

# Case study: Employing Agile Tools in Teaching Product Development to Mechatronics Students

Sondre Sanden Tørdal<sup>\*1</sup>, Andreas Klausen<sup>†1</sup> and Mette Mo Jakobsen<sup>‡1</sup>

<sup>1</sup>*Department of Engineering Sciences, University of Agder, Norway*

**ABSTRACT:** Agile tools such as Git are widely used in the industry for source control, collaboration and documentation. Such tools have been implemented in a mechatronic product development course to allow for easier collaboration between students. The course content is mainly provided using a GitLab Pages webpage which hosts software documentation and scripts. This course was first changed in 2019 to include the development of an autonomous strawberry picker. However, the use of standard learning management system and lecture slides provided a cumbersome experience for the students. Therefore, these agile tools were presented in 2020 version to improve the course. In this paper, the course content is detailed, and student feedback from both years are discussed to reveal the outcome of the changes.

## 1 INTRODUCTION

High educational quality is a fundamental necessity to achieve many of the UN's 17 sustainable development goals. To improve teaching quality and relevance, it is wise to implement tools widely used in the industry when teaching engineering courses. This is motivated both from the fact that lecturers are kept up to date with recent trends and that the students familiarize with relevant tools at the same time.

In the software industry, tools such as Git and CI/CD (Continuous Integration / Continuous Deployment) [1] are widely used for source control and automatic deployment. These tools can also be used in education for multiple purposes such as increased educational quality and collaboration among teachers and students in project-based courses. In [2–7], the authors state the advantages of using these agile tools, such as increased efficiency in teaching software development, automatic assignment grading, and efficient teaching with limited resources. In terms of a project course, the lecturers and the students can use the same agile tools to create the course material and project reports, respectively, in the shape of a webpage hosted using GitLab pages.

A product development course at the University of Agder was changed in 2019 to include more of the tools used in mechatronics software industry such as embedded Linux systems, Python programming and autonomous systems. The case has been to develop an automatic strawberry picker, and as a context, the students started up discussing this case considering the sustainable development goals. However, the standard teaching methods with slides and code uploaded to a learning management system (LMS) turned out to not be as efficient for teaching these tools. In the 2020 version of the course, agile tools are implemented to make the teaching, learning and collaboration more effective. New tools are implemented and workflow for both teacher and students have changed from the first to the second implementation. The impact of implementing the presented methods are discussed based on the midterm evaluations obtained over the two years.

The rest of the paper is organized as follows. Section 2 describes the course and the teaching setup for the two years. Then, Section 3 details in-depth the changes and how it affects the teaching. Mid-term evaluations are discussed and reflected on in Section 4. Finally, conclusions are drawn in Section 6 while possible future work is outlined in Section 5.

---

\*sondre.tordal@uia.no

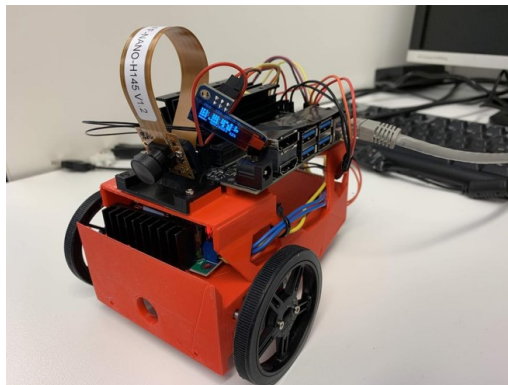
†andreas.klausen@uia.no

‡mette.mo.jakobsen@uia.no

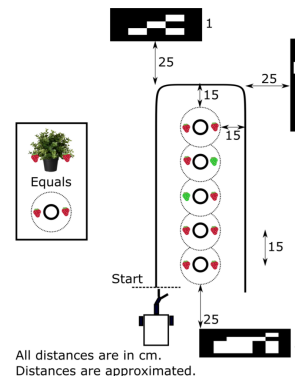
## 2 AIM OF THE COURSE

The product development course was changed in 2019 to focus on a new type of project. The goal for the students is to design a small robot that could pick strawberries. The focus should be on the product development techniques used to determine the best solutions for the robot. This course should teach the students about product development techniques, mechanical design and programming autonomous robots.

To make the project possible within a semester, the students were given some finished parts. First, the base of the robot with camera, motors, wheels, battery and Linux-based processing unit is pre-built as shown in Figure 1a.



(a) Pre-built robot (Jetbot) [8].



(b) Strawberry field layout.

Figure 1: Project equipment and task layout.

Second, the software and Python code required to operate the base functionality is already provided. This includes programming environment, motor actuation, camera usage etc. Finally, the prototype should pick the strawberries autonomously in a pre-defined field with artificial red and green strawberries hanging on artificial bushes.

The task for the students is as follows: Design a strawberry gripper and strawberry basket to attach to the base of the robot. Make a program that runs autonomously which should detect all the red strawberries, pick them up, and place them in the basket.

### 2.1 Initial course setup in 2019

Initially, the course was held using traditional methods with lecture slides and files uploaded to an LMS. However, there were many issues with this. The students were supposed to write programs based on snippets of code in the slide decks. Copying and pasting from the generated slide PDFs often resulted in misplaced characters and missing spaces for example. In addition, the files uploaded to the LMS often needed to be revised by the teacher. In that case, the students had to re-download the files and add them to their code base manually. This creates a lot of manual work for both the teacher and the students.

Additionally, the students were not familiar with Linux or Python programming, which made it harder for the students to progress outside of regular teaching hours. To program the robot and make a user interface (UI), a remote desktop application into the Linux system was used. This was at times rather slow over WiFi, and a better solution was sought after.

### 2.2 Revised course setup in 2020

In the second year, the course was revised based on the previous generated teaching material. Agile software development tools such as Git, GitLab, CI/CD and remote development were introduced in the teaching, course material creation, and to the student group workflow. The main changes from the initial setup are:

- In order to familiarize the students with how open-source projects are documented online, the course was changed to use GitLab pages [8] in combination with the Sphinx documentation system [9] to present both code snippets and small how-to guides. The main motivation for this change

was to increase the content's quality and make it more pleasant and efficient for the lecturer to change and distribute necessary code and guides among the students.

- Lectures on Git subversion control was introduced in the course. The lectures were focusing on teaching the basics of using Git, while GitLab, a cloud hosting service for Git repositories was introduced to enable the students to both collaborate using the cloud and to work as an Agile software development team.
- Remote development over secure shell (SSH) was introduced, and the UI was made using robot operating system (ROS) tools that could be viewed in a regular web browser instead of using a remote desktop solution.
- Usually the students are supposed to deliver a final project report written in  $\text{\LaTeX}$  or another documentation system. In the revised course setup, the students were instructed to use the Sphinx documentation system to produce their final report.

### **3 AGILE TEACHING USING SOFTWARE DEVELOPMENT TOOLS**

#### **3.1 Git Subversion Control**

Git is a subversion control software which is used to keep track of changes to a code base and allows for easy transition between versions of the code. It also provides opportunities to use branches for code development. For example, the main branch could contain the stable version of the software, while a development branch contains new, but not yet stable features. Once the developed features are working, they can be merged into the main branch. Git projects are often uploaded to hosting services such as GitHub and GitLab for backup and cloud based collaboration.

These agile tools provides a more streamlined integration between changes to the lecture and supporting files, and the students projects. The teacher can change some of the code used by the students, and upload the commit to the hosting service. Students can then easily check the exact changes to the code, and pull the changes to their own code base using a few git commands.

#### **3.2 CI/CD - Continuous Integration and Continuous Deployment**

CI/CD is a set of tools used to automate the required processes between software development and deployment. On change to the main code branch, the system will build the code, test the code on pre-defined use cases, and deploy the code if testing is successful. This makes the path from development to deployment very short and allows for rapidly fixing bugs and consistently testing new features with less error. The GitLab pages are created using this method, ensuring that all the students are provided with the latest course material at all times.

#### **3.3 Project Reporting using the Sphinx Documentation System**

Sphinx [9] is a Python software documentation tool which is used to rapidly make documentation for open-source projects. Sphinx has build tools for creating a full webpage with links, headers, support for code blocks, equations, citations and more. The source HTML code is compiled from reStructuredText (reST) files which are commonly used to document software projects. After the webpage is built, it should be hosted on the web. This is where GitLab Pages (and similar solutions such as GitHub pages) comes into play. These services allow for hosting the Sphinx documentation web page as well as keeping track of the version control. This completes the code documentation as the Git hosting service will allow you to keep track of the source control of code and documentation, and host the documentation in form of a web page simultaneously.

A big advantage of this form of documentation compared to WYSIWYG document editors for documentation is the ease of using source control for the files. While  $\text{\LaTeX}$  is also compiled code, it produces PDFs which can be hard to read on other platforms with small screens such as smartphones. In addition, it is easier to display for example moving GIFs in a web page, and web video players can be embedded.

One of the key features of this form of documentation is collaboration between developers. Everyone can see the latest changes and provide pull requests for proposed changes. If the maintainer sees the changes are helpful, they can be easily integrated into the documentation. Because of CI/CD, the webpage can be automatically built when a new version is pushed to the master branch.

The teacher has made everything available for students in a single GitLab repository with Sphinx documentation in GitLab Pages. In a similar manner, the students are taught how to reproduce this code documentation, and their reports will be made in a similar way. This is a result of the “show”, rather than “tell” teaching method. All the the student reports are today available as a Sphinx web-page, accompanied with a GitLab repository. The student group Sphinx webpages can be visited online at:

- <https://evenfl.gitlab.io/mas507/>
- <https://krikru.gitlab.io/mas507/>
- <https://elena1992elena.gitlab.io/mas507/>
- [https://stian\\_kristensen.gitlab.io/mas507/](https://stian_kristensen.gitlab.io/mas507/)
- <https://gurgle96.gitlab.io/mas507/>

#### 4 MID-TERM EVALUATION, STUDENT FEEDBACK AND DISCUSSIONS

The student feedback in 2019 indicates that the course could be improved in several ways. While they found the robot task quite relevant for learning the product development techniques, there were other large issues that should be addressed. The students felt the learning curve was quite steep, which is heavily introduced by the use of Python and Linux, both which has not been used in courses prior to this one. Because of the new platform, the teacher was also not properly prepared and therefore the base of the robot and software architecture was not optimal. There were several changes that had to be made during the course, which increases the workload and frustration for the students. The actual learning outcome was not properly defined, and it was therefore difficult for the students to know what part to focus on. Some of the lectures were not directly relevant to the project, which should be removed. These lectures were mainly about software, tools and theory not used by the students in the project.

While all the groups in 2019 completed the project with a fully functional strawberry picker robot, it is clear from the majority of negative feedback that parts of this course had to be changed. The 2020 version tries to fix the issues the following way:

- The Linux operating system on the robot is more hidden by introducing remote development with Microsoft Visual Studio Code and web-based UI. This way, the students do not have to interface with the user interface of an Ubuntu Linux distribution.
- The robot processing platform and the teachers familiarity with it has matured and is therefore easier to implement in a class setting.
- All the changes that had to be made during 2019 version is of course already implemented at start of 2020 version.
- Redundant lectures have been removed and the learning outcome has been better specified to focus more on product development rather than strawberry picking performance.

While these are good changes to remove some of the learning curve for the students, some other tools have been introduced which may increase workload:

- Robot Operating System (ROS) is used on the robot to get better control of the different functions and allow for better use of small separate functions that work in parallel.
- Git and version control is not familiar to the students and may increase the work load for this course. However, it is possible that the difficulty of collaboration and version control without the use of Git outweighs the difficulty and time it takes to learn Git.
- Making the documentation in form a Sphinx web page is different than the L<sup>A</sup>T<sub>E</sub>X PDFs they are used to create in previous projects.

The mid-term evaluation in 2020 compares to the 2019 version in the following ways. First, the students are more satisfied with acquiring information and files from the GitLab pages compared to the standard LSM. There were also no complaints about missing information or large changes during the course. The students did, however, point out that the use of new tools such as Git, Sphinx, Python, ROS and Linux increased the workload by a lot. They suggest to present some of these tools earlier in the education to lessen the workload in this course. The students still find the course very relevant for their education and is overall satisfied with the execution and learning outcome.

## 5 FUTURE WORK

Future work can include having the students giving feedback in terms of pull requests to the main repository, enabling the students to contribute to the course development as well. We may also investigate the possibility to implement any of the software tools earlier in the education to lessen the workload in this course.

## 6 CONCLUSIONS

This paper investigates the use of agile tools in product development class in a Mechatronics engineering setting. The case of the course is to develop the software and hardware for an autonomous strawberry picker robot. In the first iteration of the course in 2019, the students were dissatisfied with the information flow using lecture slides uploaded to the standard learning management system (LMS). Additionally, it was difficult for the students to collaborate since they were not familiar with source control software such as Git. To improve this experience, agile tools were implemented to provide teaching material on a GitLab Pages website and allow for easier collaboration between the students in the group with source control. This will also make the students more familiar with source control and software development in a way that is implemented in the industry. Feedback from the students indicate that this is a positive change, except for that fact that the workload increased because of this inclusion of more tools in this course. A way to alleviate this is to present the tools earlier in the education. Overall, the students are satisfied with the course as taught in Autumn 2020.

## REFERENCES

- [1] M. Fowler. (2006). “Continuous integration,” [Online]. Available: <https://www.martinfowler.com/articles/continuousIntegration.html> (visited on 11/17/2020).
- [2] E. Kral and P. Capek, “Towards using continuous integration tools to teach programming courses,” in *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2015, pp. 871–872. DOI: 10.1109/CSCI.2015.78.
- [3] F. Špaček, R. Sohlich, and T. Dulík, “Docker as platform for assignments evaluation,” *Procedia Engineering*, vol. 100, pp. 1665–1671, 2015, 25th DAAAM International Symposium on Intelligent Manufacturing and Automation, 2014, ISSN: 1877-7058. DOI: <https://doi.org/10.1016/j.proeng.2015.01.541>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877705815005688>.
- [4] J. C. Caiza and J. M. Del Alamo, “Programming assignments automatic grading: Review of tools and implementations,” in *7th International Technology, Education and Development Conference (INTED2013)*, 2013, p. 5691.
- [5] M. M. Muller and W. F. Tichy, “Case study: Extreme programming in a university environment,” in *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*, IEEE, 2001, pp. 537–544.
- [6] J. G. Süß and W. Billingsley, “Using continuous integration of code and content to teach software engineering with limited resources,” in *2012 34th International Conference on Software Engineering (ICSE)*, IEEE, 2012, pp. 1175–1184.
- [7] M. Goodwin and T. Drange, “Teaching programming to large student groups through test driven development,” in *Proceedings of the 8th International Conference on Computer Supported Education*, ser. CSEDU 2016, Rome, Italy: SCITEPRESS - Science and Technology Publications, Lda, 2016, pp. 281–288, ISBN: 9789897581793. DOI: 10.5220/0005789502810288.
- [8] S. S. Tørdal and A. Klausen. (2020). “MAS507 - GitLab Pages homepage,” [Online]. Available: <https://uia-mekatronikk.gitlab.io/mas507/index.html> (visited on 12/27/2020).
- [9] G. Brandl. (2020). “SPHINX Python Documentation Generator,” [Online]. Available: <https://www.sphinx-doc.org/> (visited on 11/17/2020).