer and end user.

How Students Use Mendix to Integrate the Agile Process

The agile methodology is implemented throughout the Mendix platform. Students sign up for a free account when they use their university emails. Once they log in into the Mendix platform, they can start to create a project. Within the project, students will find the full agile process integrated in the platform.

FIGURE 3
OVERVIEW OF THE AGILE PROCESS



The students will create a product backlog and list of requirements. Afterwards they will spend some time prioritizing the requirements into sprints. During a semester, the students can run anywhere from 2-5 sprints of development cycles. At the end of each sprint, the students have a product that they can demo to the professor or the class. The students can use a free sandbox environment, where they deploy their projects and share them with everyone. The benefit of iterating and collaborating with stakeholders and getting feedback from the professor and others, exposes the students to real life scenarios and prepares them with project and development experience. In addition, the platform offers a built-in mechanism to provide feedback to the students.

Mendix is an enterprise platform that partners with companies and provides guidance on the whole application lifecycle. There is an agile e-guide to provide companies with tips on best practices (Mendix 2 Agile Guide) https://www.mendix.com/agile-guide/ The guide can be used in the curriculum for students as well – giving the students insight on what companies implement in their organizations and how they incorporate agile processes. Next section will focus on the development and how it is technically.

UNIFIED MODELING LANGUAGE AND DATA MANAGEMENT

Tuskegee's System Analysis and Design course is open to students from a variety of majors. Some have database background and most do not. The majors include accounting, finance, information technology and computer science. Pending the experience of team members, additional lecture is required for students to understand the basics of database structures and why they are relevant to building systems and solutions.

Classroom Methodology

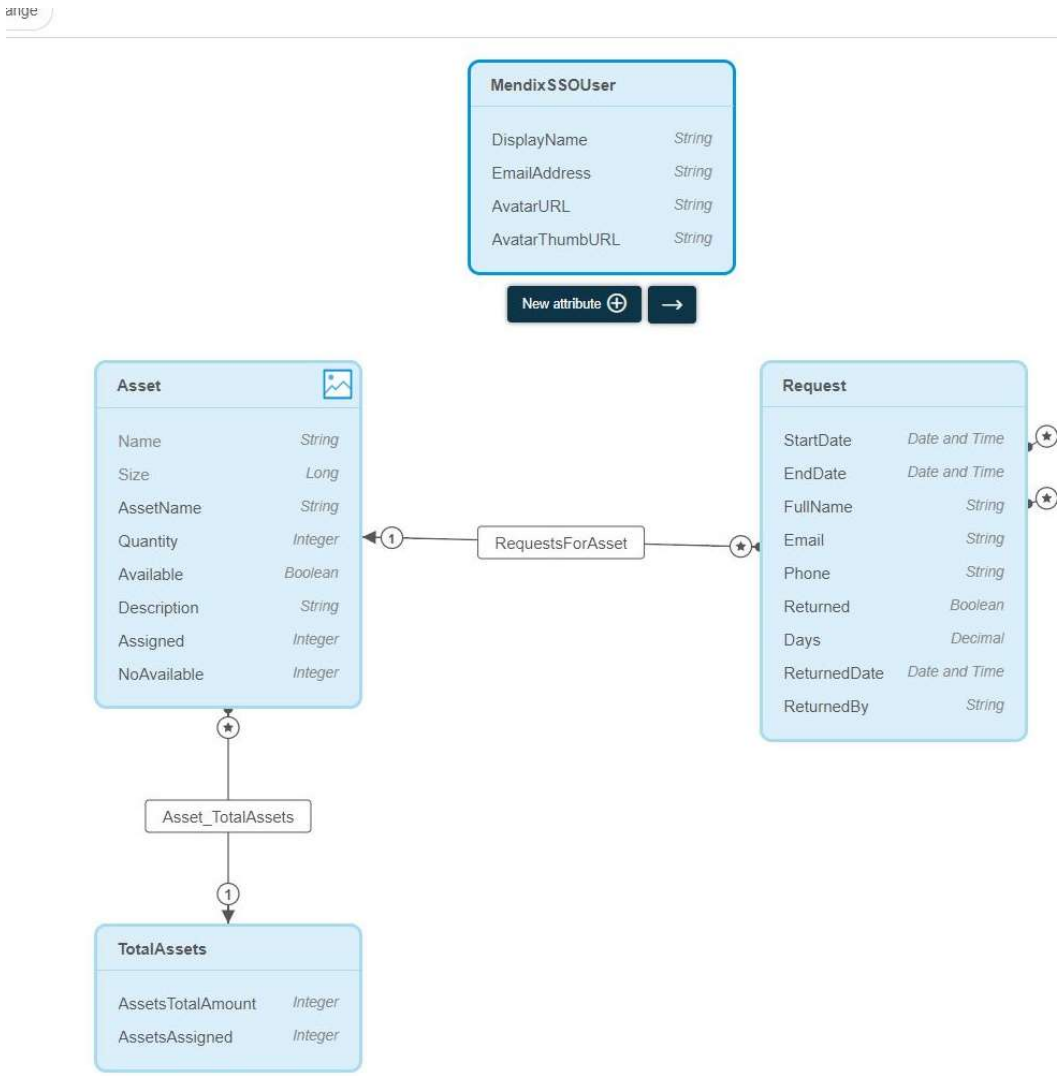
Unified Modeling Language (UML) is discussed in lecture and students are taught database fundamentals. The database fundamentals include Entity Relationship Diagrams (ERD), data

normalization and data flow diagrams (DFD). Once the students are taught database fundamentals, each team uses the Mendix platform to develop the application.

How Students Use Mendix to Create UML Diagrams

Within the platform, the database structure is created using UML diagrams. The program translates the UML diagram into database tables and relationships. A student has two development environment choices – No Code or Low-Code. Mendix tools are called Studio for No Code and Studio Pro for Low-code development. With Mendix Studio, the students can open up the development environment in any browser they use and start building the UML diagram. They define the entities, attributes and relationships as seen in the figure below.

FIGURE 4
EXAMPLE OF A UML DIAGRAM CREATED WITH THE MENDIX PLATFORM



The students simply drag and drop an entity into the browser page and then they can easily add attributes and relationship. Once the application is deployed into the sandbox environment, the entities are turned into tables, with attributes as columns and the relationships store the primary keys of the entities. Students create relationships based on cardinality such as 1-1, 1-* or *-*. They have the flexibility to

implement any and all of those relationships within the platform. The Low-code development environment called Studio Pro requires for the students to download and install the IDE (integrated development environment). The experience is still drag and dropping of entities and creating the attributes and relationships. However, students have even more flexibility and options when creating those entities.

USER EXPERIENCE AND USER INTERFACE

Any solution build needs to have a user interface and provide a smooth user experience. The user experience has to delight and provide users with a frictionless process in order to accomplish what the users need. The user interface is how the pages that the users see look and feel. For example, image the Uber app has neon green colors and the rider needs to click 5 buttons and scroll to the end of the page to order their Uber. This would not result in a good experience and uber riders would stop using the platform. Thus user experience and the design of the user interface is essential to building a great solution.

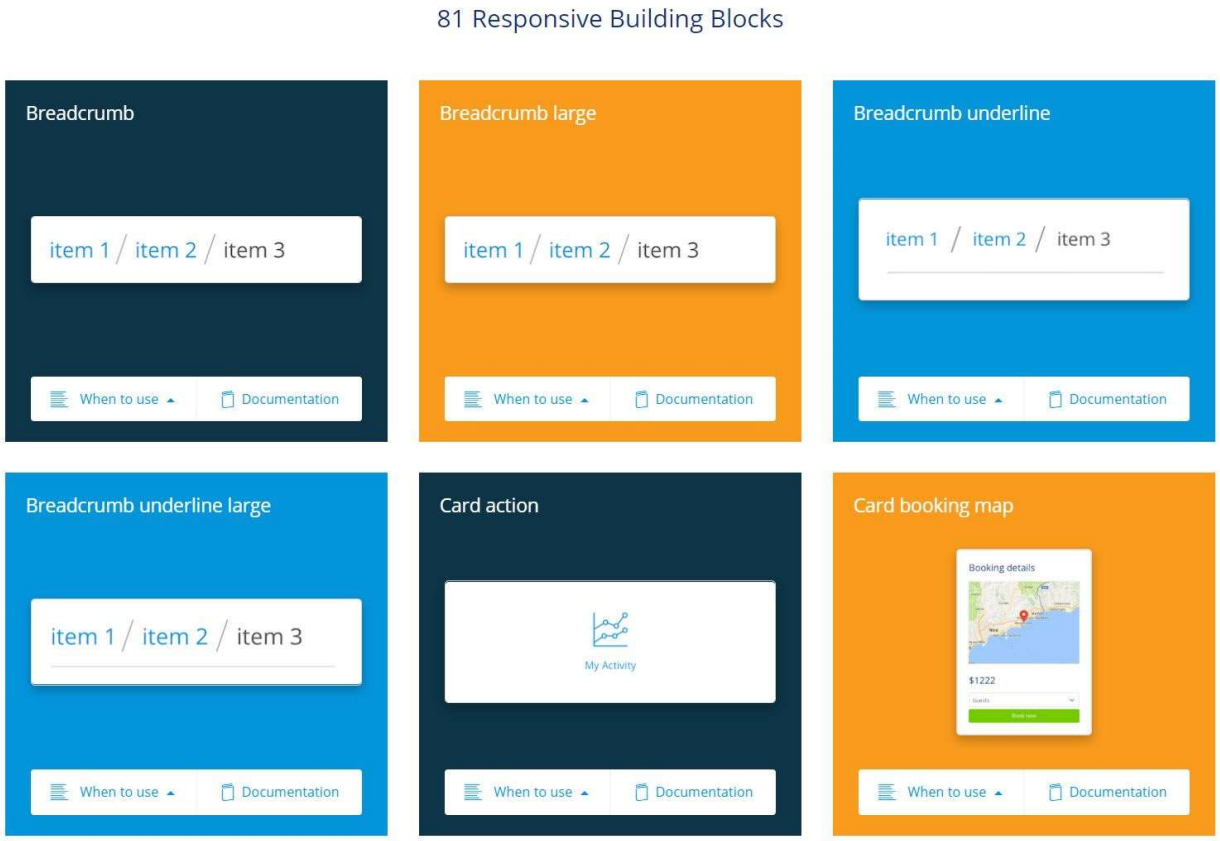
Classroom Methodology

The user interface experience starts during the design thinking workshop. During the first classes, students form teams made of 4-5 members depending on the size of the course. As a group, students identify or are assigned problems within the university or implement a business idea the student team has identified as an opportunity. The students interview respondents that are involved in the challenge. After gaining empathy from respondents, the students develop a user interface using the Mendix application platform. The Mendix application allows students to develop user interface without coding. Students develop a first draft of the user interface. Once the first iteration is developed, students give respondents an opportunity to view the initial draft. Students receive feedback from respondents and present the interface to the course instructor for additional feedback. The team conduct additional changes to the User Interface after the professor provides feedback and discusses user interface expectations. Expectations for the user interface include logos, functionality on mobile devices and data integrity (Shelly and Rosenblatt, 2012). Feedback of the user interface continues throughout the course until project completion.
Each student completes a series of exercises and completes an exam. Once the student completes the exam, they are certified to use the Mendix application suite. Users provide feedback each week of their experience within the application and with their team. Pending experience of the team, members can be removed if there isn't satisfactory participation from other members.

How Students Use Mendix Build the User Interfaces and the User Experience

Typically, if someone is building a user interface or experience from scratch, they have to know HTML, CSSS, JavaScript and much more. Within the Mendix platform, the students can drag and drop pre-built "widgets" which are user interface components. These widgets can perform various tasks such as display lists of records, card actions, breadcrumbs and various user interface functionalities. The students explores using the various widgets and focus on building a user interface and experience that aligns with their requirements. Underneath the hood, the widgets are built with the standard coding languages but are abstracted away. The figure below showcases a number of standard pre-built building blocks that the students can use. It highlights what the user interface options are and how they are used in a project. For the full highlight of the widgets and their use cases, students can go to atlas.mendix.com.

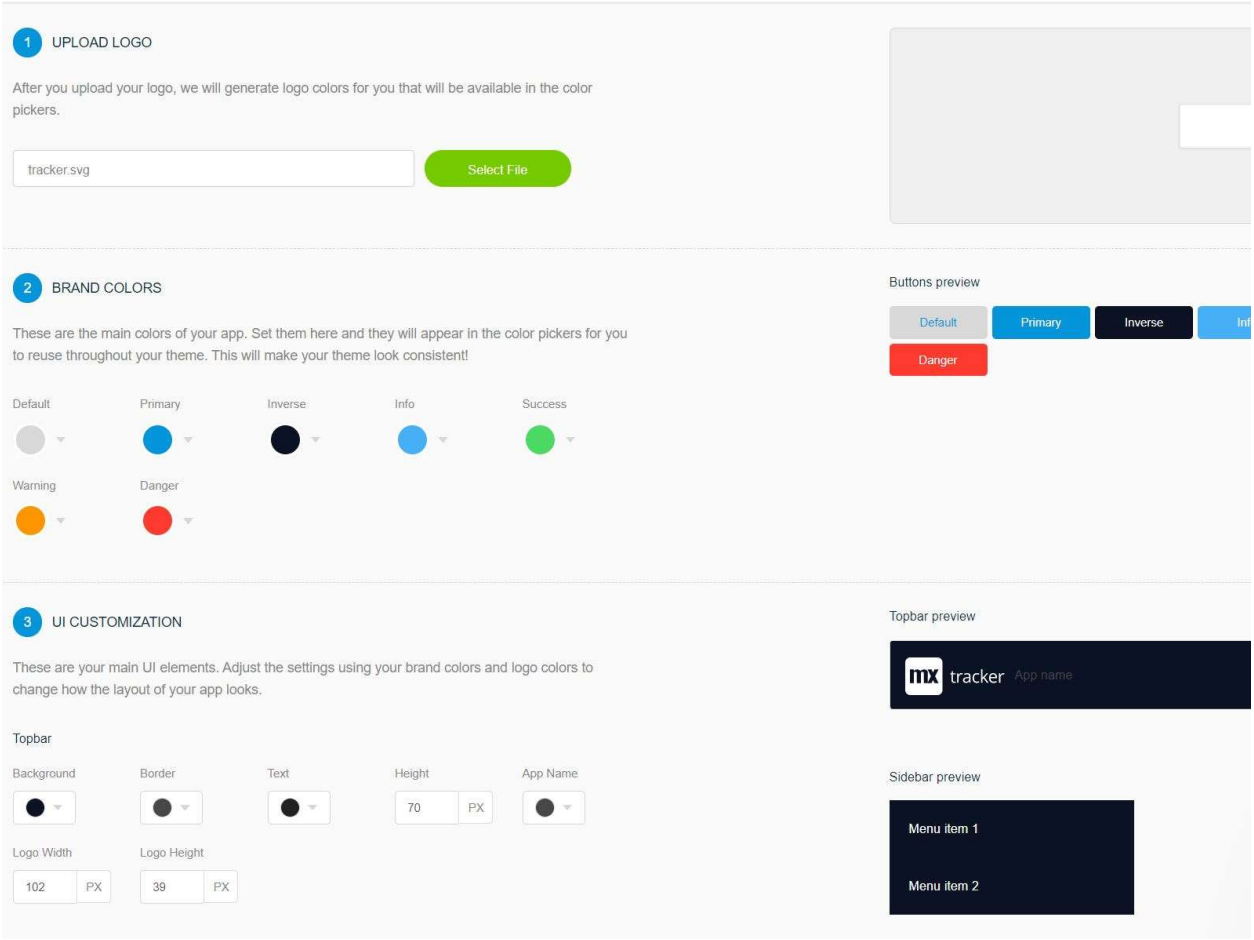
FIGURE 5
EXAMPLES OF TEMPLATES AND WIDGETS IN THE MENDIX PLATFORM



In addition, the students have the option of changing the color schema and creating a brand for their application with the click of a button. Within the Mendix platform, there is a theme customizer tool that allows students to style and design their application to their liking.

Students are encouraged to test out and use the out of the box widgets and theme styling as needed. This allows their apps to be unique and styled as needed. Depending on the number of student projects, there can be a variety of styles and applications built. The agile process allows the students to change the application's look and feel as needed. Students get to experience the evolution of their application and make changes on the fly as needed and based on feedback from the professor or stakeholders – typically a campus representative or external companies. Figure 6 below showcases what the theme customizer looks like. Underneath the hood, the colors the students pick change the CSS styling of the application, when the application is deployed.

FIGURE 6
THEME CUSTOMIZER AND OPTIONS FOR STYLING AND BRANDING THE STUDENT APPLICATION



The professor encourages the students to spend time developing the experience and the interface, because it is integral to user adoption of the solution.

BUSINESS LOGIC AND WORKFLOW

A system and solution are more than just a data structure and front end layer. It has to have logic and workflows implemented in order to manage processes, manipulate data and integrate with other systems. Thus, students need to spend time defining the processes within their system.

Classroom Methodology

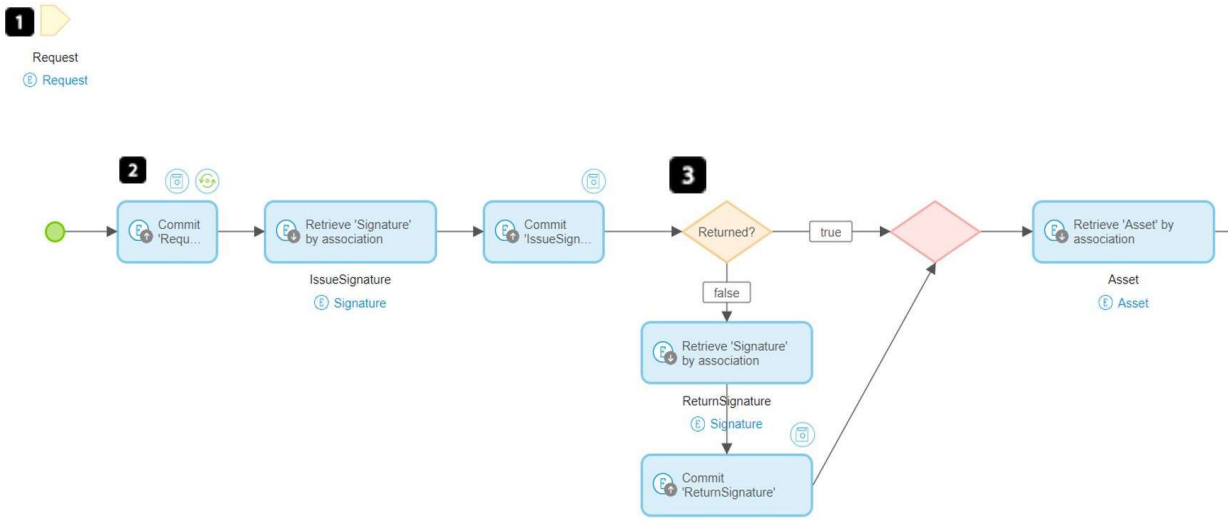
As students gain experience of the Mendix application, the instructor discusses business logic and workflow. The professor discusses business logic, data logic and workflow using the Shelly and Rosenblatt textbook. This concept is discussed throughout the beginning of the course and it is emphasized when discussing the SDLC and agile methodologies. After lectures have been completed, students are given the task to develop business logic and workflow related to their project to display understanding. Student teams present business and data logic to the professor and information technology professionals such as consultants and Mendix team members. Student teams receive feedback and submit

business logic and workflows again for feedback. Once they receive additional feedback, the logic is implemented in the Mendix application suite.

How Students Use Mendix to Build the Business Logic

Within the Mendix platform, the students can start to build microflows. Microflows are the logic layer in Mendix and allow the students to build processes and workflows as needed. Figure 7 below shows an example of a microflow. The microflows follow the Business Process Modeling Notation (BPMN) principles.

FIGURE 7
MICROFLOW EXAMPLE THAT FOLLOWS BUSINESS PROCESS MODELING NOTATION



Microflows provide a visual representation of the logic with an application. The figure above highlights an example of submitting a request for an asset. For example, you are a new employee and need to get a laptop from the company's IT department.

Microflows building blocks are:

1. Input parameters – the Request is being passed as input parameter to the microflow. Microflows can have 0 to as many input parameters as needed.
2. Activities – the blue boxes – Activities can do things like database transactions such as retrieval and commits, counting of records, changing records and so on
3. Decision – is a decision point, where you can check values and based on the answers, there is different out flows

There are additional building blocks to microflows such as loops for iterating over a list of records and so on. Students learn all about microflows as they go through the Mendix rapid developer course. Then they apply the knowledge they gain in their projects and implement various processes such as approvals, decisions workflows and so on.

USER ROLES AND SECURITY

Any system build needs to have different user roles defined. Each user role will have different access to the system. For example, in the Uber app example, there is user roles such as driver, rider and administrators. Each user role has a different function in the application. For example, the driver accepts riders and drives the rider to their location. The rider requests rides. The administrator has access to more

application functionality. The driver and rider will not be able to see the details of every user in the application. This is what makes the application secure.

Classroom Methodology

User roles are established once students complete the Mendix Rapid Application development course. Students are encouraged to identify user roles in the application based on the interviews and the discovery they do. Security is implemented during application development. Team members are encouraged to gain empathy and understanding from respondents. Once students develop the application, it is mandatory to develop security for the application. Once the security process is implemented, it is presented to the course professor for feedback. Team members receive feedback, changes are made and presented to the course professor. Finally, the team presents security features to IT professionals and respondents as a final presentation.

How Students Use Mendix to Create User Roles and Manage Their Access

Within the Mendix platform, the students must define the user roles for the application. They typically implement at least 2 user roles in the application. As they develop, they define the access that a user role should. User roles have access to pages, microflows and entity access. Entity access is defined as Create, Read, Update & Delete (CRUD). A user role can have full access to a page or no access. In addition, within a page, the students can make parts of the page conditionally visible based on the user role. On the microflow, if a user role does not have access to the microflow, they will not visibly see or be able to trigger the business logic.

Students deploy their applications in the sandbox environment, and they test the application features and functionalities based on the different user roles. As they iterate through the sprints within the semester, they update the user's functionality based on feedback and testing.

CONCLUSION

The pedagogical method used was constructivism by placing students in pair work/group work pattern and project based learning (tes.com). Pair/Group work was used in the course to allow students to learn the application development process in roles predetermined in the Mendix rapid application development process. Project based learning provides students with the opportunity to develop an application through a project based process. Students were asked to identify problems on campus using the design thinking process. After completing design thinking, student groups have the opportunity to identify a project of their interest and develop the application. If students are not familiar with a specific subject, a lecture is provided to discuss issue. Mendix provides experts to discuss specific development processes that are not fully understood. The students implement the process within the Mendix application and receive feedback from the course instructor and Mendix team.

At the end of the classroom, the students do a final presentation where they demo their application and highlight all the concepts covered above.

Overall, the experience has provided students somewhat familiar with Information Systems processes with the opportunity to research, plan, and develop an application system without programming experience. This introduces students to an environment where they can go through a process to discover problems, use an application tool to develop a solution and evaluate the solution with experienced clients and judges. Organizations have requested education institutions to develop "market ready" candidates who are able to solve problems and provide solutions. The projects within the Systems Analysis and Design course allow students to engage in identifying problems, discovering potential solutions and developing applications.

These are the 3 key takeaways:

1. Students need real hands experience with problem solving and building solutions.
2. Students should be exposed and use the latest technology tools in order to enhance their learning and prepare them for the job market.

3. Students get the opportunity to get certified for free in one of the latest technologies and explore a job market in a field they did not know or have exposure to before.

At the final presentation, the students showcase their solutions and the challenges that they faced. The challenges that they faced are typical to what they will experience in an organization. Thus, the class gives them hands-on learning and real world experience. It is satisfying to see a full solution after the students spend a full semester engaged in solving a business problem.

In the previous classes, the students had to submit a very large system analysis document on the problem they were assigned. They did not get to experience building and solving the actual problem. Thus, adding Mendix as a tool, has been a great improvement in the students' experience in building a solution versus simply documenting a solution.

ACKNOWLEDGEMENTS

The authors would like to express thanks to the entire Mendix team as well as the Tuskegee University students who actively participated in these exercises. We hope that this paper provides students with emerging skill sets.

REFERENCES

Abdalhamid, S., & Mishra, A. (2017). Factors in Agile Methods Adoption. *TEM Journal*, *6*(2), 416-421. ISSN 2217-8309.

Cerpa, N., & Verner, J. (2009). Why Did Your Project Fail? *Communications of the ACM*. DOI: 10.1145/1610252.1610286

Cerpa, N., & Verner, J.M. (2009). Why Did Your Project Fail? *Communications of the ACM*, *52*(12). DOI: 10.1145/1610252.1610286

Frydenberg, M., Yates, D.J., & Kukesh, J.S. (2017). Sprint, then Fly: Teaching Agile Methodologies with Paper Airplanes. *2017 Proceedings of the EDSIG Conference*. Austin, Texas.

Hoag, A.M., Jayakar, K.P., & Erickson, K. (2003). The Role of Trust in Virtual and Interpersonal Environments: Implications for Team Learning and Case Method Pedagogies. *Journalism and Mass Communication Educator*.

Knupffer, W., & Field, D. (2019). *Introduce low code tools to your curriculum*. TREO (Technology, Research, Education, Opinion) Talks at 40th International Conference on Information Systems, Munich, Germany.

Litman, M., & Field, D. (2018). Mendix as a solution for present gaps in Computer Programming in Higher Education. *Twenty-Fourth Americas Conference on Information Systems*. New Orleans.

Mendix Tech BV 1. (2020). *Mendix Rapid Developer Course*. Retrieved from https://learn.mendix.com/

Mendix Tech BV 2. (2020). *Agile e-Guide*. Retrieved from https://www.mendix.com/agile-guide/

Mendix Tech BV 3. (2020). *Atlas user experience and interface*. Retrieved from https://atlas.mendix.com/

Mew, L., & Field, D. (2018). The Course Registration App: A Low Code Development Scenario to Support Core IS Courses. *2018 Proceedings of the EDSIG Conference*. Norfolk, Virginia.

Mew, L., & Field, D. (2018). A Case Study on Using the Mendix Low Code Platform to Support a Project Management Course. *2018 Proceedings of the EDSIG Conference*. Norfolk, Virginia. ISSN: 2473-3857 v4 n4721

Royse, D.D. (2001). *Teaching Tips for College and University Instructors: A Practical Guide*. Allyn and Bacon, The University of Michigan. ISBN: 0205298397

Shelly, G.B., & Rosenblatt, H.J. (2012). *Systems Analysis and Design* (9th edition). Boston, MA: Cengage Technology, Cengage Learning.

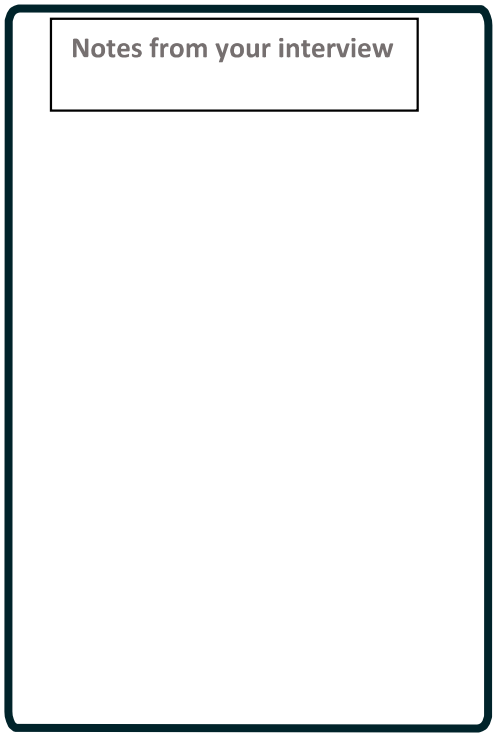
Vincent, P., Kimihiko, I., Wong, J., & Natis, M.D.Y. (2019). *Gartner, Magic Quadrant for Low Enterprise Code Application Platforms*.

APPENDIX

Your mission: Redesign the gift-giving experience . . . for your partner. Start by gaining empathy.

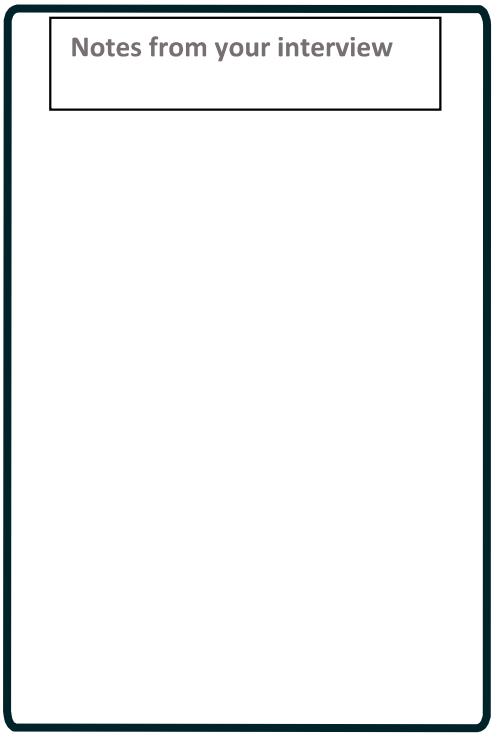
1. Interview

   8 minutes (2 sessions x 4 minutes each)

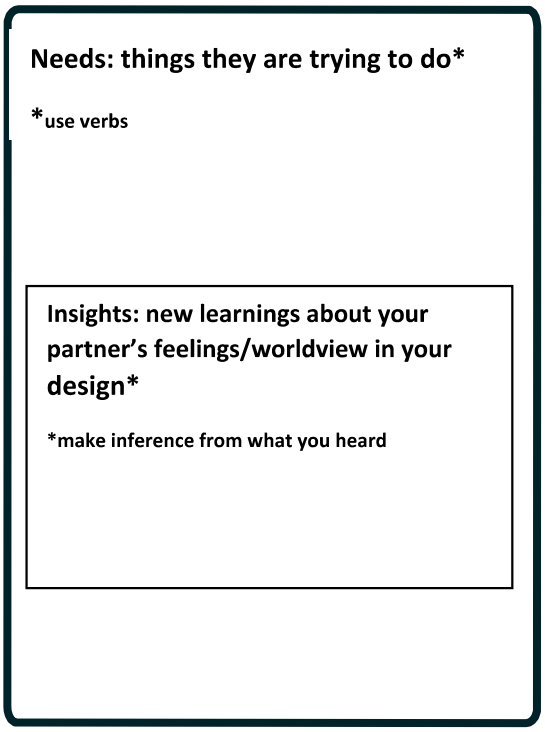


2. Dig Deeper

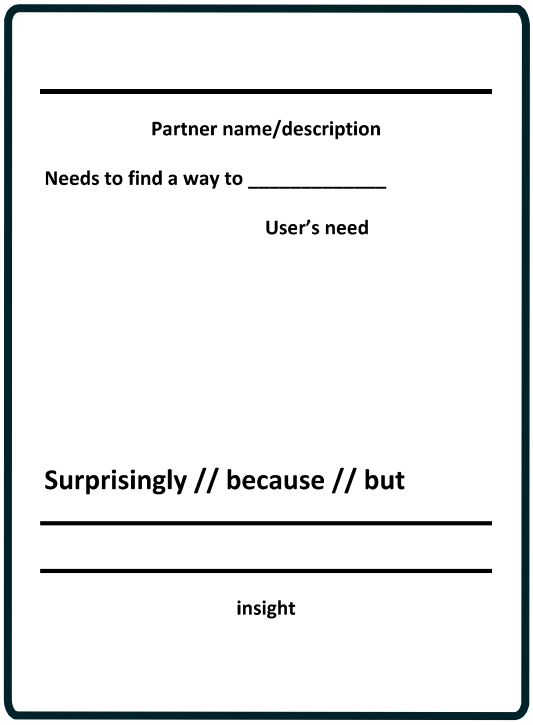
   8 minutes (2 sessions x 4 minutes each)



3. Capture Findings 3 min

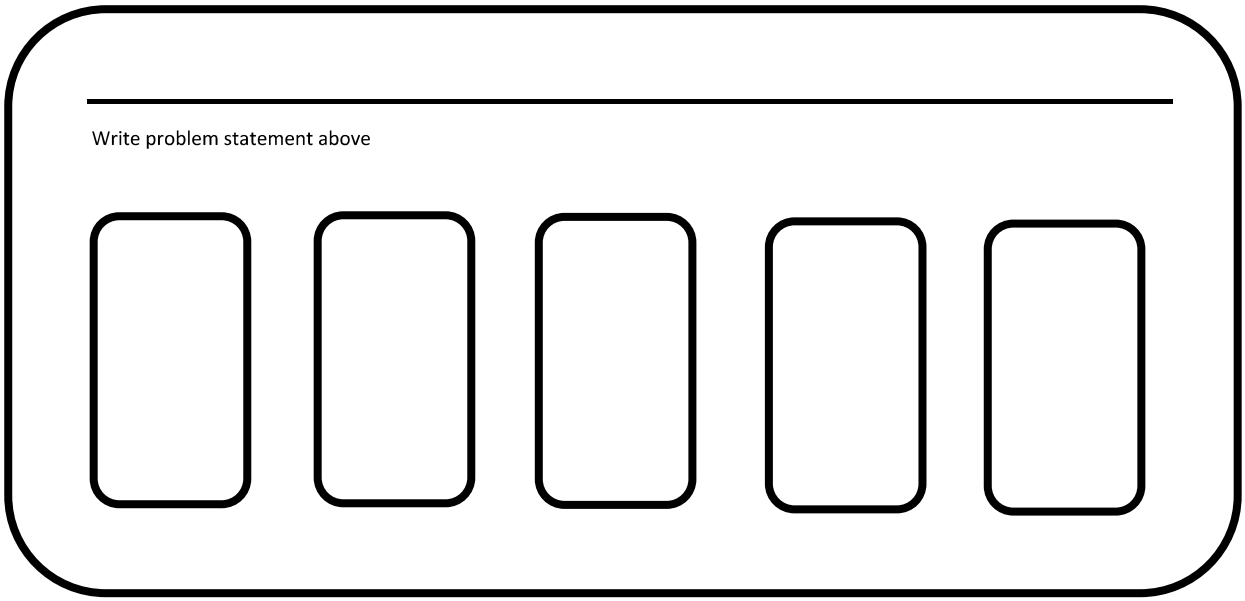


4. Define problem statement 3 min



Ideate: generate alternatives to test.

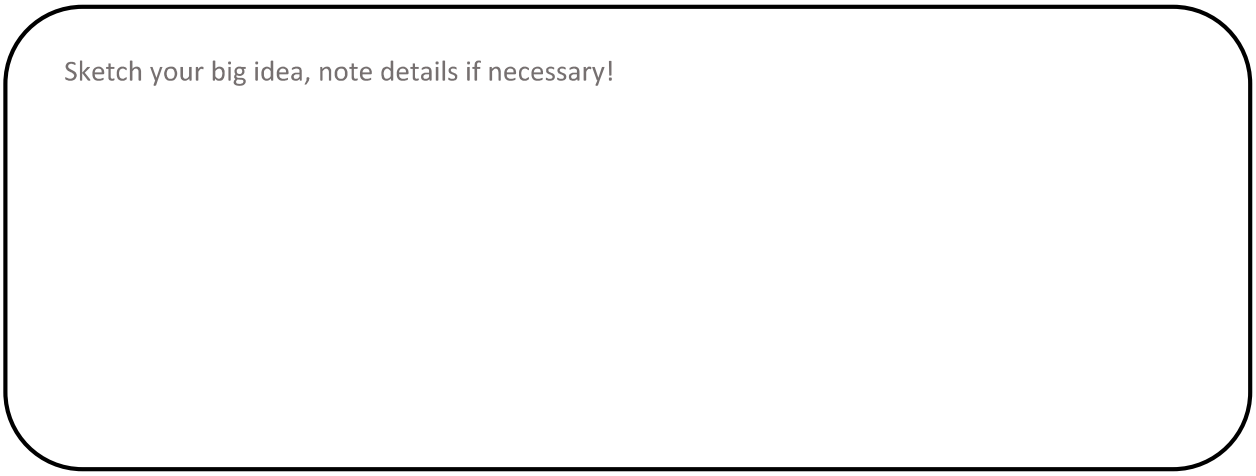
5. Sketch at least 5 radical ways to meet your user's needs. 4 min



6. Share your solutions & capture feedback. 8 min (2 sessions x 4 minutes each)

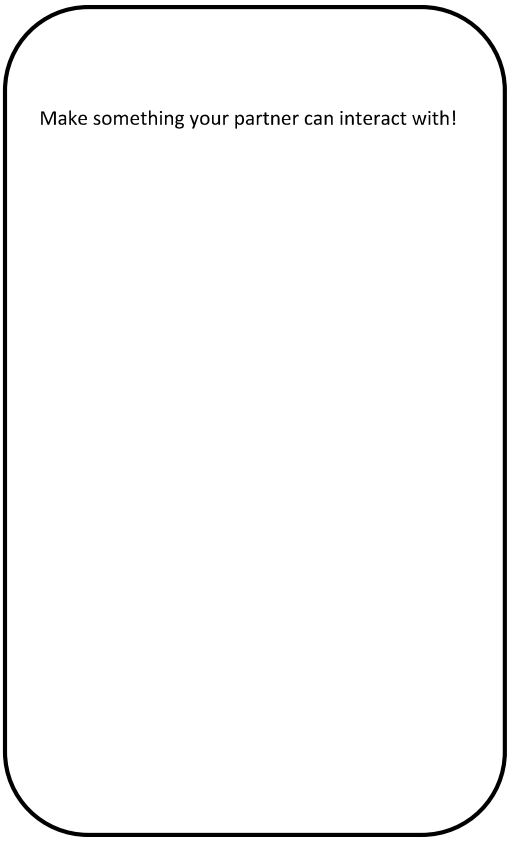
Iterate based on feedback.

7. Reflect & generate a new solution. 3min



Build and test.

8. Build your solution.



9. Share your solution and get feedback.