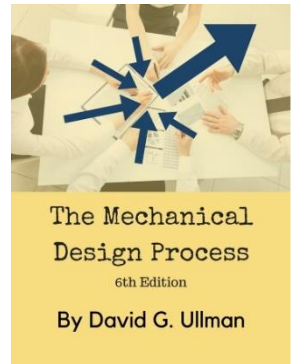
A Student Team Designs a Prosthetic Arm Using Scrum Methods



A Case Study for The Mechanical Design Process

Introduction

A team of four students at Olin College created a below-the-elbow prosthesis for amputees using the Agile/Scrum design process. This team spent eight weeks designing the “Smart Arm” complete with an intuitive feedback controller using commonly available DIY components.

The Smart Arm was designed Victoria, Liani, Celine and Ellie, students in a mechatronics class taught by Professor Aaron Hoover. He included the Scrum framework so the students could experience the process while they designed a device requiring mechanical, electronic, software and systems engineering skills. Prof. Hoover felt that the Scrum methodology well supported interdisciplinary teams and would increase their chances of design success. The students could design any product they wanted that had mechanical, electrical/ sensing and real-time control components. The eight-week project was structured so that there were four two-week sprints. At the end of the project the team had to produce a demonstratable prototype. The Smart Arm team’s result is shown with the outer covering off on the left next to human arm in Fig. 1.

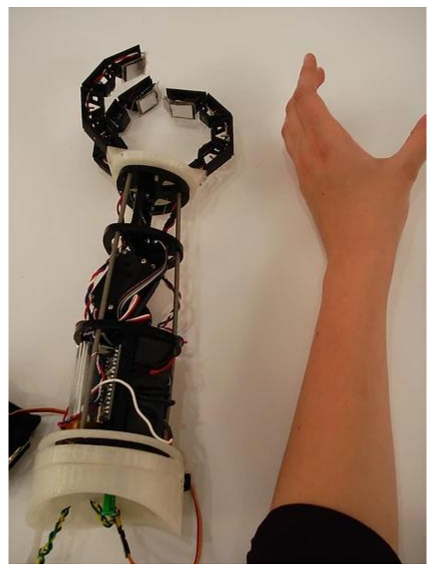


Figure 1. The partially assembled Smart Arm next to a human arm

The Problem: Learn mechatronics by developing a device that emphasizes each student’s individualized learning objectives and lets all experience an engineering design best practice.
The Method: Use the Scrum process to manage product evolution and team communication increasing the chances of success.
Advantages: Scrum allowed the team to go from zero to a working am in eight weeks enhancing mechatronic design, communication, and decision-making skills.

Background

The Smart Arm project was part of an experiential class on mechatronics taught by Prof Aaron Hoover of Olin College. The learning objectives for this course were, in Prof Hoover's words:

"At the end of this course, students will be able to:

- Work effectively as a member of a project team.
- Develop design concepts and create technical specifications that address defined needs.
- Balance trade-offs and make defensible choices among design alternatives.
- Use modern tools to construct mechatronic systems.
- Assess and select appropriate components for mechatronic circuits and systems.
- Use written, oral, and graphical communication to convey design ideas and solutions, electronic system analyses, and experimental results.
- Undertake an iterative prototyping process to improve design ideas."

To meet these objectives, early in the term he introduced the Scrum design process. His goal in using Scrum was to provide the students with a framework giving the multidisciplinary teams a high chance of design success.

After Prof. Hoover introduced mechatronic basics during the first few weeks of the term, the class self-chose four-person teams and picked their own projects that had to include:

- A non-trivial mechanical system
- A non-trivial electrical/sensing system
- Real-time control (i.e., must use a microcontroller)

The teams had eight weeks to develop their products and were to work in two-week sprints. His grading system gave 65% of the course credit for the deliverables presented during the Sprint Review at the end of each sprint. He introduced the Scrum framework with a slide presentation built around Fig. 2.

Professor Hoover leads about 20 teams a year through this class. The projects cover a wide range: e.g., games, music generators, wearables, cooking devices, bike products, visual arts, and prosthetics. The team featured in this case study consisted of Victoria,

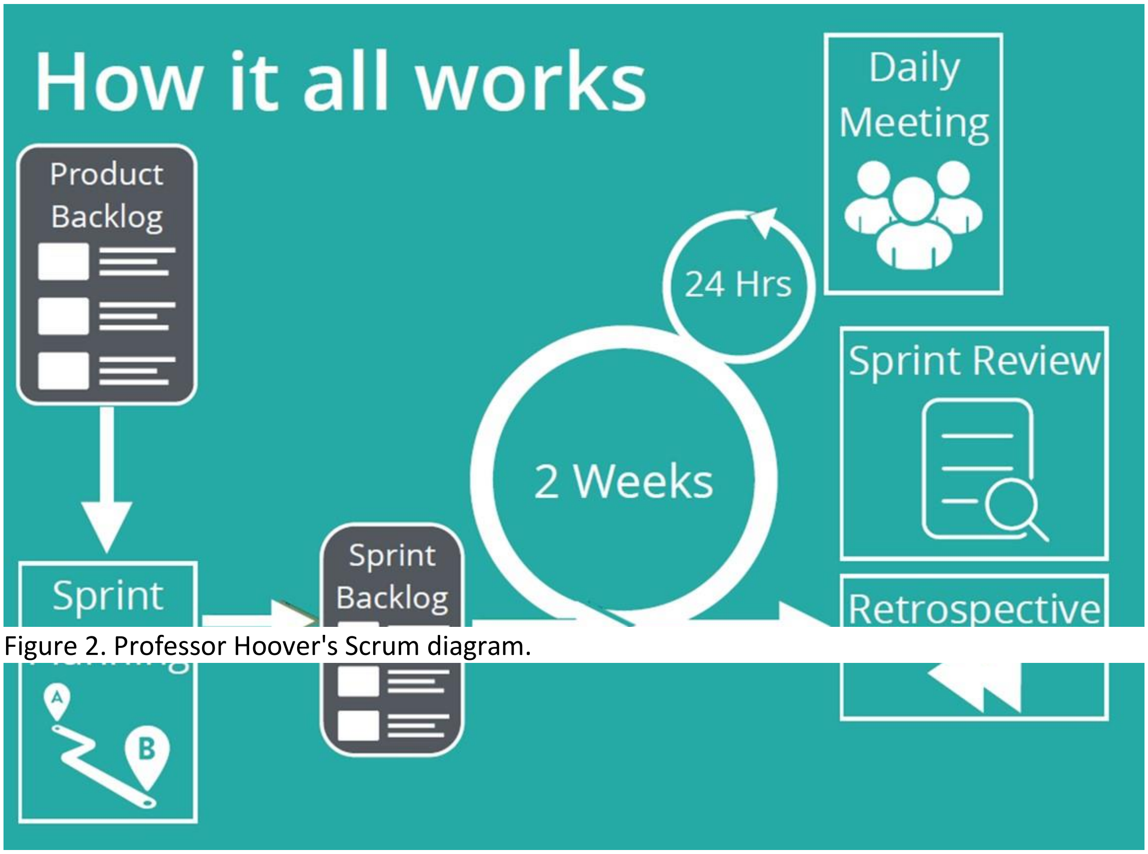


Figure 2. Professor Hoover's Scrum diagram.

who focused on the controller and software; Celine who identified and implemented the sensors; and Liani and Ellie, both mechanical engineers, who developed the physical components and their manufacture. All were sophomores except Victoria, a junior.

At the beginning of the project, they had to "pitch" their idea about what exactly to design to each other and then to Prof. Hoover. No one remembers who first proposed the Smart Arm, but all were all drawn to the idea that "hobby" technologies such as 3D printing, Arduinos, and hobby servos can be used to improve day-to-day experiences for prosthetic users. They also liked that this project would allow each of them to learn more about a skill they were eager to practice: e.g.,sensor-feedback design, simple state-machines, design for manufacture.

To guide the students, the class was formatted in 2-week sprints with a certain amount of progress expected in all subsystems (electrical, software and hardware) at the end of each sprint. Further, they were judged on how well they identified and managed risk, how they made decisions, and how clearly they articulated the next sprint's goals.

Each team was given a budget of $250 covered by Olin, but they could spend more using personal money if they wanted. The Smart Arm team spent $214.

The Tools

Besides the traditional engineering design tools such as solid modeling, spreadsheets and Arduino IDE (open-source Arduino development environment) they used Trello[1], a web-based system for managing lists. It served as their Scrum Board complete with a "Project Backlog," "To Do," "Doing," as seen in Fig. 3. Here they could post the tasks and keep track of the work. Besides the lists shown, as each sprint was completed, the finished tasks were put on a "Done" list. A sample Done list for Sprint 3 is shown in Fig. 11. The team found that using Trello for collaboration was very useful.

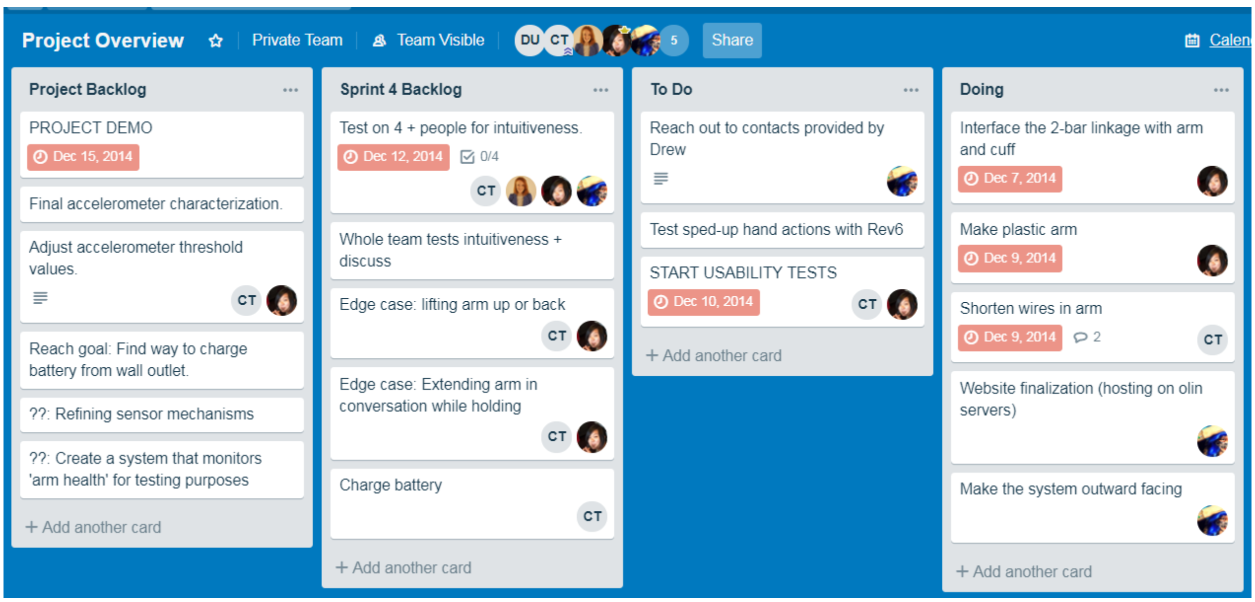


Figure 3. Partial Trello Scrum board

Additionally, they logged their progress in blog posts. See a typical example in Fig 4. Creating a project website or blog was part of the course requirements. During the project, there were 36 blog posts, most with images or videos. All the team members posted their progress, and this became their repository for product information. Where Trello was an inward facing communication tool, the blog was outward facing.

---

[1] www.trello.com

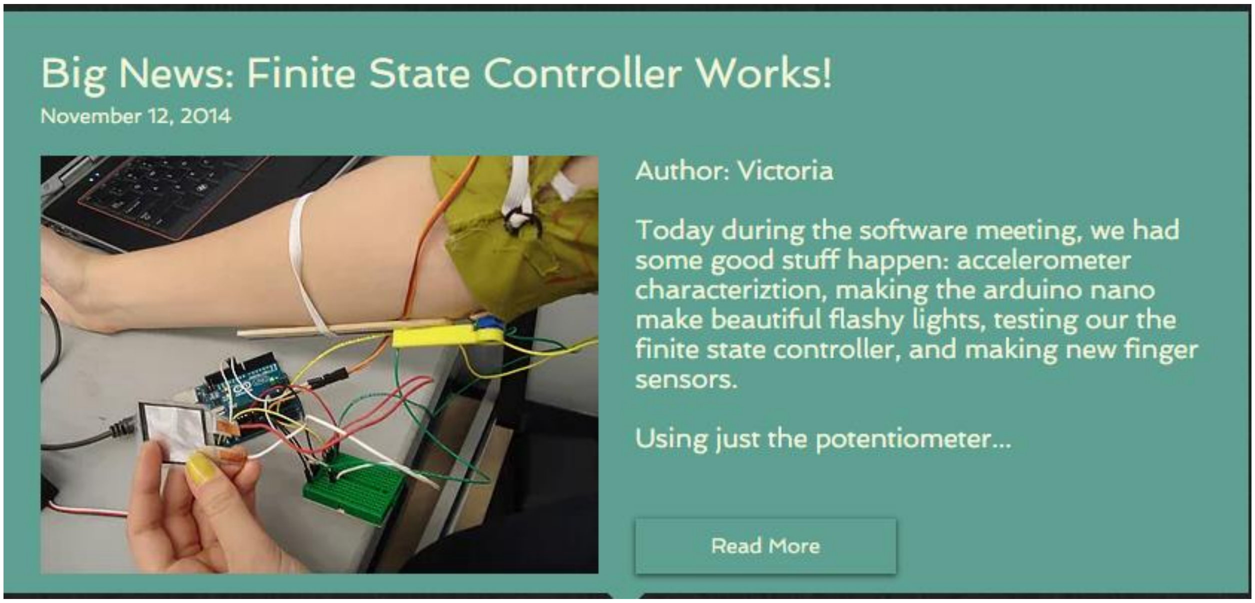


Figure 4. Sample blog post

The Process

Rather than track the project chronologically through the four sprints, examples from them will be used to explore what the team did well and where the project was lacking. This retrospective is by no means complete as the team's Trello cards, their blog, and other material captured a rich history of the project. Only examples sufficient to support the important Scrum learning objects are included.

The process followed by the team closely adhered to that shown in Fig. 2. It must be remembered in reading this material that the high-level goals for the project were to learn about mechatronics with each individual given the opportunity to learn more about their technical area of interest. The use of the Scrum framework was to both facilitate the mechatronic learning and to experience a design process best practice.

Team Organization

Usually, Scrum teams are 4-9 individuals with one person filling the role of Scrum Master – driving the process so the technical team can operate efficiently, a second person being the Product Owner – representing the voice of the customer and the remainder being the technical team. Since this team was only four people in an academic situation, the team structure was simplified. Prof. Hoover assumed the role of Product Owner since the team's sprint reviews (a formal chance to get feedback on their work) served as the graded deliverable capping each sprint. There was no Scrum Master on this team, but some teams choose to have one person fill that role while also being part of the technical team.

Further, in the ideal software Scrum team, the members have broad enough knowledge and experience to do many of the needed tasks. Here, one goal was for each student to gain knowledge and experience in their individual fields. So, the team broke into "expertise" areas, corresponding with individual learning goals. Luckily the areas of interest were independent enough that allocation of tasks was not an issue. For tasks that didn't line up with a specific area of expertise or learning objective, someone would volunteer during team meetings.

Figure 5 Two team members working together.

Additionally, as in many multidisciplinary software/hardware systems, there was little overlap in expertise, and thus tasks were often completed by the only person. That is not to say that the team did not work together. Collaborative efforts were common as in Fig 5.

Design Goal Development

Since the class was not primarily focused on making a product, the amount of domain research and goal development was somewhat limited. In retrospect, team members felt that they should have more spent more time up-front understanding the users and the issues.

The team did not write formal user stories or develop customer requirements. However, partway through the effort, they realized that they did not have a clear picture of exactly what actions the Smart Arm needed to do and developed the system state model in Fig. 6. Here the steps that the hardware and software needed to accomplish were codified. This diagram helped focus the team for the remainder of the project.

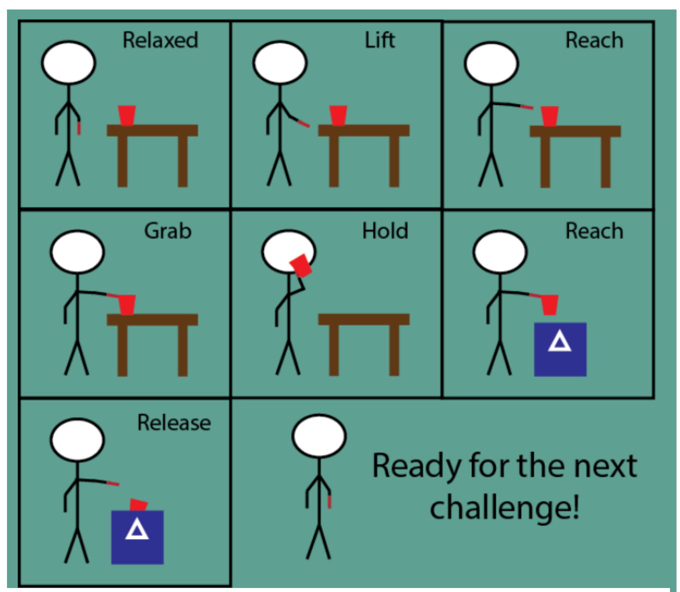


Figure 6. The system states for the Smart Arm

Product Backlog Management

The team's use of Trello (Fig. 3) made backlog management easy. As new requirements were realized they were added to the Product Backlog list. While there was no formal rank ordering in this list, the team paid attention to sequencing - those items that would delay someone else were dragged up nearer the top of the list. They were driven primarily by the Scrum cadence of a sprint review every two weeks.

Task Identification, Measures, Targets and Tests

The team did not follow any formal method to generate "tasks." The titles on the Trello cards were generally “actions” to be taken (see Figs 3 and 11) with no formally stated measures, targets, and tests.

For some of the tasks, "done" was clear. For example, "go to the store to pick up x, y, z" having an obvious conclusion. For other tasks, the deliverables were driven by an unspoken expectation for quality of finished products. Often this would require team discussion to confirm that all were on the same page about the status of a task.

There was implicit “testing” to address those more open-ended tasks. For example, a task like “verify sensor filtering” would include a test in which the sensors would have to be hooked up, run through several scenarios, and the results discussed.

Task Time Estimation

Each person "assigned" to a task would estimate the amount of time it would take. Some drew on previous experiences on similar tasks, some used timeboxing force a projected amount of time on something, and others would do a little research to see what external resources might help.

For the software tasks, since they were using an Arduino as the controller, the collective experiences of the open-source community members for similar projects helped project timeboxing.

In later sprints, time estimation became more accurate as is common with Scrum teams.

Sprint Focus

The team had long discussions over which tasks to work on during each sprint. The team balanced the product need with their specific learning goals to allocate items to each person. They were also driven by "It would be awesome if we showed this off at the sprint review."

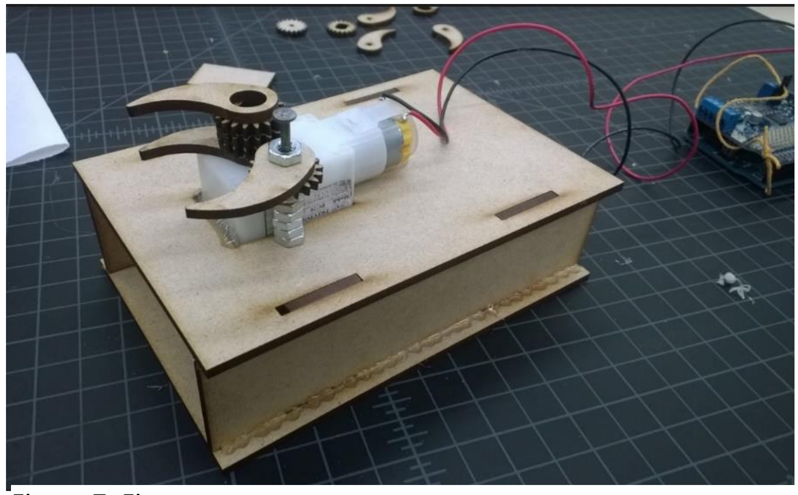


Figure 7. First prototype

During team meetings, they added to the lists in Trello and reordered the cards to help manage the sprint focus.

Risk and Uncertainty Management

This exercise was all about learning; learning about the technologies and learning about Scrum. Learning removes uncertainty, as can clearly be seen by the team's progress through their prototypes and models (Figs 7, 8, 9, and 14).

When they began, they were uncertain about how to grip objects, how to sense the gripping force and how to provide feedback to the wearer. The first model, built very early in Sprint 1, consisted of a laser-cut wooden box that was hot glued together (Fig. 7). It had simple pincers geared together as a gripper. A potentiometer was rigged to give feedback to control the gripping. It could respond to sensor input and gripped or released based on system state. It only took about 2.5 hours to build, but the base was not sturdy, and the gear drive slipped due to poor tolerance and uncertain placement. Also, the team had set a goal of being able to pick up a cup, and the fingers proved too short to accomplish this task. Despite these limitations, this simple model taught the team a lot about gripping, and sensing.

The second model (Fig. 8) was built later in the first sprint and was part of the "show and tell" for the first Sprint Review. It used the same sensors and inputs as the first model but had more sophisticated grabbing action and a bread-boarded controller circuit to manage sensing.

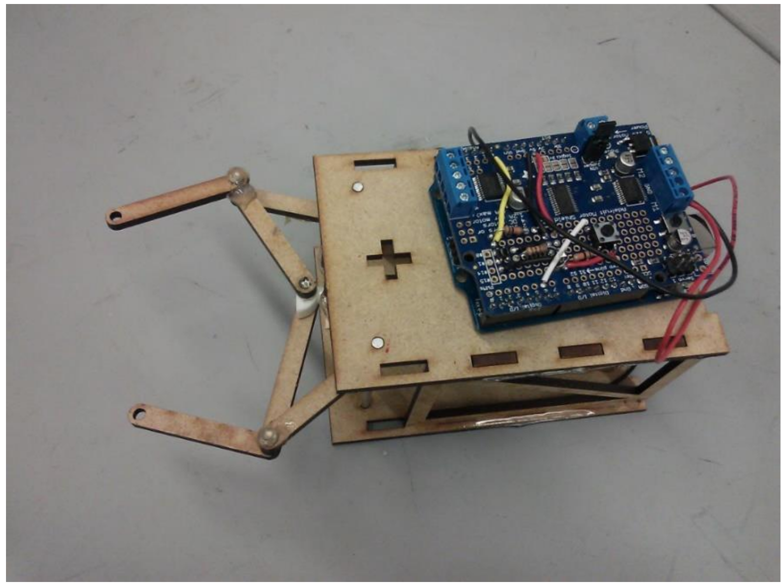


Figure 8. Second prototype in Sprint 1

As part of this the first sprint, one of the team members began to research sensing. They wanted to sense both the gripping force and how they might control it through arm and elbow movement.

The Smart Arm is a "below-the-elbow" prosthetic, which assumes that the elbow joint is intact and able to be actuated. Through sensing the elbow position or motions, the team designed the controller to "intuit" the user's intent. For example – a user extending their arm, as though to reach or grab something - could implicitly tell the hand to close. Alternatively, if the hand was already holding something, extending the arm could imply that the user would like to release.


Figure 9. Arm 1.0 built early in sprint 2.

To sense the user's intent, they began by considering piezoelectric, electrode muscle sensors, flex sensors, and stretch sensors. They then found some very inexpensive flexible customizable sensors at Adafruit, a DIY supplier[2].

Based on what was learned about gripping and sensing in Sprint 1, Sprint 2 began with work on "Arm 1.0". Made of cardboard, eyehooks, blue foam, spring wire, and thread (Fig. 9), Arm 1.0 was a great thought experiment around arm size, actuation technique, and aesthetics. After building it, they began moving into creating

[2] DIY Sensor film kit. http://www.adafruit.com/product/1917

a real-life model that integrated the servo and motor systems with 3D printed fingers, and the sensors.

Known Stable Interface Definition

One strong design feature in Scrum for hardware is to design known stable interfaces early in the process. The team identified interfaces:

- Between the human arm and the Smart Arm
- Between the physical structure and the PC board
- Between the hand and arm (Seen as the joint between the blue foam hand and cardboard arm in Fig.9.)

The team made an effort to define these interfaces early in the project and keep them fixed.

Modular Design Creation

With the interfaces determined work could progress on the modules. The arm structure (for example) rapidly progressed from the cardboard in Fig. 9, through the laser-cut wooden model in Fig. 10. Here the wooden bulkheads of the arm as connected with threaded rods. This module later was transformed into a plastic structure as seen in Figs. 1 and 14.

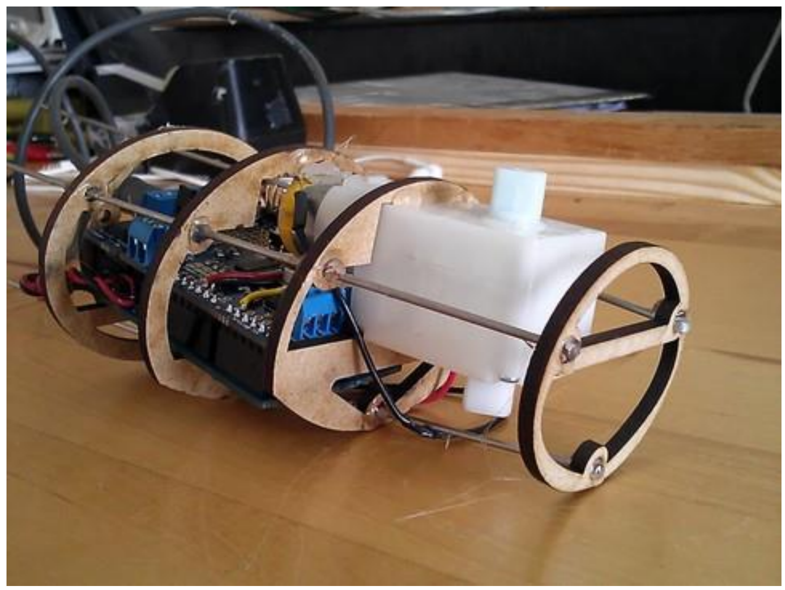
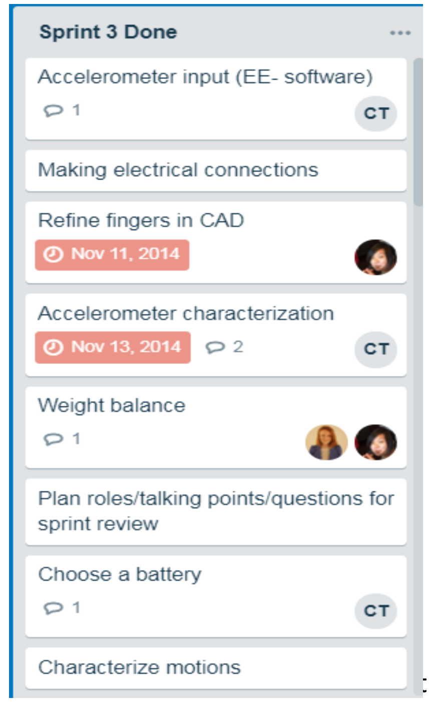


Figure 10. The laser-cut wooden arm with early PC board mounted.

Sprint Standup Meetings

The team did "stand-ups" at every all-hands meeting. These occurred in class twice a week and additionally 1-3 times a week outside of class. Team members also met as sub-teams (2-3 members at a time) 1-3 times a week. Each team member put in a minimum of 10 hrs./week outside of class time, though most weeks it was quite a bit more (20+ hrs./week/person). Seventy percent of their time was spent in groups of two or more. While each was enrolled in at least three other classes at the time, they all wished that this was their only class so that they could spend more time on the project. This feeling is common in project design classes.

The team did "stand-up" type communication at the beginning of every meeting. At the end of a meeting, they would also re-iterate everyone's task “doing” items, so they would each know what was expected of them by the next meeting. Between meetings, they used instant messaging or impromptu chats in the dining hall and dorms to keep up.



was completed in Sprint 3.

Sprint Progress Tracking

The team used Trello to track sprint progress. As seen in Fig 11, an example of "Done” tasks from Sprint 3, each of the Trello cards provided a history of work accomplished.

Sprint Review

At the end of each sprint, the team presented their deliverables as part of the Sprint Review.

Sprint reviews involved presenting progress, in the form of power point presentations or demonstrations, to the class at large. It involved discussing progress, hold-ups/sticking points, and intents for the next sprint cycle. This was an important forum to receive feedback from voices outside the team on project pacing, prioritization of various goals, and design.

Sprint Retrospective

The team used Google Sheets to manage team retrospectives (Fig 12, a sample from the first sprint). The use of a spreadsheet allowed them to do some self-reflection and posting of thoughts individually before the retrospective meeting.

They set up four columns on the spreadsheet to capture which sprint (SCRUM), what went well (PLUS), what they would change (DELTA) and what to do about it (SOLUTION). Further, they each entered their thoughts using a different color.

It is worth noting that even in Sprint 1, there was the realization that tasks were not well defined (2nd from the bottom DELTA entry).

A	B	C	D
SCRUM	PLUS	DELTA	SOLUTION
1	Whole-team understanding of system components	May have to step away from this as things get more in-depth/specialized?	Good documentation!
1	Dropbox for easy access to subsystems	I need to refresh my stuff in there more often	
1	We're all civil to each other	Have a clearer idea of what we want to explore in terms of teaming per scrum	Revisit teaming conversation, chat with Aaron about talking with us for a longer amount of time
1	People following through on deliverables	Updating when a deliverable isn't likely to be met before a deadline	Quick email updates; remember to also tell people about falling behind
1	Fast turn around time, with system integration in mind	Sharing goals for a design (software or hardware) before implementation	Delta <- Failure Mode
1		Change work environments (i.e. work in hallway on occasion)? May / may not be feasible; I find that changing enviirons helps clear my mindset sometimes	
1	Using trello to kick-off meetings	Not using trello very dynamically (maybe ok)	Make more concerted efforts to make quick checks on trello during small meetings
1	Emphasis on documentation and thoroughness	Updating in different places (PDM, Drive, Trello, Email)	Update website and jazz; blog section
	Fast prototyping	Tasks not fully described	Endgoals on trello for each [sub]system
	Just-do-it mindset	License to do things individually? vs. in subteams	Do stuff! Just inform all!

Figure 12. Sample retrospective entries

What Was Learned

The final Smart Arm is shown in Fig. 13 with one of the team members simulating its use and in Fig. 14 with its outer cover off. The resulting product is quite an accomplishment for eight weeks of work while taking other classes. The team felt that they learned a lot and were all quite happy with their results.

Figure 12. The Smart Arm in simulated motion

In terms of the course's learning objectives, Scrum directly enabled the students to:

- Work effectively as a member of a project team by scaffolding the distribution of work across team members, project time, and project objectives.
- Develop design concepts and create technical specifications that address defined needs.
- Undertake an iterative prototyping process to improve design ideas.

The Scrum process worked very well for a multidisciplinary team learning about mechatronics. Based on this success, it is clear that the framework can be used in other design experience courses. It is also clear that the structured form of meetings in Scrum greatly aided in communication both internally and with customers (i.e., the professor and classmates).

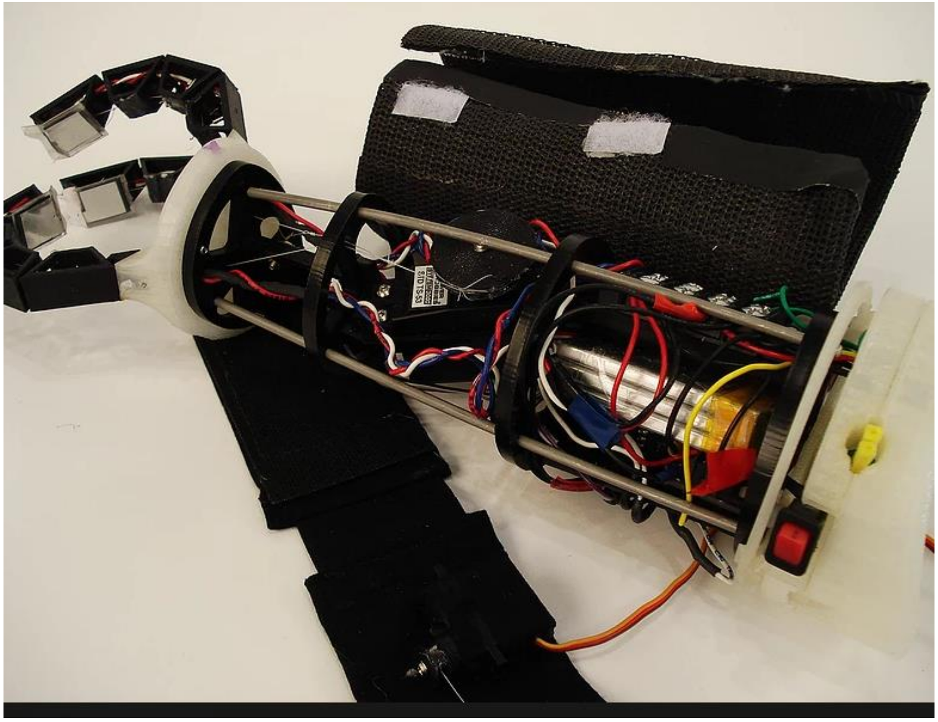


Figure 13. The final Smart Arm with the covering removed.

What was accomplished regarding the Scrum framework is summed up in Table 1.

Table 1 Summary of Scrum learning

Scrum learning objective	Result
Team Organization:	The agile/ Scrum methodology generally fostered student ownership of the product and process
Design Goal Development:	Design goals were poorly developed. In retrospect, one team member said: "I believe that spending more time in the ideation/research phase would have improved this project a lot."
Product Backlog Management:	The use of Trello worked very well for this team.
Task Identification, Measures, Targets and Tests:	The Trello cards were task-centric. However, functional targets were not set with clear measures, targets, and tests. The main driving force the desire to have something to demonstrate at the next sprint review. Prof. Hoover pushed each team for a revised definition of their "minimum viable product" at the end of every sprint based on what they have learned about their problem in the last two weeks.
Task Time Estimation:	Efforts were made to tailor task lengths to sprint schedule although this was done informally with no recording of estimates.
Sprint Focus:	Good use of Trello to manage the Sprint Backlog. A method for choosing which tasks to tackle was ad-hoc.
Risk and Uncertainty Management:	This project was all about learning, the reduction in uncertainty. While no formal methods to address risk were used, the students used the early sprints to learn, iterate and zero in on a working product.
Known Stable Interface Definition:	While not a conscious effort, stable interfaces were established between the user and the prosthesis, and between the frame and the control boards, and between the arm and the hand.
Modular Design Creation:	Not a large effort here, but the interfaces would allow multiple boards and hands very easily.
Daily Meetings:	The team did an excellent job of frequent meetings. Being in an academic situation fairly much precluded daily meetings at a set time.
Sprint Progress Tracking:	The Trello lists were well used to track sprint progress. No burndown chart was used.
Sprint Review:	The structure of the course included a sprint review every two weeks.
Sprint Retrospective:	The retrospective notes kept in Google Charts was well used.

Both Prof Hoover and his students at Olin should be proud of this effort. The students enjoyed the experience and even a couple of years removed from it thought it a great learning experience.

Acknowledgments

This case study was supported by Prof Aaron Hoover of Olin college and his students: Eleanor Funkhouser (class of 2017), Liani Lye (class of 2017), Victoria Preston (class of 2016), and Celine Ta (class of 2017). They kindly shared their blog, Trello lists, and other material. Further, they answered detailed questions about their experience, and much of the material here is paraphrased from their responses.

