WIP: Integrating the Entrepreneurial Mindset into a Software Requirements Course

Dr. Walter W Schilling Jr., Milwaukee School of Engineering

Walter Schilling is a Professor in the Software Engineering program at the Milwaukee School of Engineering in Milwaukee, Wisconsin. He received his B.S.E.E. from Ohio Northern University and M.S. and Ph.D. from the University of Toledo. He worked for Ford Motor Company and Visteon as an Embedded Software Engineer for several years prior to returning for doctoral work. He has spent time at NASA Glenn Research Center in Cleveland, Ohio, and consulted for multiple embedded systems companies in the Midwest. In addition to one U.S. patent, Schilling has numerous publications in refereed international conferences and other journals. He received the Ohio Space Grant Consortium Doctoral Fellowship and has received awards from the IEEE Southeastern Michigan and IEEE Toledo Sections. He is a member of IEEE, IEEE Computer Society and ASEE. At MSOE, he coordinates courses in software verification, real time systems, operating systems, and cybersecurity topics.

WIP: Integrating the Entrepreneurial Mindset into a Software Requirements Course through Project Based Learning

Abstract

One of the most challenging aspects of software engineering is teaching students requirements elicitation. Software requirements elicitation requires complex thinking and a thorough understanding of the customer and business needs. Traditionally, requirements elicitation courses have focused on pure documentation of requirements, the focus being on drafting unambiguous statements properly formatted to follow an IEEE standard. However, the challenge of requirements elicitation is often not in the documentation of requirements, but rather in understanding the needs of a customer.

This work in progress paper intended to provide a case study of a novel approach to integrating the entrepreneurial mindset into a software requirements course. Working in teams, students are given an extensive scenario related to a real-world medical issue introduced by a brief video. Through the remainder of the course, students interview other students, real world practitioners, and others to understand the value of the product and the needs of potential clients before drafting a final requirements document which then could be used to develop the project. Through this approach, engineers communicate with nurses, athletes, pharmacists, and other non-engineers, learning the skills of teamwork, the perspectives of non-engineers, the limitations of technology, and in some cases, learn that a project that seems advantageous may actually not be successful. The paper will describe the project, the materials created for the project, and provide student observations on the success of this approach.

Introduction

> *"The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later."* [1]

Teaching software requirements represents a fundamental aspect of any software engineering program. Students enrolled in a software engineering program are generally very savvy from a technical standpoint and have a good ability to solve problems. However, these very traits often lead to problems when building large scale software systems. As has been reported by Felder [2], most engineering students tend to be introverted and sensing in their nature. This makes it difficult for students to interact with others and can lead to problems when building complex software systems. Software engineers often fail to understand the human aspects of the systems they design, yielding non-optimum results when finished.

We know from the research that appropriate requirements gathering, and elicitation are critical to the success of software engineering projects. Efforts in effectively defining requirements pay off in both faster deliveries and more successful projects [3]. Thus, it is essential that a software engineering program properly teach requirements engineering. However, even within the discipline of software engineering there are many activities that can be emphasized within requirements engineering. Figure 1 shows one model of the requirements engineering process. The model indicates that the process begins with the elicitation and gathering of requirements from customers. During analysis, these requirements are checked for accuracy and conflict, categorized, and assigned an appropriate priority. This leads into definition, where preliminary documentation of the requirements occurs, prototyping, where rapid prototypes of the system are constructed, and review, where a review is held with the customer over the documented requirements. This then leads into requirements specification and final signoff.

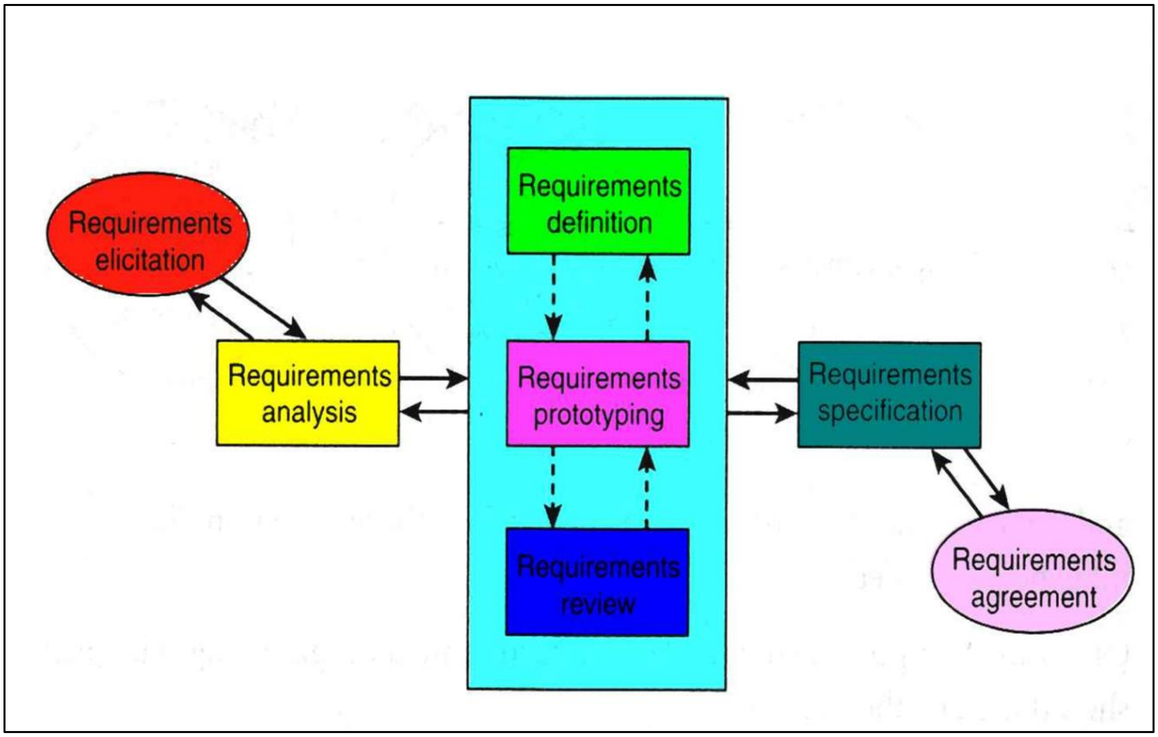


Figure 1: A Requirements Engineering Process [4]

There have been many different approaches used to teach software requirements over the years. The Software Engineering Institute (SEI) developed an initial model for teaching requirements in 1990 [5], and since then, there have been many other publications on teaching requirements. Many requirements courses focus on the formality of documenting the requirements. Garbers and Periyasamy [6] discuss their Napkins tool which is used in a software engineering course to document requirements in IEEE 8390 format. Brown [7] discusses how documenting requirements is important even for a CS1 course. Other courses focus heavily on documenting requirements and subsequent designs using the Unified Modeling Language (UML), focusing on modeling of use cases a mechanism for capturing requirements [8]. Other institutions rely on the capstone project or gaming to teach requirements management [9] [10]. Still other institutions teach requirements in the context of developing a project [11]. While beneficial, each of these methods has its own drawbacks.

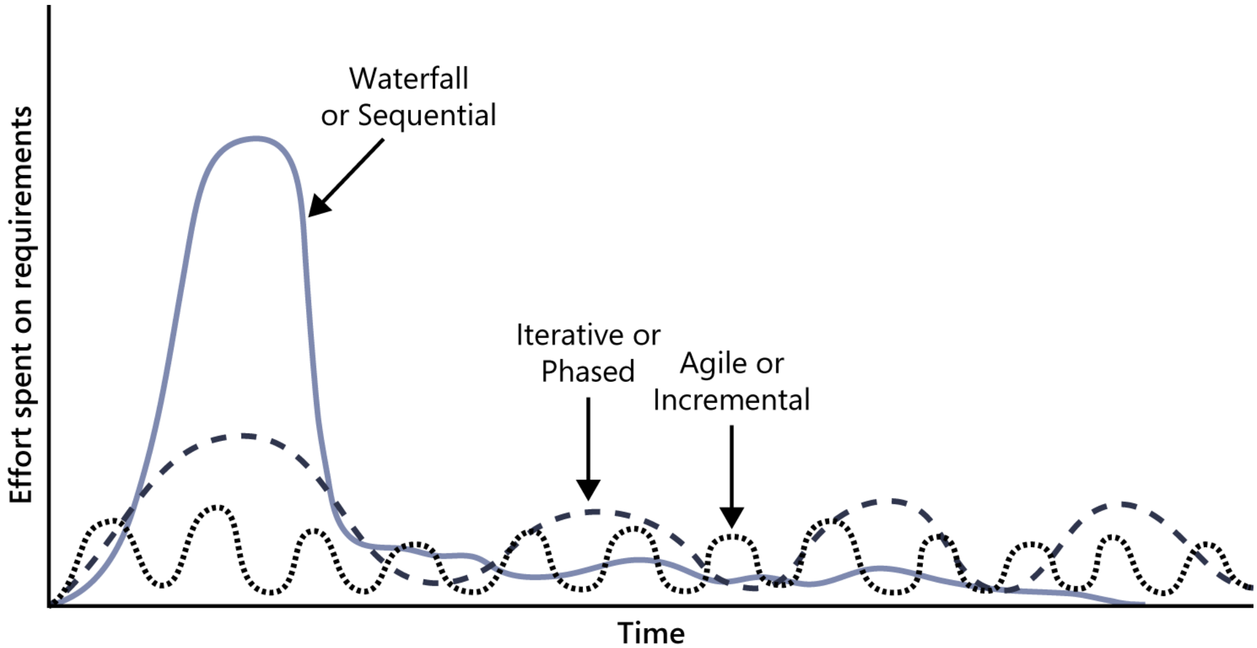


Figure 2: Requirements effort using various software development processes [1]

In industry, there also is a struggle to define appropriate requirements. Traditional projects have relied on larger amounts of time being spent upfront gathering requirements, with subsequent efforts being less as the project goes on. Agile[1] projects reject this notion, aiming to define requirements just in time as the project develops, resulting in small but reoccurring bursts of requirements activity.

The documentation of requirements tends to be different for agile projects as well. Agile processes tend to use user stores to document requirements. These short, simple statements in the form of "As a <role>, I want <goal/desire> so that <why>" [12] where the <role> is a role that a user of the system takes on, the <goal/desire> is something that the person wants to have done and the <why> explains why the goal or desire is relevant. These user stores are then combined to form an epic, which is a combination of multiple user stories required to implement a feature, and placed on the backlog, which is the work remaining to be completed on the project. As the epics are worked on, the team spends some time eliciting requirements, analyzing requirements, specifying requirements, and validating requirements while implementation

[1] In the context of this paper, Agile refers to a set of frameworks which embody the principles documented in the "Agile" manifesto. There are numerous agile frameworks currently in existence.

occurs. An Epic, by its very nature, requires multiple sprints to be completed. Each of these sprints requires close communication with the project stakeholders.

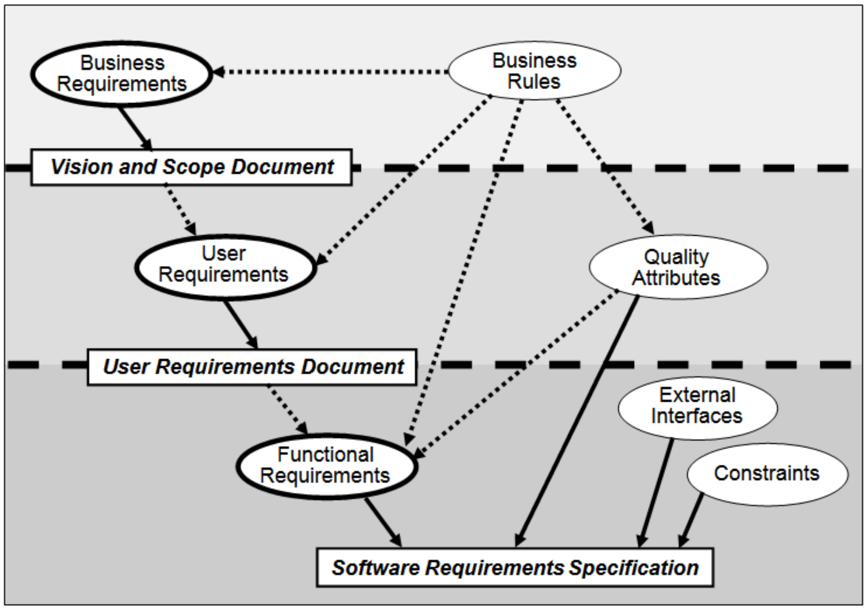


Figure 3: The Three levels of software requirements [13]

User stories, while very useful, fail to capture the full context of software requirements. In general, user stories sit in the middle of the system and focus on user requirements, not business requirements or specific functional requirements. Business requirements are necessary for the development team to truly understand why the project is occurring and what business value is put forth by completing the project. Business value should always be driving the software development, and it is important for all members of the team to understand the needs of the business. Developers though, must implement individual functional requirements.

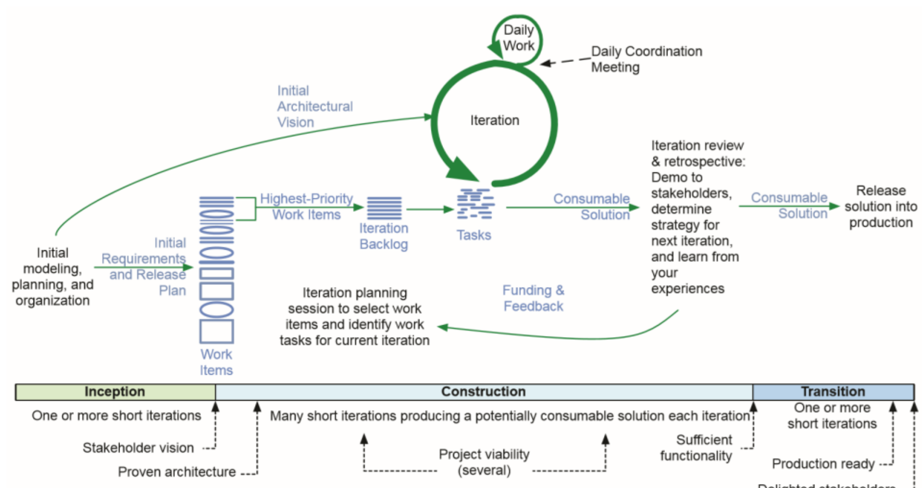


Figure 4: A Disciplined Agile Delivery Lifecycle. [14]

While this model is generally the perception for agile projects, even agile projects require upfront planning. The average agile team spends at least one month in upfront project planning and requirements modeling [15]. Without doing this, Agile does not scale effectively. The Disciplined Agile Delivery methodology adds to agile processes the Inception Phase, which provides definitions of vision, roadmaps, release planning, and other artifacts. These steps, specifically called out in older waterfall processes tend to be areas where agile projects fail.

These initial Inception Phase activities, namely planning and organization and developing stakeholder vision, closely align with the Keen Entrepreneurial Mindset. In order to goals and vision, students must be curious about the needs of the customers and whether the project will create value. Students must connect input from multiple stakeholders, weighing what may be contradictory inputs. And last, for all of these to be successful, software engineering students must learn to effectively communicate with those who have different areas of technical expertise.

Institutional Profile

The Milwaukee School of Engineering (MSOE) offers an accredited Bachelor of Science degree in software engineering and has been accredited since 2001. As an institution, there is a strong emphasis on small class sizes 13:1 student to faculty ratio) and extensive laboratory experience. Students graduating from MSOE spend on average 600 hours in laboratories related to their major. Institutionally, there is more square footage devoted to lab space than lecture hall space. All engineering students are required to complete a three-course capstone experience. While most students on campus are in the engineering fields, the school also offers a nursing program, a user experience program, and several business programs.

MSOE prides itself in having very few traditional computer labs on campus. Instead, all students enrolled in the university are issued a laptop as part of a technology package which includes the laptop and all relevant software needed for the program the student is enrolled in.

The software engineering program offers students several unique learning opportunities. One part of the program is a 10 credit Software Development Laboratory experience where students work on large-scale, industry-sponsored projects. Students are also required to take an application domain sequence of three related, specialized courses which emphasize the application of software engineering material to different domains. Most software engineering courses are offered in the 3+2 format, meaning the course meets in lecture three times for one hour and have a 2-hour associated lab period.

Teaching Requirements at The Milwaukee School of Engineering

SE 3821 - Software Requirements and Specification is a junior level course taught at MSOE to all Software Engineering students. This course leads into a 3 course lab sequence referred to as the software development lab, where students work on a large scale project in a simulated real-world environment [16]. The ABET outcomes for this course are provided in Figure 5.

Upon successful completion of this course, the student will be able to:

1. Understand the role of requirements engineering in a variety of software development models
2. Elicit requirements from system stakeholders and to overcome common obstacles to the elicitation process
3. Analyze and negotiate software requirements
4. Specify software requirements using industry standard documentation techniques (e.g.. UML, use cases etc.)
5. Specify requirements that are verifiable, traceable, measurable and testable
6. Verify that specified requirements are accurate, unambiguous, complete and consistent
7. Communicate software requirements in written documents and oral presentations

Figure 5: ABET Outcomes for SE3821 Software Requirements and Specification

The basic philosophy for the course has remained unchanged through multiple curriculum revisions, even though the specifics have varied. First and foremost, the course is structured to incorporate active learning. It is well known that students learn best by doing. Secondly, we want to use this course to expose students to an unfamiliar domain. By the junior year, students are very competent at developing software. However, they lack domain knowledge outside of the field of software engineering. Lastly, we want students to interact with those outside of software engineering.

In earlier versions of the course, students worked closely with Biomedical Engineering (BE) students to document the requirements for their senior design projects. The BE students provided all their documentation to the SE student teams, and from there, requirements were elicited from the stakeholders. This approach has been documented in several publications [17] [18] [19].

While this approach provided to be highly successful at first, problems began to develop as the course was taught multiple times. There were timing issues with the course, as the BE students had already made many of their fundamental design decisions prior to determining the requirements, which made an appropriate elicitation process impossible. Scheduling proved to be very difficult, as teams had conflicting goals and there was little chance for interaction. Ultimately, a significant change in the Biomedical engineering curriculum made this approach impossible.

New Approach

In developing a new approach to teaching requirements, the first goal was to determine what we felt was most essential for our students to learn. As a school of engineering, our students were extremely competent and savvy from a technical standpoint. However, because of the monolithic culture of campus, many of the students faced problems communicating with non-engineers. While the biomedical engineers certainly were not software engineers, they still shared many of the aspects of traditional engineering students. We also felt, that while the documentation of requirements is an essential practice for students to understand, spending significant amounts of time teaching a specific documentation format may not be beneficial. The software engineering field is currently in flux in how requirements are represented, especially given the proliferation of agile development techniques, and there are numerous tools which automatically generate requirements documents by simply setting up a template report. Most agile processes have somewhat abandoned traditional software requirements specifications in favor of user stories, a lighter weight approach which makes change easier to manage, and in many cases may be documented on note cards or in simple text fields.

Given this situation, a different approach was taken. The new approach focused greatly on the skills necessary to elicit requirements from stakeholders. These elicitation skills are the soft skills which are not often exercised in an engineering curriculum yet are often discussed as weaknesses in engineering graduates by industrial advisory committee members. The project would need to include external interviews and stakeholders.

In addition to engineering, MSOE also offers a nursing degree on campus. Nursing students represent a significantly different population from engineering students, as their mindset and personalities are very different than engineering students. This allowed for a great opportunity to collaborate, with the nursing students gaining insight into the thought process of software engineers and the engineers gaining an understanding of the nursing field.

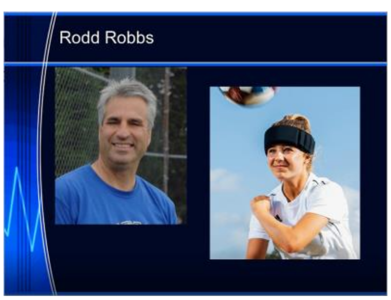
The Project Video

This led to the development of a project for the students to work on. In creating the project, a scenario was created whereby the students are tasked by an entrepreneur to determine the requirements for a glucose monitoring system. The scenario is explained to the students through a multimedia video lasting slightly more than 10 minutes. This video provides a starting point for the project, introducing the characters in the scenario, the technical domain, and what is known right now. The video uses synthesized voices, with a unique voice for each character, and these voices are used later to provide answers to student’s questions as the project goes along.

Hello, my name is Matthew Riggs. I am an entrepreneur from Cleveland, Ohio, home of the rock and roll hall of fame. I have a new startup company named bio memories incorporated. We are attempting to develop a revolutionary new product that will help to greatly improve the lives of diabetic athletes. Our product will help young athletes battle a debilitating disease without diminishing their ability to compete with their teammates. We are still early in the concept phase, trying desperately to determine what the requirements are for our system and whether it is feasible to build. That's where you come in. We need a set of experts to help us understand the real-world requirements of our product. Once we have this and an idea of cost, we can put together a proposal to the Ohio Venture Capital Fund for seed money. Coach Rod Robbs is going to give you a bit of an introduction to the problem.

Figure 6: Introductory Video and Narration Introducing the project

In this video, students are introduced to the entrepreneur, named Matthew Riggs, who is trying to solve the problem of improving the lives of diabetic athletes. Matthew wants to work with the Ohio Venture Capital Fund (which is a real-world entity) to fund his project. As he talks, numerous newspaper articles describing the challenges faced by diabetic athletes scroll by. He then introduces us to a second character, Rodd Robbs, who is a coach that is very concerned for the safety of his athletes.



My name is Rod Robbs. For the last 8 years, I've been a coach on the Fremont Flames traveling soccer team and I've also helped with the Bellevue Boilers softball team. I love coaching these sports, watching the young athletes learn the importance of sporting competition. Over the past few years, we have had a few scary moments. With soccer, there always are slips and falls, sprains, and other things happening. And with softball, I've had twisted ankles from slides and an occasional broken bone. But as a coach I take safety to be very important. Having had one of my girls get a concussion on the soccer field, I now require all my players to wear a Game Breaker Aura Soccer Headband, which is the #1 rated protective soccer headband. And to protect my pitchers and infielders, all players must wear face shields. They've proved invaluable at protecting my pitchers from batted balls. Just last spring our pitcher took a direct hit to the mouth. With the face shield, her lip was split. Without it, we would have a had trip to the emergency room.

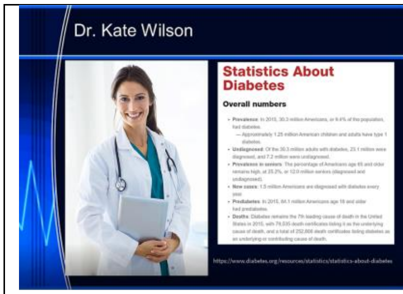
Recently, I've had a new set of problems. In the last few years, I have had more and more athletes on my team who are diabetic. They are great athletes and I'm glad they can compete. But this is a real challenge, as playing on a traveling team can be grueling and is not conducive to a regular diet or routine eating. I've been lucky. So far, I have not had anyone go into diabetic shock. We try to watch fluid intakes and food, and the teammates look out for each other.

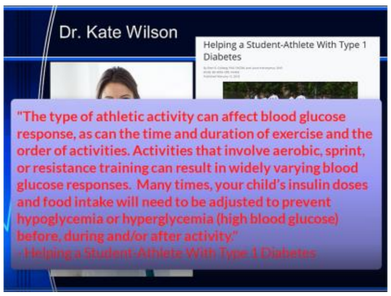
But competitive softball or soccer is strenuous physical activity, and we do not always eat well before, after, or during our tournaments. Plus, the weather is often changing. Tournaments may be in the blazing sun one day and then in fog the next. And, after the games, players often fall asleep in the vans on the long rides back home.

At a tournament a year or so ago, I met Dr. Kate Wilson. She is a pediatric endocrinologist in Cleveland, and her daughter happened to be playing against our soccer team. We won the game, but struck up a great conversation, as our opponents also had several diabetic players. What intrigued me is that she is working with researchers to use non-intrusive means to measure blood sugar, trying to detect hyperglycemia and hypoglycemia without constant needle sticks. This would be ideal, as I'd love to be able to monitor my players in real time. She's going to tell you a bit more about the science behind the project.

Figure 7: The coach explains the problem

Rodd Robbs serves to justify a need for the project, what his goals are, what his problems are, and what he hopes to accomplish. In doing this, he establishes that his goal is safety for his players, be it the soccer team he coaches or the software ball team he helps with. As his narration is occurring, He then introduces us to a third character character, Dr. Kate Wilson.





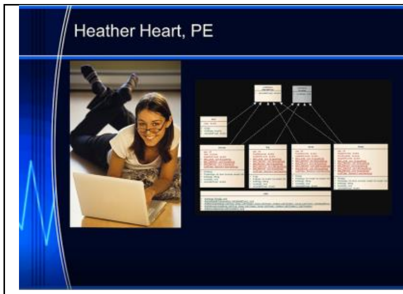
Thanks Rod. My name is Kate Wilson. I grew up in Gloucester, but moved to Ohio as a child. My degree is from The Ohio State University and I've specialized in juvenile diabetes. Diabetes is a growing problem in the united states. In 2015, thirty million Americans, or roughly ten percent of the population, had diabetes, and roughly one and a half million new cases are diagnosed each year. It is the seventh leading cause of death in the united states. Amongst youth, nearly two hundred thousand persons are diagnosed as diabetic, but this is growing very rapidly. Between two thousand and one and two thousand nine, there was a twenty one percent increase in the prevalence of diabetes amongst those under age twenty.
For young athletes, diabetes can be very problematic. The type of athletic activity can affect blood glucose response, as can the time and duration of exercise and the order of activities. Activities that involve aerobics, sprint, or resistance training can result in widely varying blood glucose responses. Many times, insulin doses and food intake will need to be adjusted to prevent hypoglycemia or hyperglycemia before, during, and or after activity. For traveling teams, parents often cannot accompany their athlete on trips, and trips may be of long duration.   The insulin pump with continuous glucose monitoring has helped many athletes to compete. However, insulin pumps do not always work well in contact sports.

I've been very interested in some work done by Marleen Olde Bekkink, a fellow at the Radboud University Medical Center in the Netherlands. Research has shown that hypoglycemia speeds one's heart rate and alters heart rate variability, which is the normal beat-to-beat variation in heartbeats. By measuring heartrate with a commercially available biosensor from VitalConnect, her team can detect the start of hypoglycemia before it was detectable with a continuous glucose meter. The data is transmitted wirelessly to a mobile device which then applies specific algorithms to detect the onset of hypoglycemia.

What I would like to do is to develop a proof of concept system for coaches and trainers. I would like for them to have the peace of mind that they are looking out for their players and are attempting to detect blood sugar problems in the young athletes they are responsible for. That's where you come. We are working with a software consultant named Heather Heart, and she is reaching out for assistance. Heather is a former MSOE Software Engineering student now working for a biomedical startup here in Cleveland. Heather will briefly explain her needs.

Figure 8: The doctor explains the domain and an area of research

Dr. Kate Wilson serves several purposes in the context of the scenario. While diabetes is extremely common in the United States, we can not assume that any given student in the class will know anything about it. As a technical expert, she provides introductory domain knowledge to the students: what is diabetes, how prevalent is it, how does it impact diabetic athletes, and the limitations of current treatments. She then introduces real world research that is going on into non-invasive detection techniques of hypoglycemia using heart rate variability. While this piece introduces a little bit of science fiction into the scenario, as HRV has not yet been fully proven as a non-invasive detection technique for hypoglycemia, it is current and state of the art research and plays a key role in the development of the scenario. She then transitions the groups to the Professional Engineer, Heather Heart, who will be leading the project.



Thanks Kate. Hey class. Please tell Dr. Taylor I said hello. I still have animal sounds set up to play when my laptop boots up thanks to freshman programming. I also need to come back for a visit. That new building looks great!

As a startup, we are looking to formalize the requirements for this software system. Overall, we know that the system will require the athlete to wear some sort of a monitor that detects Heart rate variability or HRV and connects to a mobile device. We'd like the mobile device to be individual, but we also know that there is the possibility of multiple athletes on the team having diabetes, so there may be a need to combine monitoring for multiple players. There also is the possibility of one of our athletes having undiagnosed diabetes, as it can strike at any time or age, so we'd really like all of the players to be able to wear this, not just known diabetics. The system needs to alert the coach, trainer, and potentially parents as well.

While the initial research used the HealthPatch from VitalConnect, we believe that we may be able to use other devices which record heart rate variability so long as they measure Heart Rate Variability. We don't know, however, what exactly our software will need to interface with other heart rate monitors. We know there is a website, Elite HRV, that has some information on it, but we have not tracked down the specifics yet.

From our research, we know that many things impact blood sugar. Some medicines, such as steroids, can cause tremendously high blood sugar readings. And we know that other medicines can cause hypoglycemia. Bactrim, a common antibiotic used to treat ear infections, urinary tract infections, and respiratory infections, commonly causes hypoglycemia as a side effect.

However, to accurately diagnose things, we think we may need to keep track of meals, water, and other environmental conditions. Age, stress, and sleep may also factor in.

From a software standpoint, we need to know what platform we are going to have to develop our software to run on. We need to figure out what the use cases are for this system, who the stakeholders are for this system, and then try to capture the requirements for our system. Given the medical nature of this device, we need to figure out if there are privacy laws or other government regulatory issues that we will need to overcome as well. We think there might be a need for some cloud data storage, but don't know for certain.

In doing this, we want you to elicit requirements from as many people as you can. Over the next few weeks, you will be working with experts in each of these areas to define the users meets for such a system. You will be talking with nurses, pharmacists, and diabetics. You probably want to talk to an athlete as well, to try and find out what they might want to know. This trawling for requirements will ultimately result in a better understanding of this system, both for you and for us, leading to a better product.

Overall, we do not want to develop a system that no one will use. We want something that is both technically viable and user friendly.

Figure 9: The doctor explains the domain and an area of research

Heather serves, in the context of the video, to provide initial known constraints and use cases for the project. She also introduces the teams to some of the devices that the team expects will be useful, as well as other things that may cause hypoglycemia that the students will need to investigate.

Lab Period 1

- Identify the business problem for a software problem.
- Identify the project scope for a project
- Construct a context diagram for a software problem.
- Identify software stakeholders
- Determine the project goals for a system.
- Develop Questions for the client related to the project

Lab Period 2

- Develop A use case diagram for the system you are building
- Define use cases for your system

Lab Period 3

- Plan elicitation techniques for a project
- Practice interviewing techniques to determine stakeholder requirements
- Observe competitive products at work and gleam requirements from their system

Lab Period 4/5

- Develop a pair of personas to guide your elicitation process
- Prepare to present on your project

Lab Period 6/7

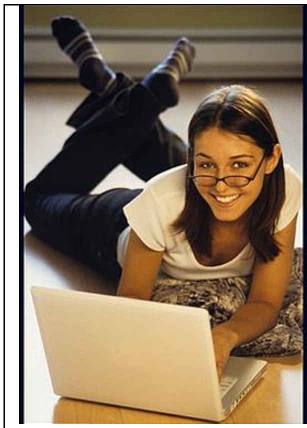
- Document functional and non-functional requirements in a meaningful fashion
- Construct fit criteria for functional and non-functional requirements
- Document rationales for software requirements
- Use snow cards to aid in the creation of a requirements document
- Create a requirements document

Figure 10: Lab Exercise Objectives

The Lab Sequence

After viewing the video, the students start down a multi-lab sequence to determine project requirements, the Lab objectives being shown in Figure 10. The first lab focuses on really identifying the initial problem, the goals, and the purposes for the project. Students are free to ask questions of the "team". However, to keep the believability of the scenario, students "ask" questions by emailing them, and the response back comes via way of an audio recording simulating a voice mail that might have been left with answers to the questions. In answering the question, the answers tend to be in the form of suggestions – suggestions for questions that the students may ask when they interview people in person about the project. An example response is given in Figure 11.

As the labs progress, students must interview potential stakeholders for the system. At a minimum, students are required to interview an athlete and a diabetic in order to gleam important aspects of the project and challenge their initial assumptions about the project. The instructor lines up interviews with nursing students in which the students are encouraged to ask about diabetes, Heart Rate Variability, diabetes treatment, and anything else which is relevant to the project. The instructor also lines up interviews with pharmacists whereby the students can discuss some of the pharmacological aspects of diabetes with a professional expert. These interviews help the students not only to better understand the problem, but also to practice communications skills.



Team Diabetes
Hey team Diabetes. This is Heather. Sorry to have to leave you this lengthy voice mail message, but with the Cleveland browns being two and two, I decided to take a trip out to san Francisco to cheer them on as they play San Francisco, so I'm going to be out of the office for a few days.

I'm going to try and answer your questions as best as I can right now. In terms of wearing the monitor, our guess is this will be like other athletic monitors. Some runners wear heart monitors and other exercise monitors mainly when exercising. We suspect the players will wear the monitor mainly during practice and games, but we could also see the players using it during workouts, bicycle trips, and other exercising. For diabetics, hypoglycemia can strike at any time, and if this system gives advanced warning, that is what we want, be it for the athlete, coach, parents, or team mates.

In terms of entering data, that is a very good question that I do not have an answer for. Personally, I think there may be some relation between recent meals and hypoglycemia setting in, maybe with some relationship to blood sugar measurement. But, it probably would be wise to ask that of a nurse or diabetic when you interview them, as I really don't know how far back things impact hypoglycemia.

In terms of measurement, right now the main focus of the system is detecting hypoglycemia before it impacts the athlete. There are other uses for Heart rate variability, but right now we want to focus on diabetes in athletes.

.
.
.
In terms of notifying emergency services, we are concerned about false positives. In the case of a game, there are likely coaches and trainers that can assess the situation and notify emergency services if necessary. But, if the athlete is running or biking by themselves, then it might be wise to contact emergency services if the athlete doesn't acknowledge the problem shortly after hypoglycemia is detected. This might be a good question for a diabetic or an athlete.
Thanks for the questions and keep up the good work!
Heather.

Figure 11: Sample response to student questions about the project from the character Heather Heart. Note that the Cleveland Browns were 2-2 at this time in the season and were playing the San Francisco 49ers that evening, tying in real world events into the fictional scenario.

Student Assessment and Feedback

To assess the effectiveness of the project and the course, students were surveyed as to their opinion of the project. Specifically, we desired data on two aspects: did the students feel that working with other majors was useful (quantitative) and did they find the project useful at learning requirements and understanding the mindset of a non-engineer(qualitative). The first question was analyzed using a 5-point Likert scale. The second from free response questions. Given that this is a work in progress, and this is the first year the course has run in this structure, it is hard to perform a deep analysis of the results. In the future, there is a definitive need for a more complete and thorough assessment of the project beyond simple surveys.

Question	**Percent Agreeing or Strongly Agreeing**	**Percent Disagreeing**
I enjoyed working with those in a different major.	78%	0%

Figure 12: Assessment results.


It was practical and required work, but never so much that it became burdensome.
Very good experience. The interaction both with teams and interviews was eye-opening.
I liked it, but it was pretty abstract.
Well planned and a good experience creating a requirements document.

Figure 13: Sample Student Written Comments to the question “What was your feeling toward the project?”

I found the lab insightful.
I did learn about the different ways to find out what their needs are.
Yes! I learned a lot about different ways to think about requirements.
Yes I did!
Yes!
Yes! It was helpful to get a non-technical perspective.
Yes, thinking from someone else’s view was super cool and important.
Yes! Lot of knowledge was gained!
Yes, much more that needs to be consider than previously thought.

Figure 14: Sample Student Written Comments to the question “From the project, did you get a better understanding of the needs of a non-engineer?”

Challenges to Teaching

While this project has been very well received by the students, there are some significant logistical challenges to this approach. During the most recent offering, there were multiple teams working on the project in parallel. While there was no evidence of academic dishonesty between teams, with only a single project, that is a significant concern. However, it also is a great Advantage, as during the project, teams were required to give oral presentations and these presentations often had very contradictory findings from stakeholders about virtually identical needs, which is a contradiction often found when talking to stakeholders for real projects.

Beyond that, however, there are issues organizing interviews with practicing professionals. For a project like this to be successful, there needs to be an understanding from the professional that this is a learning project, and the students may not always have the most advanced interviewing skills. This requires a certain type of working relationship and finding the number of working professionals in an unrelated field can be challenging.

It also is challenging to develop a realistic project for another domain. Obviously, in the field of software engineering, we are familiar with other domains, but to try and construct a realistic problem in a different domain is hard. Luckily in this case there were several professionals who volunteered their time to help develop a more realistic scenario.

The video introduction to the scenario, while not difficult to create, does make it more difficult to change the problem on an annual basis without reconstructing the introductory video. The video clearly increased the engagement of students on the project, which made it beneficial. However, it also increased the level of difficulty to revise the project in the future, as making the video is substantially more complex than a simple lab handout.

The approaches here are certainly applicable to other domains, though the specific practices may vary, as different disciplines require different elicitation techniques. These approaches could be adapted to certain interdisciplinary freshman engineering courses as well as applied to project-based learning scenarios in other majors.

At the present time, there is potential interest for students in the User Experience major at MSOE to enroll in this course, and this variety of majors may help to drive future developments for the course. However, since this course started running, MSOE has determined to make a change to the academic calendar, moving from a quarter-based model to a semester-based model. This change will result in significant changes to the curriculum, including potentially combining this requirements course with a software architecture course. In doing this, the ability to work with other majors, such as nursing or Biomedical Engineering, may be diminished, and the ability to have a core 10-week project like this may also be reduced.

References

[1] K. Weigers, Software Requirements, Redmond: Microsoft, 2013.

[2] R. M. Felder and G. N. Felder, "The Effects of Personality Type on Engineering Student Performance and Attitudes," *Journal of Engineering Education,* vol. 91, no. 1, pp. 3-17, 2002.

[3] K. Wiegers, "Jama Software.com," 12 9 2012. [Online]. Available: https://www.jamasoftware.com/blog/the-return-on-investment-from-better-requirements/. [Accessed 10 1 2020].

[4] F. Tsui and O. Karam, Essentials of Software Engineering, Sudbury: Jones and Bartlett, 2011.

[5] J. W. Brackett, "Software Requirements: SEI Curriculum Model SEI-CM-19-1.2," Software Engineering Institute, Pittsburgh, 1990.

[6] B. Garbers and K. Periyasamy, "A Light-Weight Tool for Teaching the Development and Evaluation of Requirements Documents," in *ASEE Annual Conference,* Chicago, 2006.

[7] D. Brown, "Requiring CS1 students to write requirements specifications: a rationale, implementation suggestions, and a case study," in *SIGCSE '88 Proceedings of the nineteenth SIGCSE technical symposium on Computer science education*, Atlanta, 1988.

[8] W. Hankley, "Software Engineering Emphasis for Engineering Computing Courses: An Open Letter to Engineering Educators," in *ASEE Annual Conference,* Salt Lake City, 2004.

[9] J. Preston and S. Acharya, "Using Software Engineering Concepts in Game Development - Sharing Experiences at Two Institutions," in *ASEE Annual Conference,* Atlanta, 2013.

[10] M. Johnson and Y.-H. Lu, "Teaching Software Engineering Through Competition and Collaboration," in *ASEE Annual Conference*, Chicago, 2006.

[11] J. Tuya and J. Garcia-Fanjul, "Teaching requirements analysis by means of student collaboration," in *Frontiers in Education*, San Juan, 1999.

[12] M. Cohn, "Advantages of the “As a user, I want” user story template.," Mountain Goal Software, 25 April 2008. [Online]. Available: https://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template. [Accessed 15 1 2020].

[13] K. Wiegers, "10 Requirements Traps to Avoid," The Startup, 22 11 2019. [Online]. Available: https://medium.com/swlh/10-requirements-traps-to-avoid-fb103bfeaaac. [Accessed 30 11 2019].

[14] S. Ambler, "Going Beyond Scrum: Disciplined Agile Delivery," Disciplined Agile Consortium, October, 2013.

[15] M. Lines and S. Ambler, Introduction to Disciplines Agile Delivery, Monee, IL: Disciplines Agile Consortium, 2019.

[16] M. Sebern, "The Software Development Laboratory: Incorporating Industrial Practice in an Academic Environment," in *15th Conference on Software Engineering Education and Training*, Covington, KY, 2002.

[17] D. Suri, "Introducing Requirements Engineering in an Undergraduate ENgineering Curriculum: Lessons Learned," in *ASEE Annual Conference*, Montreal, 2002.

[18] D. Suri and E. Durant, "Teaching Requirements through Interdisciplinary Projects," in *ASEE North Midwest Regional Conference*, Milwaukee, 2004.

[19] D. Suri and J. Gassert, "Gathering Project Requirements: A Collaborative and Interdisciplinary Experience," in *ASEE Annual Conference*, Portland, 2005.