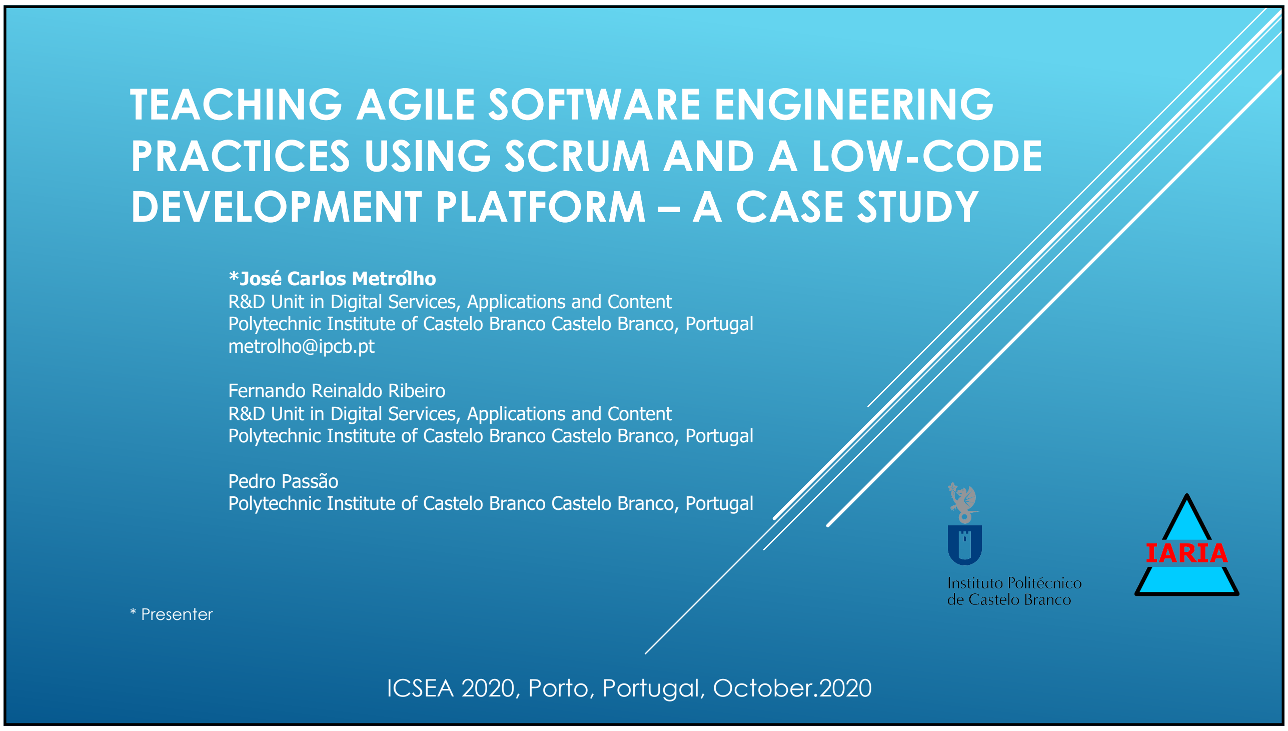


TEACHING AGILE SOFTWARE ENGINEERING PRACTICES USING SCRUM AND A LOW-CODE DEVELOPMENT PLATFORM – A CASE STUDY

***José Carlos Metrôlho**
R&D Unit in Digital Services, Applications and Content
Polytechnic Institute of Castelo Branco Castelo Branco, Portugal
metrolho@ipcb.pt

Fernando Reinaldo Ribeiro
R&D Unit in Digital Services, Applications and Content
Polytechnic Institute of Castelo Branco Castelo Branco, Portugal

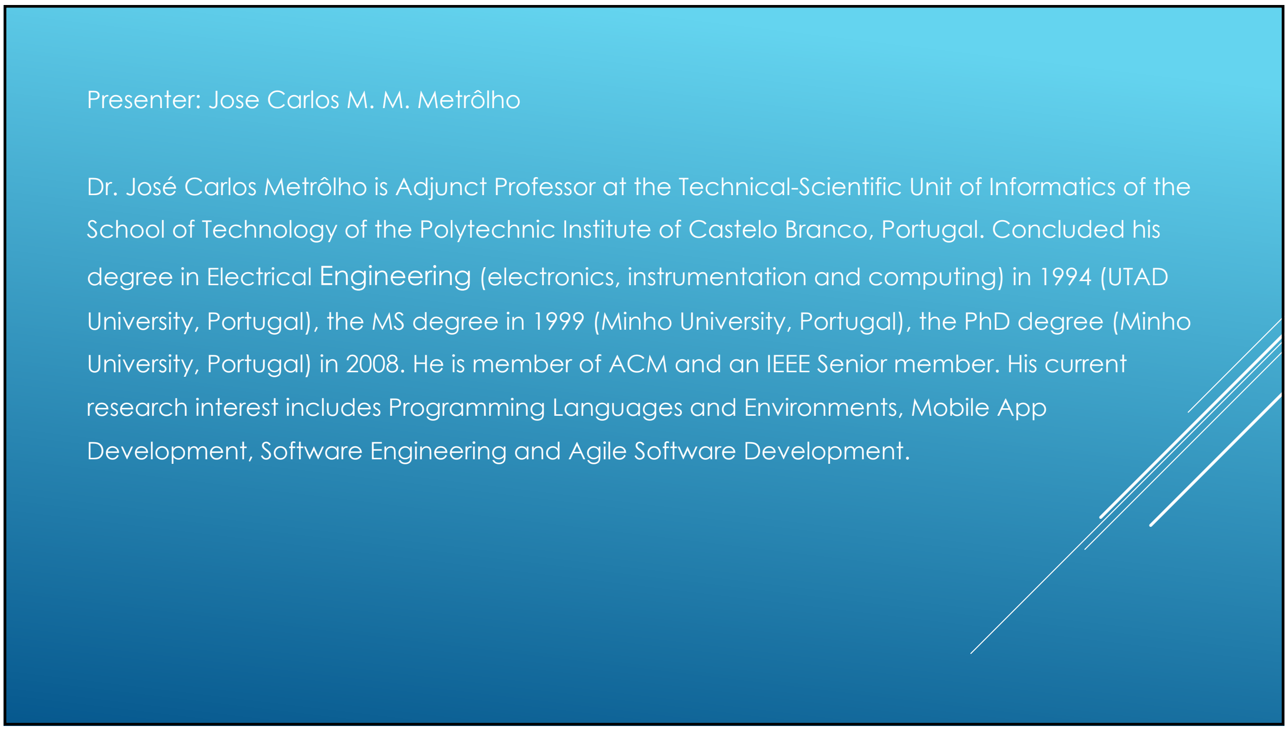
Pedro Passão
Polytechnic Institute of Castelo Branco Castelo Branco, Portugal

Instituto Politécnico de Castelo Branco

IARIA

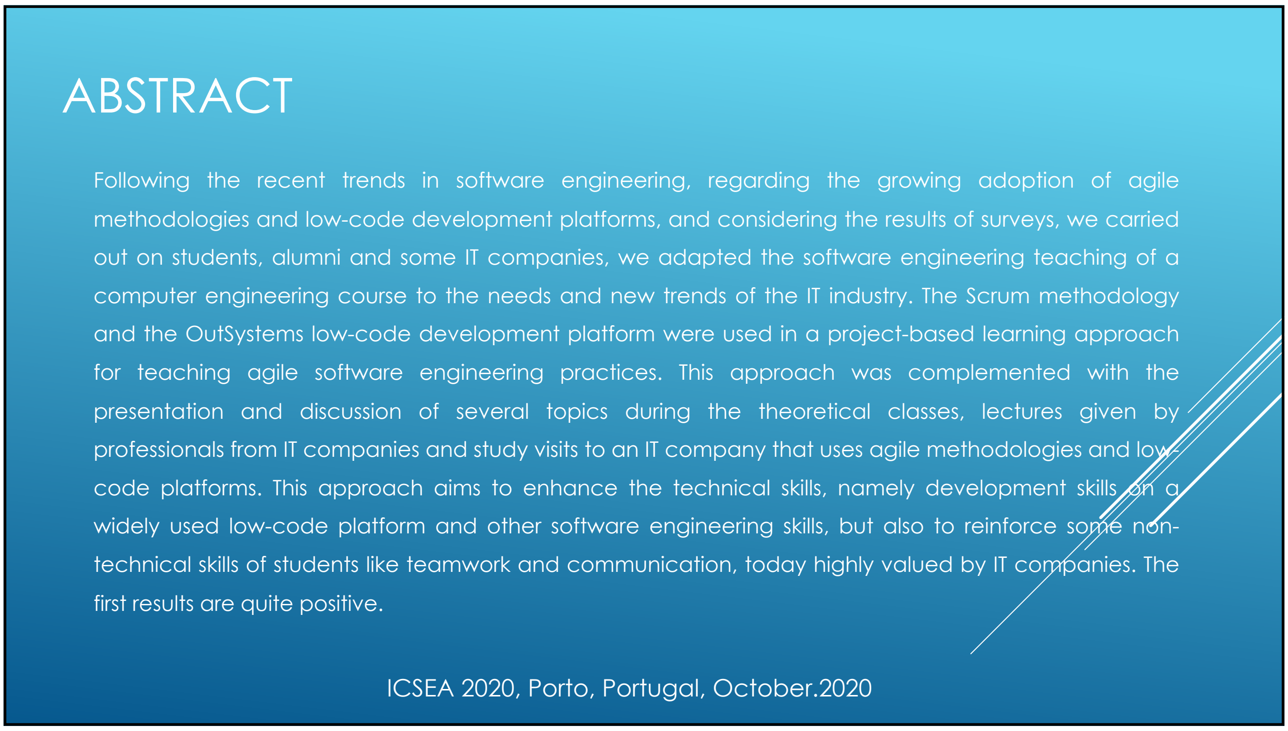
* Presenter

ICSEA 2020, Porto, Portugal, October.2020



Presenter: Jose Carlos M. M. Metrôlho

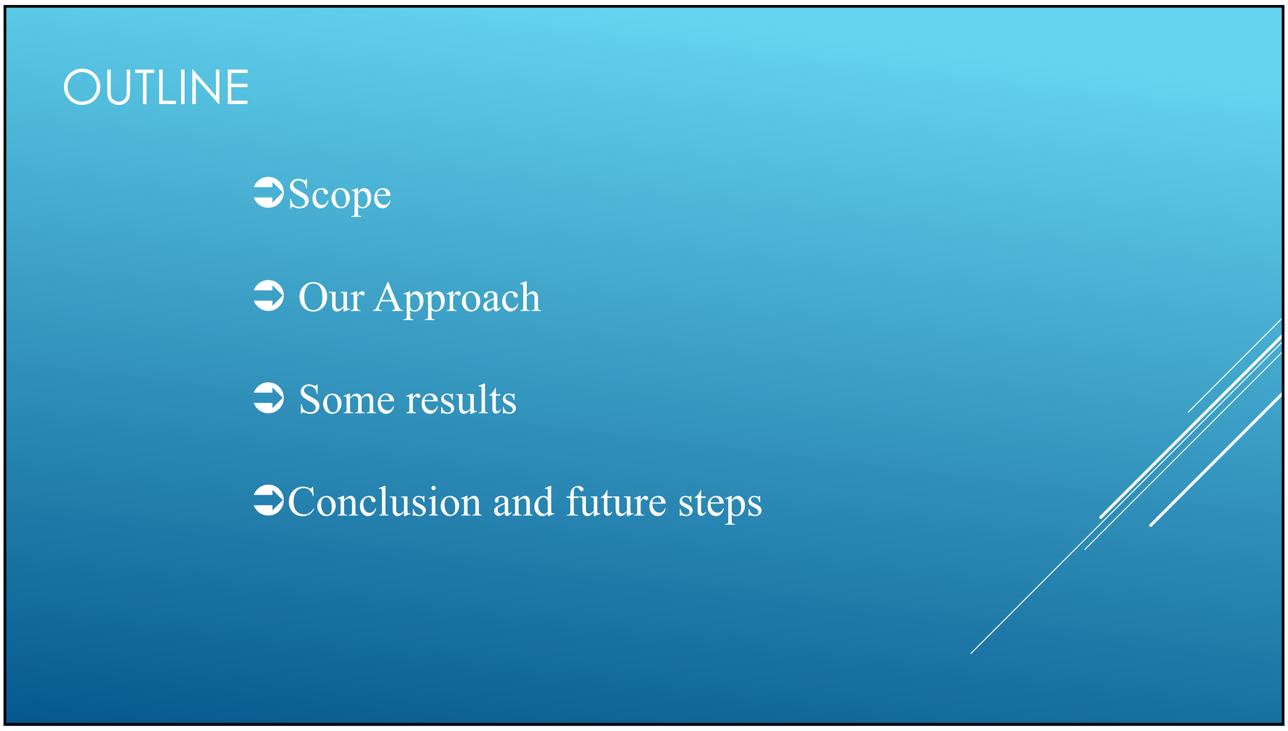
Dr. José Carlos Metrôlho is Adjunct Professor at the Technical-Scientific Unit of Informatics of the School of Technology of the Polytechnic Institute of Castelo Branco, Portugal. Concluded his degree in Electrical Engineering (electronics, instrumentation and computing) in 1994 (UTAD University, Portugal), the MS degree in 1999 (Minho University, Portugal), the PhD degree (Minho University, Portugal) in 2008. He is member of ACM and an IEEE Senior member. His current research interest includes Programming Languages and Environments, Mobile App Development, Software Engineering and Agile Software Development.



ABSTRACT

Following the recent trends in software engineering, regarding the growing adoption of agile methodologies and low-code development platforms, and considering the results of surveys, we carried out on students, alumni and some IT companies, we adapted the software engineering teaching of a computer engineering course to the needs and new trends of the IT industry. The Scrum methodology and the OutSystems low-code development platform were used in a project-based learning approach for teaching agile software engineering practices. This approach was complemented with the presentation and discussion of several topics during the theoretical classes, lectures given by professionals from IT companies and study visits to an IT company that uses agile methodologies and low-code platforms. This approach aims to enhance the technical skills, namely development skills on a widely used low-code platform and other software engineering skills, but also to reinforce some non-technical skills of students like teamwork and communication, today highly valued by IT companies. The first results are quite positive.

ICSEA 2020, Porto, Portugal, October.2020



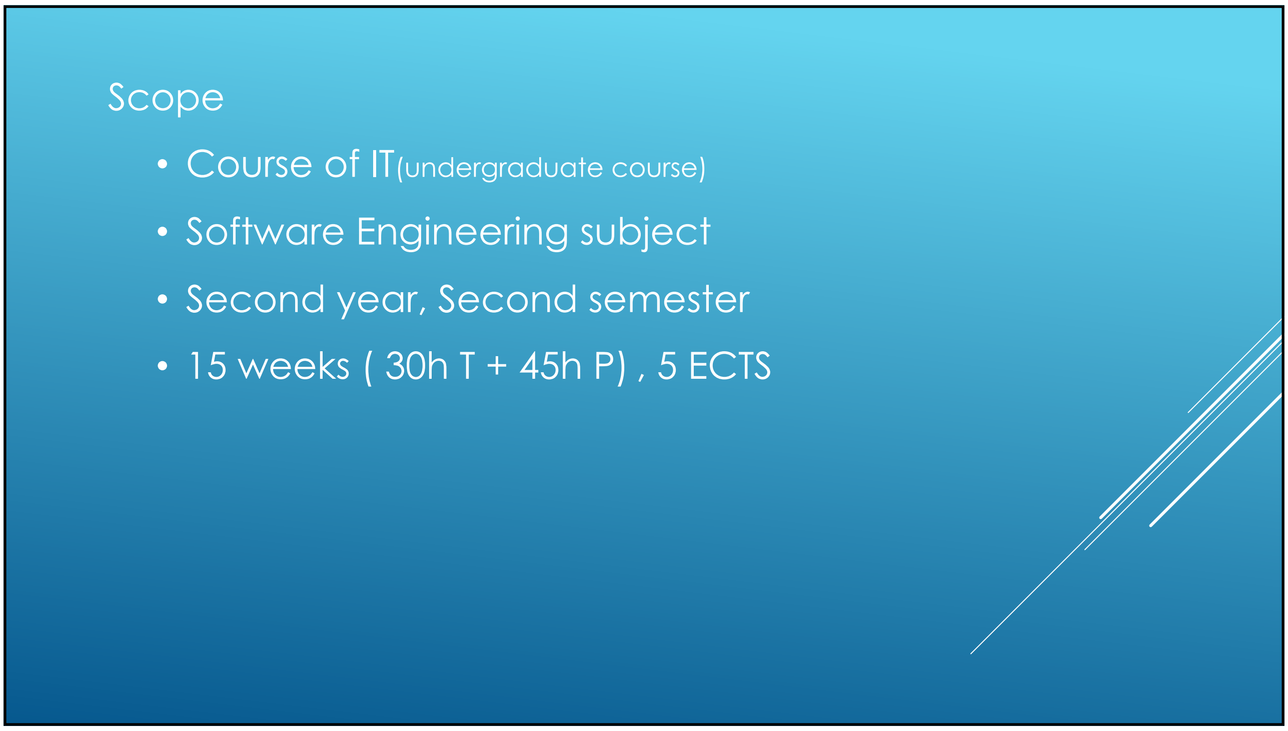
OUTLINE

➲Scope

➲ Our Approach

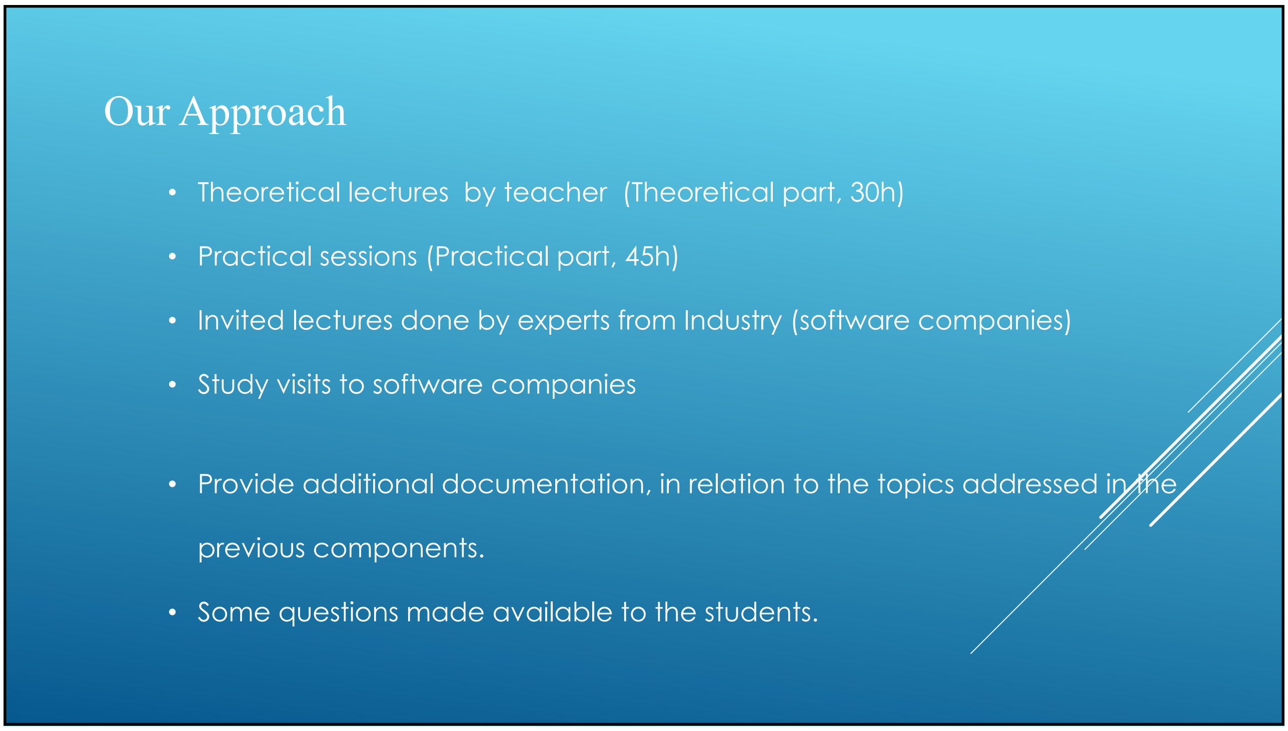
➲ Some results

➲Conclusion and future steps



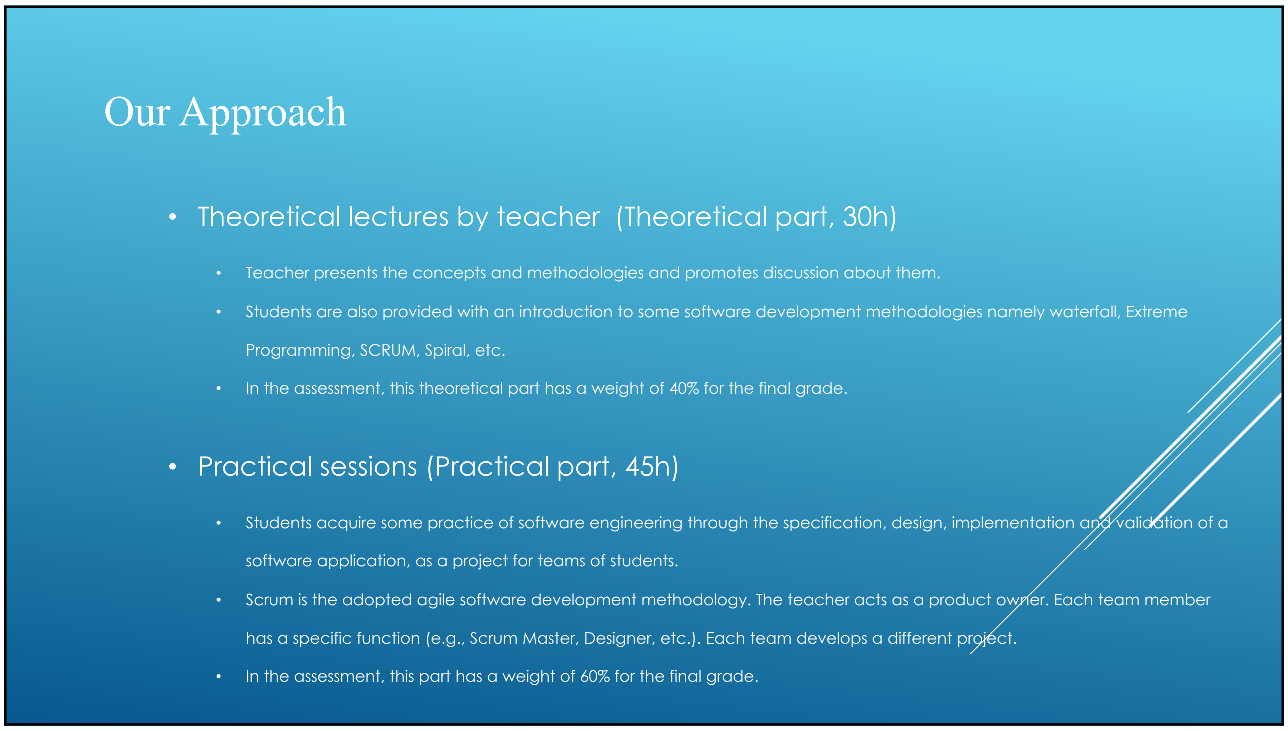
Scope

- Course of IT(undergraduate course)
- Software Engineering subject
- Second year, Second semester
- 15 weeks (30h T + 45h P) , 5 ECTS



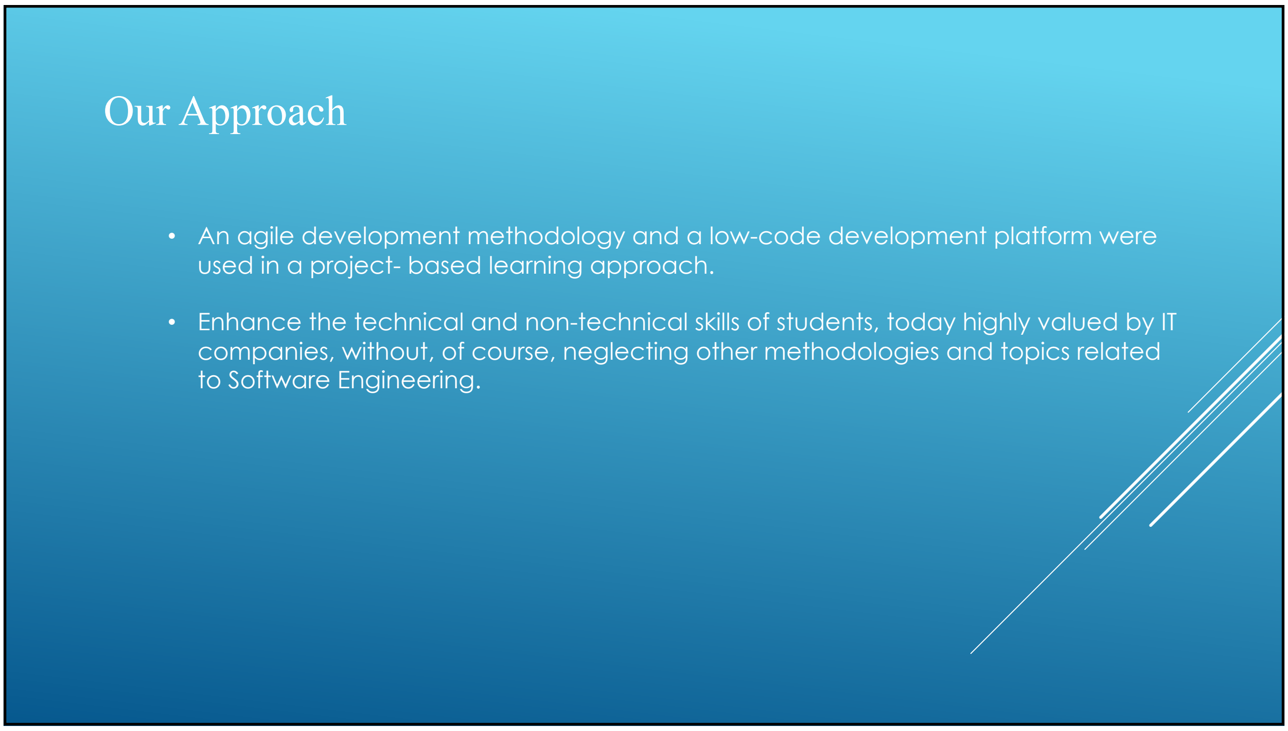
Our Approach

- Theoretical lectures by teacher (Theoretical part, 30h)
- Practical sessions (Practical part, 45h)
- Invited lectures done by experts from Industry (software companies)
- Study visits to software companies
- Provide additional documentation, in relation to the topics addressed in the previous components.
- Some questions made available to the students.



Our Approach

- Theoretical lectures by teacher (Theoretical part, 30h)
  - Teacher presents the concepts and methodologies and promotes discussion about them.
  - Students are also provided with an introduction to some software development methodologies namely waterfall, Extreme Programming, SCRUM, Spiral, etc.
  - In the assessment, this theoretical part has a weight of 40% for the final grade.
- Practical sessions (Practical part, 45h)
  - Students acquire some practice of software engineering through the specification, design, implementation and validation of a software application, as a project for teams of students.
  - Scrum is the adopted agile software development methodology. The teacher acts as a product owner. Each team member has a specific function (e.g., Scrum Master, Designer, etc.). Each team develops a different project.
  - In the assessment, this part has a weight of 60% for the final grade.



Our Approach

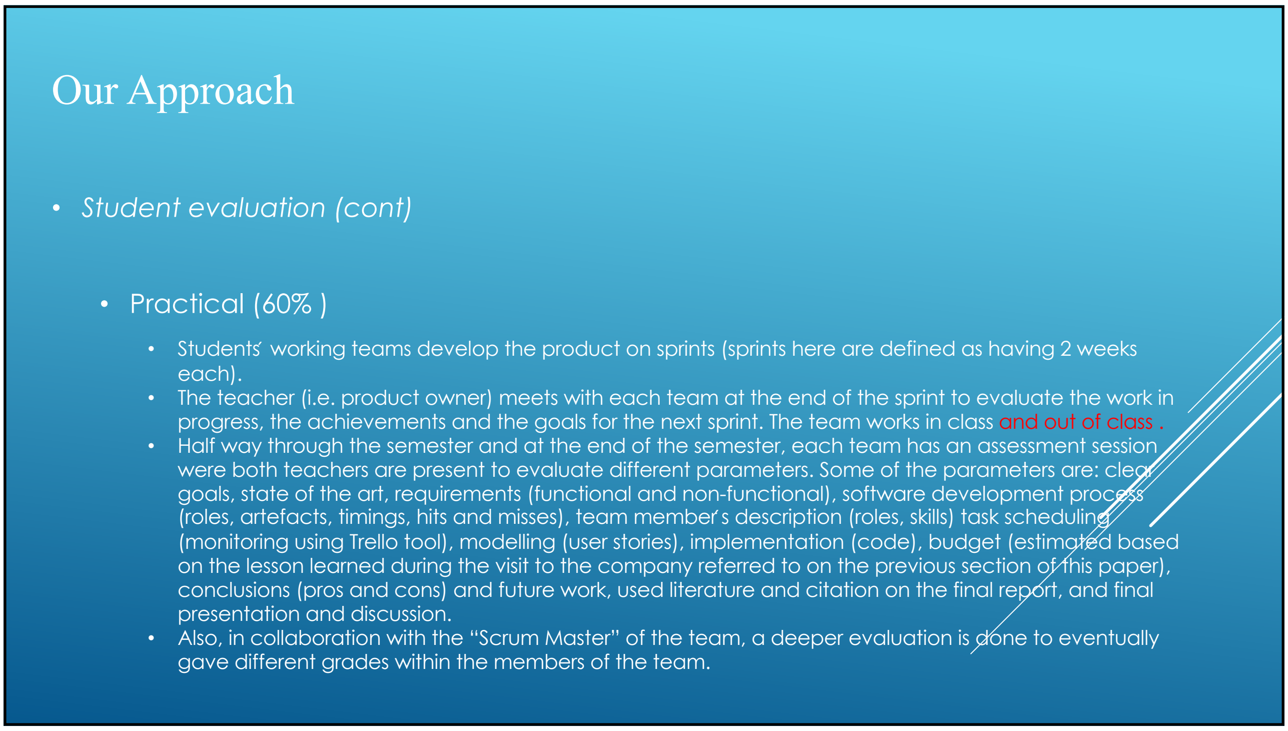
- An agile development methodology and a low-code development platform were used in a project- based learning approach.
- Enhance the technical and non-technical skills of students, today highly valued by IT companies, without, of course, neglecting other methodologies and topics related to Software Engineering.

Our Approach

- Invited lectures done by experts from Industry (software companies)
  - Invited speakers, experts from companies, share know-how about extra topics like Feature Driven Development (FDD), Behaviour Driven Development (BDD), etc.
- One day visits to software companies
  - Professionals explain to the students what they are doing, and which technologies and tools are used to support their activities. Students also had a brief session about software cost estimation.
- Provide additional documentation, regarding topics addressed in the SE subject.

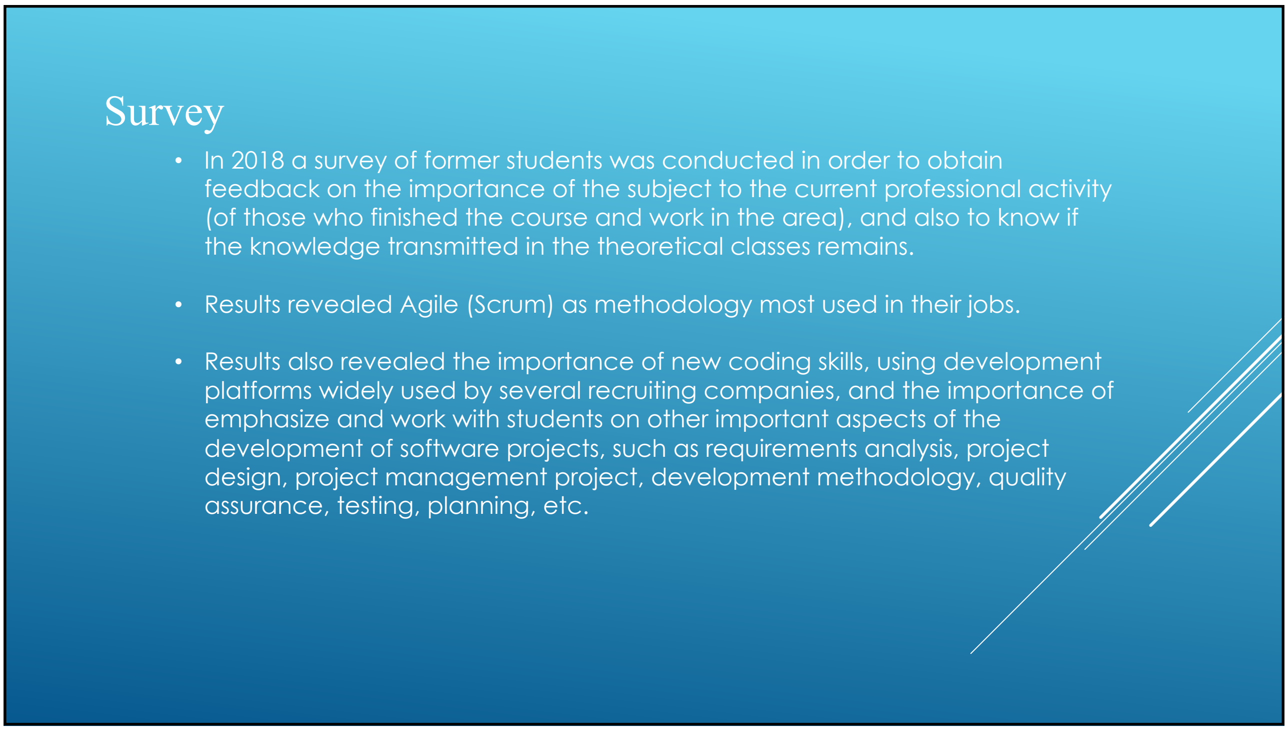
Our Approach

- *Student evaluation*
  - Theoretical (40%)
    - The theoretical evaluation is a written exam over the course material.
    - The exam consists of questions, some of them chosen from a list of questions that are made available to the students along the semester.
    - Reflective questions about software engineering topics.
    - Desirable that students learn and acquire knowledge for a long-life period, mainly to be used after graduation on their job integration experience.



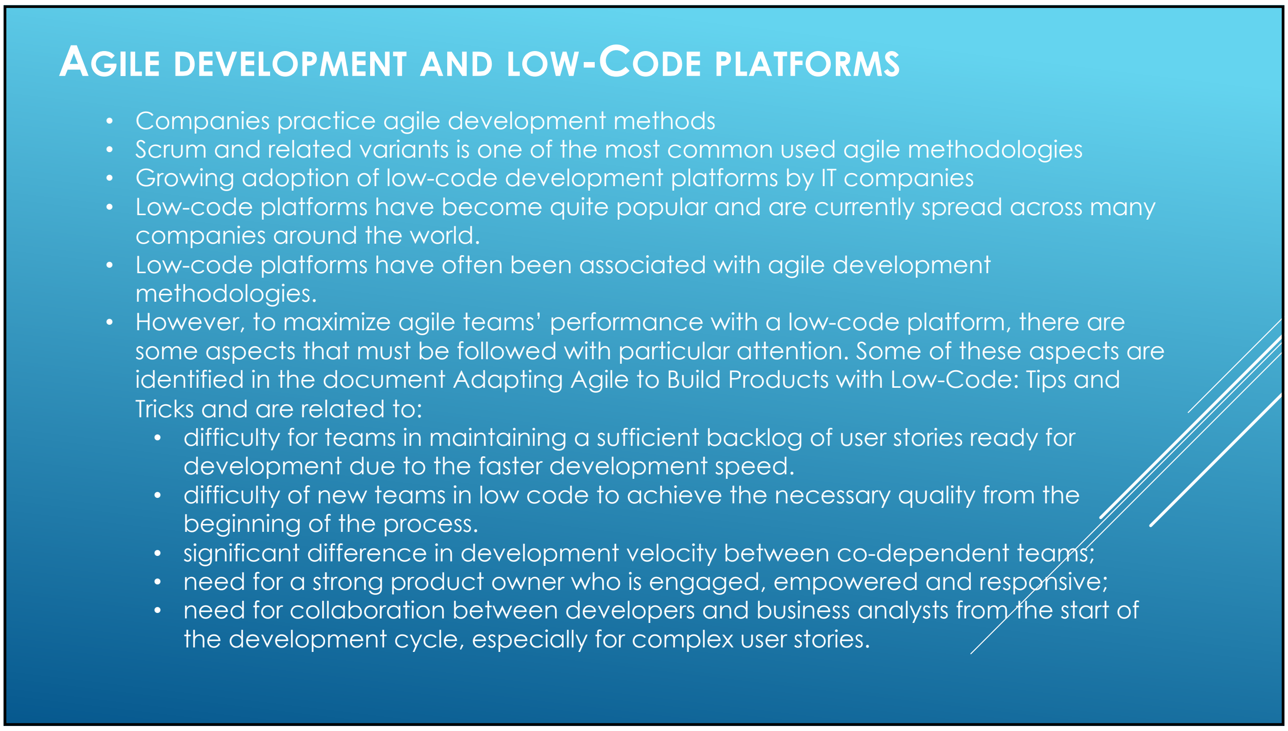
Our Approach

- *Student evaluation (cont)*
  - Practical (60%)
    - Students' working teams develop the product on sprints (sprints here are defined as having 2 weeks each).
    - The teacher (i.e. product owner) meets with each team at the end of the sprint to evaluate the work in progress, the achievements and the goals for the next sprint. The team works in class and out of class .
    - Half way through the semester and at the end of the semester, each team has an assessment session were both teachers are present to evaluate different parameters. Some of the parameters are: clear goals, state of the art, requirements (functional and non-functional), software development process (roles, artefacts, timings, hits and misses), team member's description (roles, skills) task scheduling (monitoring using Trello tool), modelling (user stories), implementation (code), budget (estimated based on the lesson learned during the visit to the company referred to on the previous section of this paper), conclusions (pros and cons) and future work, used literature and citation on the final report, and final presentation and discussion.
    - Also, in collaboration with the "Scrum Master" of the team, a deeper evaluation is done to eventually gave different grades within the members of the team.



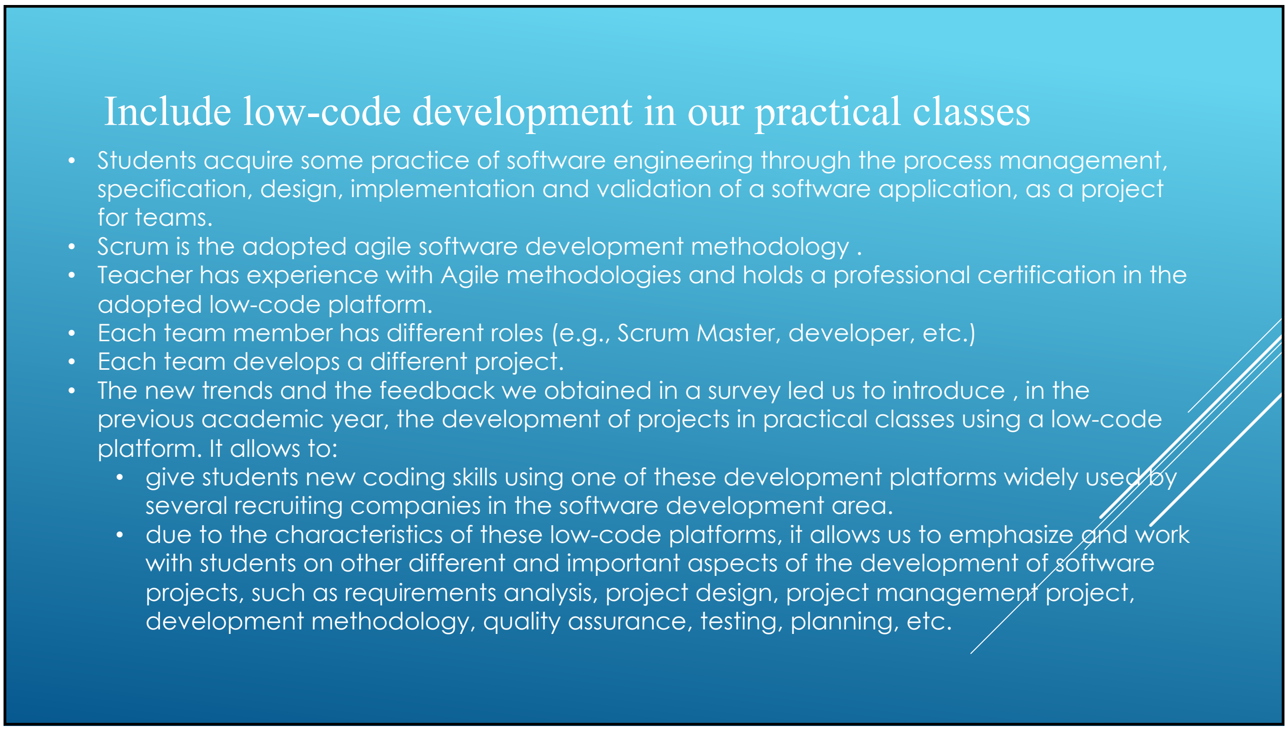
Survey

- In 2018 a survey of former students was conducted in order to obtain feedback on the importance of the subject to the current professional activity (of those who finished the course and work in the area), and also to know if the knowledge transmitted in the theoretical classes remains.
- Results revealed Agile (Scrum) as methodology most used in their jobs.
- Results also revealed the importance of new coding skills, using development platforms widely used by several recruiting companies, and the importance of emphasize and work with students on other important aspects of the development of software projects, such as requirements analysis, project design, project management project, development methodology, quality assurance, testing, planning, etc.



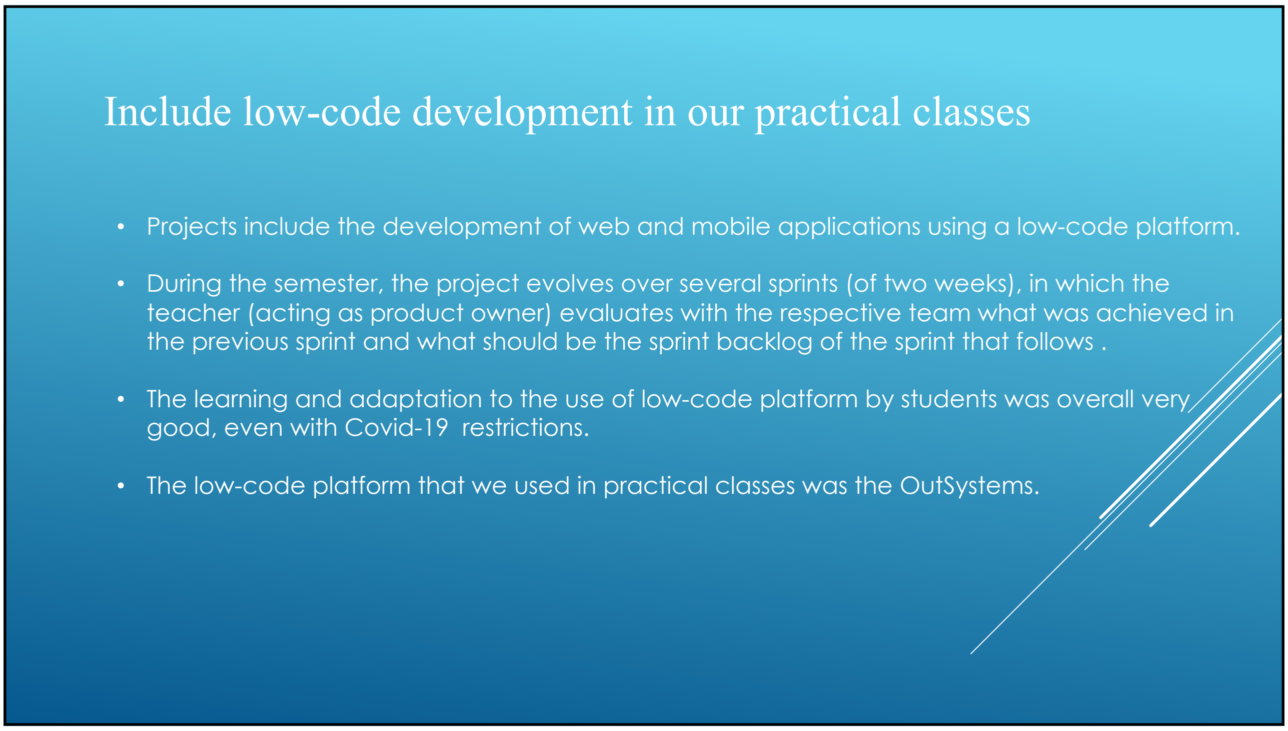
Agile development and low-Code platforms

- Companies practice agile development methods
- Scrum and related variants is one of the most common used agile methodologies
- Growing adoption of low-code development platforms by IT companies
- Low-code platforms have become quite popular and are currently spread across many companies around the world.
- Low-code platforms have often been associated with agile development methodologies.
- However, to maximize agile teams' performance with a low-code platform, there are some aspects that must be followed with particular attention. Some of these aspects are identified in the document Adapting Agile to Build Products with Low-Code: Tips and Tricks and are related to:
  - difficulty for teams in maintaining a sufficient backlog of user stories ready for development due to the faster development speed.
  - difficulty of new teams in low code to achieve the necessary quality from the beginning of the process.
  - significant difference in development velocity between co-dependent teams;
  - need for a strong product owner who is engaged, empowered and responsive;
  - need for collaboration between developers and business analysts from the start of the development cycle, especially for complex user stories.



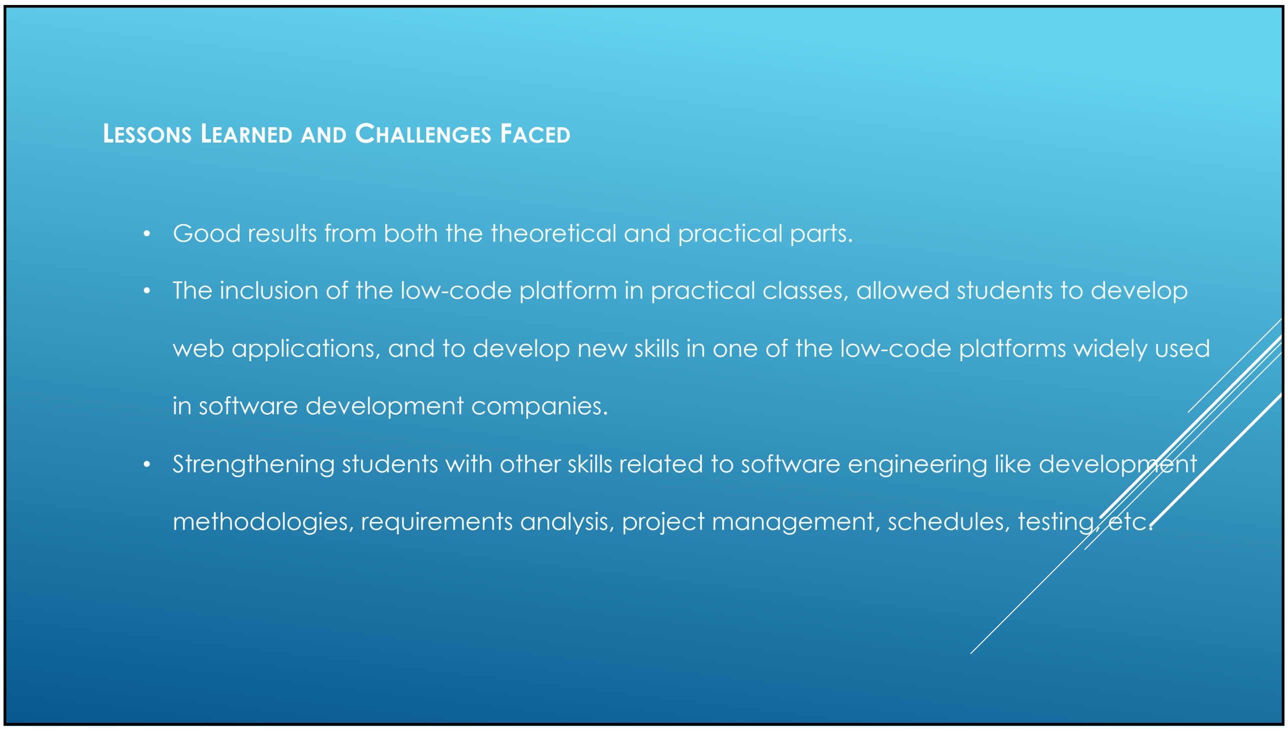
Include low-code development in our practical classes

- Students acquire some practice of software engineering through the process management, specification, design, implementation and validation of a software application, as a project for teams.
- Scrum is the adopted agile software development methodology .
- Teacher has experience with Agile methodologies and holds a professional certification in the adopted low-code platform.
- Each team member has different roles (e.g., Scrum Master, developer, etc.)
- Each team develops a different project.
- The new trends and the feedback we obtained in a survey led us to introduce , in the previous academic year, the development of projects in practical classes using a low-code platform. It allows to:
  - give students new coding skills using one of these development platforms widely used by several recruiting companies in the software development area.
  - due to the characteristics of these low-code platforms, it allows us to emphasize and work with students on other different and important aspects of the development of software projects, such as requirements analysis, project design, project management project, development methodology, quality assurance, testing, planning, etc.



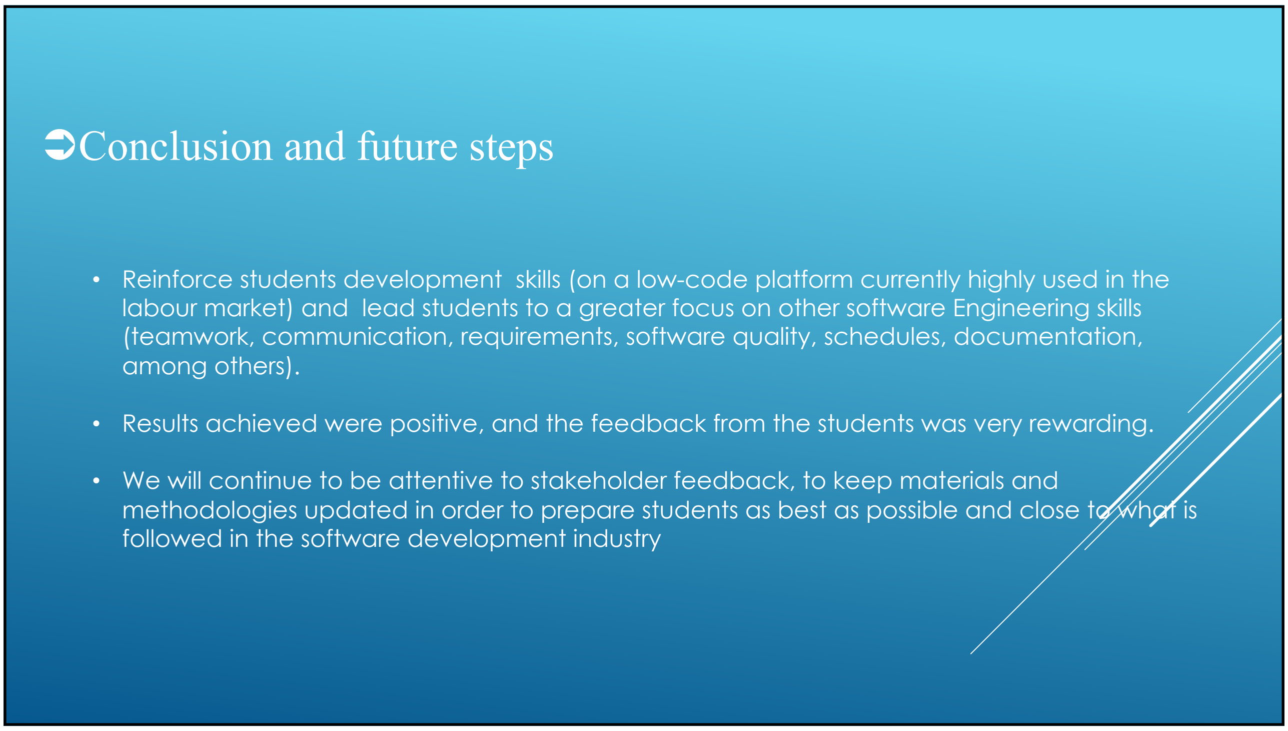
Include low-code development in our practical classes

- Projects include the development of web and mobile applications using a low-code platform.
- During the semester, the project evolves over several sprints (of two weeks), in which the teacher (acting as product owner) evaluates with the respective team what was achieved in the previous sprint and what should be the sprint backlog of the sprint that follows .
- The learning and adaptation to the use of low-code platform by students was overall very good, even with Covid-19 restrictions.
- The low-code platform that we used in practical classes was the OutSystems.



Lessons Learned and Challenges Faced

- Good results from both the theoretical and practical parts.
- The inclusion of the low-code platform in practical classes, allowed students to develop web applications, and to develop new skills in one of the low-code platforms widely used in software development companies.
- Strengthening students with other skills related to software engineering like development methodologies, requirements analysis, project management, schedules, testing, etc.



➲Conclusion and future steps

- Reinforce students development skills (on a low-code platform currently highly used in the labour market) and lead students to a greater focus on other software Engineering skills (teamwork, communication, requirements, software quality, schedules, documentation, among others).
- Results achieved were positive, and the feedback from the students was very rewarding.
- We will continue to be attentive to stakeholder feedback, to keep materials and methodologies updated in order to prepare students as best as possible and close to what is followed in the software development industry