DOI: 10.2195/lj_proc_enke_en_202510_01

Application-Oriented Learning in Engineering: A Retrospective Study on Teaching Cyber-Physical Intralogistics Systems

Constantin Enke[1], **Marvin Rüdt**[1], **Lars Ohnemus**[1], **Pietro Schumacher**[1], **Hao Pang**[1], **Kai Furmans**[1]

Institute for Material Handling and Logistics
Karlsruhe Institute of Technology

Engineering education faces declining enrolments, international competition, and the challenge of AI solving traditional exam tasks. Intralogistic systems link mechatronics and algorithms, offering tangible challenges for application-oriented learning. This paper introduces a classification framework based on *Input*, *Application*, and *Examination*, and applies it to evaluate three representative courses at the Karlsruhe Institute of Technology. The analysis shows that freedom in learning, such as self-study phases, requires corresponding structure in assessment to ensure knowledge acquisition. Team-based tasks benefit from clearly defined roles and organizational scaffolding to prevent conflicts and unequal participation. The workload distribution strongly depends on course duration: very compact formats leave little room for catch-up, while extended formats require intermediate milestones to maintain motivation. Grading remains a particular challenge, as examinations must balance fairness with recognition of teamwork and practical achievements. Finally, the growing role of artificial intelligence introduces both risks and opportunities: while AI can reduce the need for routine coding, it creates new demands for creative, critical, and system-level tasks. These findings provide practical guidance for designing and evaluating interactive courses also in other areas then intralogistics.

[Keywords: Teaching, Application-Oriented, Intralogistics]

1 Introduction

Engineering education is undergoing significant changes. In Germany, declining enrollment numbers in technical subjects and a highly competitive job market create pressure on universities to offer attractive and relevant programs. At the same time, the emergence of artificial intelligence tools, capable of solving a large share of traditional exam tasks, challenges conventional assessment methods and calls for new approaches to teaching and learning.

At the Institute for Material Handling and Logistics (Karlsruhe Institute of Technology), we address these developments through application-oriented course designs centered on intralogistics and, in particular, cyber-physical intralogistics systems (CPIS) [1]. CPIS offer a coherent way to model and integrate physical assets (e.g., mobile robots, robotic arms, conveyors) and logical assets (e.g., simulations, control systems), including their interfaces and communications. This domain is especially well suited for interactive learning: it combines mechatronic components with data-driven and algorithmic methods (e.g., pathfinding, storage strategies, system dimensioning) in tasks that are tangible and widely understood (e.g., e-commerce warehouses, automotive supply systems). The inherent complexity enables meaningful team-based work with clear role ownership while remaining accessible to students.

Prior work indicates that interactive learning—such as group work, discussions, problem solving, and hands-on projects—tends to outperform purely teacher-centered formats in terms of depth of understanding, retention, engagement, and critical thinking [2]. Building on this, our contribution is threefold: (i) a classification along the dimensions of *Input*, *Application*, and *Examination* to structure interactive course design, (ii) an evaluation of multiple courses in intralogistics, and (iii) a synthesis of experiences and best practices for application-oriented teaching.

The paper proceeds as follows. Section 2 introduces the classification and its characteristics. Section 3 outlines the evaluation methods. Section 4 presents three representative courses in detail. Section 5 discusses cross-course findings, including design trade-offs and practical guidance for instructors. Section 6 concludes with implications for technical logistics education. Extended course descriptions are included to provide practitioners with concrete, transferable models.

2 Framework for Classifying Application-Oriented Courses

Analyzing and comparing application-oriented courses in engineering education requires a systematic and reproducible framework. This section introduces such a framework by distinguishing the three areas of teaching *Input*, *Application*, and *Examination* and by specifying the key characteristics that describe how these areas are implemented. Each area, together with its characteristics, forms the building blocks of courses.

The building blocks represent interdependent elements of the educational concept. Together they reflect both the pedagogical intent and the practical implementation. The overall design of a course results from the selection and weighting of these blocks, which strongly influences student engagement, the achievement of learning objectives, and the transfer of theoretical knowledge into practical competence [3].

2.1 Input

The input phase constitutes the primary mode of knowledge acquisition and is therefore a central pedagogical decision in application-oriented learning. It determines how new material is introduced and how actively students are engaged in the learning process.

2.1.1 Input Types

Input can be either **theoretical**, for example by conveying abstract concepts such as modeling approaches or algorithmic methods, or **practical**, for example by presenting industry case studies, introducing algorithms in relation to hardware or logistics processes, or explaining software concepts that will later be applied in labs or simulations. In both cases, the focus remains on providing knowledge and orientation. The balance between these forms determines both the level of abstraction and the immediate relevance for subsequent application.

2.1.2 Role of Instructor and Student

Input formats can also be positioned along a continuum from **teacher-centered** to **student-centered**. In the teacher-centered model, the instructor acts as the primary source of knowledge, efficiently conveying foundational content to large groups through lectures or demonstrations [4]. By contrast, a student-centered approach emphasizes active knowledge construction, self-directed learning, and collaboration, as described in constructivist learning theories. Methods such as flipped classrooms [5], problem-based learning (PBL) [6], and project-based learning (PjBL) [7] embody this orientation. Empirical studies show that these active strategies foster deeper cognitive engagement and more sustainable learning outcomes than purely passive formats [2]. The balance between teacher- and student-centered inputs therefore shapes both the roles of instructors and students and the effectiveness of subsequent application-oriented tasks.

2.2 Application

The application addresses how students actively engage with and utilize the knowledge they have aquired, thereby bringing the practical, application-oriented dimension of learning into focus. Different types and characteristics of application can be distinguished, each serving specific pedagogical functions.

2.2.1 Application Types

Exercises typically consist of structured, often repetitive tasks that reinforce theoretical concepts and foundational skills. They provide a low-stakes environment in which students practice procedures or apply formulas, thereby consolidating knowledge through repetition and incremental mastery. **Laboratory work** involves hands-on activities directed toward solving real-world problems, using physical systems or simulations. This type of application is closely aligned with experiential learning theories, which emphasize learning through active experimentation and reflection [8]. Beyond technical proficiency, laboratory experiences cultivate competencies in scientific work, data analysis, and critical evaluation [9]. The selection and design of application formats directly affect the depth of student understanding, the development of problem-solving and critical thinking skills, and ultimately the extent to which theoretical knowledge is transformed into professional competence.

2.2.2 Organizational Structure

The organizational structure describes the extent to which a course prescribes its schedule, workflow, and sequence of activities. A **high organizational structure** corresponds to tightly scaffolded formats, where learning pathways are clearly delineated and instructor-directed. This scaffolding is particularly effective for novice learners, as it reduces cognitive load by chunking information and guiding their attention through complex tasks. In contrast, a **low organizational structure** grants students considerable freedom to plan and manage their activities, promoting learner autonomy and aligning with theories of self-regulated learning. However, it relies on students having developed sufficient self-management skills [10].

2.2.3 Technical Structure

The technical structure refers to the scope of restrictions regarding the tools, methods and technologies used in the course. A **high technical structure** mandates specific hardware, software, or procedural approaches, facilitating comparability across students and easing support requirements.

Conversely, a **low technical structure** empowers learners to select and experiment with their own tools, fostering creativity, problem-solving, and innovation. This openness resonates with the concept of affordances, which highlights how different tools offer distinct opportunities for learner interaction and outcomes [11].

2.2.4 Task Design

The task design determines the nature and scope of challenges students encounter, directly influencing learning outcomes and cognitive engagement. **Uniform tasks**, identical for all learners, facilitate comparability and are commonly used in traditional instructional models. **Varied tasks** offer different but equivalent challenges tailored to diverse learner readiness or interests; a principle supported by differentiated instruction frameworks that enhance academic achievement and engagement by responding to individual needs and preferences [12]. **Complementary tasks** are distinct yet interdependent components within an integrated project, promoting systems thinking, collaboration, and project management skills vital in engineering contexts. This collaborative integration mirrors design-centered educational models that cultivate interdisciplinary thinking in real-world engineering scenarios [13].

2.2.5 Collaboration Structure

The collaboration structure specifies whether learning tasks are completed through individual effort or collaboration. **Individual work** promotes personal accountability, independence, and mastery of content. By contrast, **group work** reflects the collaborative reality of contemporary engineering practice and is strongly supported by educational theory and empirical research. Meta-analyses show that cooperative learning increases content acquisition, retention, and higher-order problem-solving compared to individual work [14]. Effective group work enhances communication, negotiation, and team-based problem solving, yet its success hinges on deliberate structuring, specifically the inclusion of individual accountability and task interdependence to mitigate issues such as social loafing. Studies demonstrate that when group tasks are structured so that each student's contribution is necessary and visible, accountability and overall engagement is enhanced, and free-riding is reduced [15].

2.2.6 Transfer Challenge

The transfer challenge refers to the cognitive distance between knowledge acquisition and its application, commonly framed in educational psychology as transfer of learning. **Near transfer** occurs when learners apply knowledge in contexts closely resembling the instructional setting, such as solving familiar problems or exercises. **Far transfer** requires applying concepts to novel or ill-structured contexts, often encountered in real-world professional practice. While near transfer supports consolidation of procedural skills, far transfer represents a higher-level educational objective, as it demonstrates adaptive expertise and deep understanding [16].

2.3 Examination

The examination structure defines how student learning is evaluated and varies in the degree of freedom students have in shaping their assessment.

2.3.1 Examination Types

At one end of the spectrum, **self-directed** approaches dispense with formal exams and rely on portfolios, projects, or learning contracts. These maximize autonomy and emphasize intrinsic motivation but require strong self-regulation and close instructor support [17]. More structured formats include **presentations** or **reports**, in which students either select their own focus or follow instructor-specified topics or criteria to ensure core learning outcomes are addressed. Both approaches emphasize reasoning, problem-solving, and communication skills, with guided formats providing greater comparability across students [18]. **Oral examinations** allow in-depth assessment of knowledge and reasoning through interactive dialogue, offering rich insights but demanding significant resources [19]. At the most standardized end, **written examinations**, in essay, problem-solving, or multiple-choice format, enable efficient, large-scale assessment but tend to privilege recall and procedural skills over application and creativity [20]. The choice of examination structure thus has a decisive impact on how students prepare for a course, the skills they prioritize, and the extent to which assessment fosters short-term performance versus long-term competence development.

2.3.2 Grading

Grading approaches in application-oriented learning range from formative, feedback-based models to highly standardized systems, each influencing how students engage with learning tasks. **Ungraded** lectures eliminate scores in favor of reflection and feedback, encouraging intrinsic motivation and deeper learning, though it requires robust support structures [17]. **Pass/Fail** offers a binary judgment of competence, reducing stress and competition but limiting differentiation across performance levels [21]. **Letter or numeric grades** offer a ranked and easily interpretable measure of achievement, particularly for external stakeholders, but have been criticized for promoting surface-level learning and performance-oriented strategies over mastery [21].

Examination may take the form of isolated exams at the end of the course or continuous assessment distributed across multiple smaller tasks, such as quizzes, assignments, or project milestones. Continuous assessment encourages regular engagement and integrates feedback throughout the learning process, though it can increase workload if not carefully balanced [22].

3 Evaluation of the Field Studies

When analyzing the field studies in Section 4, three main aspects are considered: the course structure, the expected student effort and time investment, and the retrospective evaluation of the courses by both students and instructors.

3.1 Course Structure

The course structure represents the chronological sequence of course components. To highlight similarities and differences across courses, we use swimlane charts as a visual representation. They are divided into the three areas *Input*, *Application*, and *Examination*. Each area contains the relevant building blocks, which are evaluated according to the characteristics introduced in Section 2.

Figure 1 illustrates a typical university lecture as an example. Here, recurring events such as lectures are shown as continuous blocks to improve readability. The purpose of this illustration is also to make both the temporal structure and the logical sequence of building blocks visible. Changes in characteristics across the timeline (e.g., transfer challenge, technical structure) can be observed within the same area. Next to each block name, the block type is indicated. For instance, a lecture block is categorized as Type *theoretical* and has the center of teaching characteristic *teacher-centered*. For reasons of space, icons are used to represent characteristics, as summarized in Table 1.

For additional clarification, the structure and special features of each course are described in the *Content* section of the respective analyses in Section 4.

3.2 Expected Student Effort and Time Investment

The scope of the courses is determined by the European Credit Transfer and Accumulation System (ECTS), where one credit point corresponds to approximately 30 hours of student work. To illustrate how this workload is typically distributed across different activities, we use a sunburst chart as a visual representation.

Figure 2 presents the classic 4 ECTS course (120 hours total workload) as an example. The chart depicts the relative shares of lectures (2 hours per week), exercises (2 hours per week), and the written examination, including associated preparation and follow-up. The inner ring indicates the absolute distribution of hours, while the outer rings highlight the proportional contribution of each component, with magnification where applicable (indicated by solid lines).

Table 1: Iconography for course characteristics used in figures.

Area	Characteristic	Aspect	Icon
Input	Type	Theoretical	
		Practical	
	Center of Teaching	Teacher	
		Mixed	
		Student	
Application	Type	Exercise	
		Lab	
	Structure	High	↑
		Medium	↔
		Low	↓
	Transfer	Near	
		Far	
	Task Design	Uniform	
		Complementary	
	Collaboration	Individual	
		Group	
Exams	Type	Presentation	
		Report	
		Oral	
		Written	
	Grading	Ungraded	
		Pass/Fail	✓ / ✗
		Numeric	#

3.3 Evaluation of Teaching Success by Students and Teachers

Each course is evaluated retrospectively using a combination of quantitative and qualitative methods. Where available, structured course assessments and standardized student surveys capture direct feedback on learning outcomes, teaching quality, and course design. In cases where such instruments are not administered, e.g., if the number of participants was to small to allow anonymous participation, the analysis draws on instructors' experience reports and, when accessible, direct feedback from participating students. These narrative reflections complement the formal evaluations by offering insights into perceived strengths and challenges of the course and highlighting individual perspectives on its impact. In addition, grading distributions and participation statistics are examined to assess how specific course characteristics influence learning quality and engagement. Together, these methods provide a balanced

DOI: 10.2195/lj_proc_enke_en_202510_01

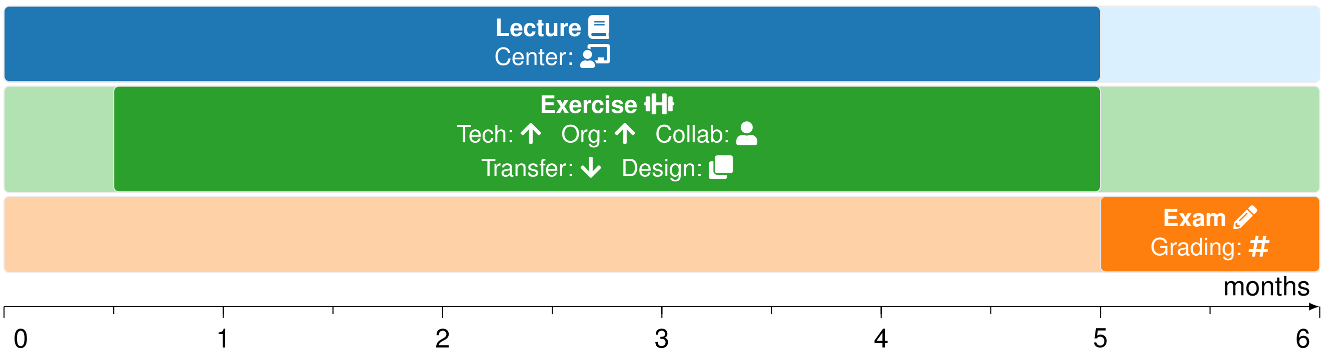

Figure 1: Schedule and distribution of an exemplary course consisting of lectures (Input, blue), exercises (Application, green), and a written exam after 5 months (orange). Characteristics according to Table 1.

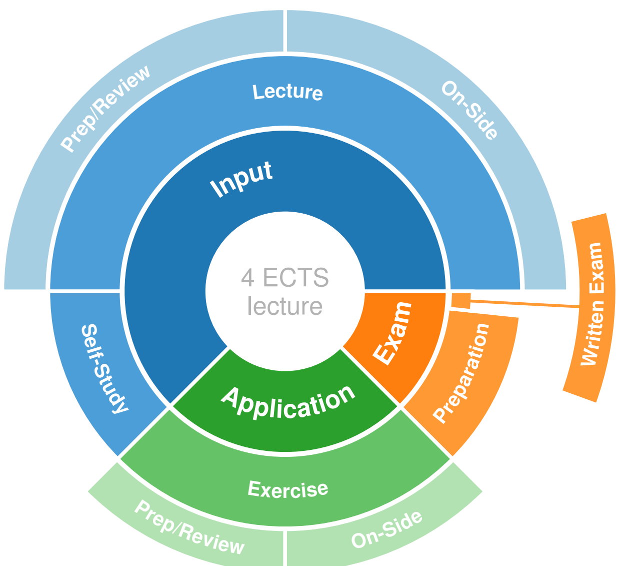

Figure 2: Workload distribution of a 4 ECTS/120 hours course divided into Input (75 h; 62.5%), Application (30 h; 25%), and Exams (15 h; 12.5%).

basis for assessing the development and effectiveness of the course over time, as presented in the following chapter.

4 Field Studies

Based on the previously presented classification and evaluation methods, this chapter introduces a set of representative courses at the Institute for Material Handling and Logistics. We begin with the *Plug-and-Play Material Handling Lab*, the first interactive course introduced at the IFL. This is followed by *Seamless Engineering*, which has the largest number of participants. Both of these courses use a small scale model hardware. The *Industrial Mobile Robotics Lab*, on the other hand uses industrial scale hardware and is the newest lecture, introduced in 2024. Finally, we briefly describe additional fields for application-oriented learning in engineering at our institute.

The following section provides detailed descriptions of the representative courses. The level of detail is intentional: the section is designed not only to document lectures and results but also to serve as a transferable resource for practitioners seeking concrete models of application-oriented teaching.

4.1 Plug-and-Play Material Handling Lab

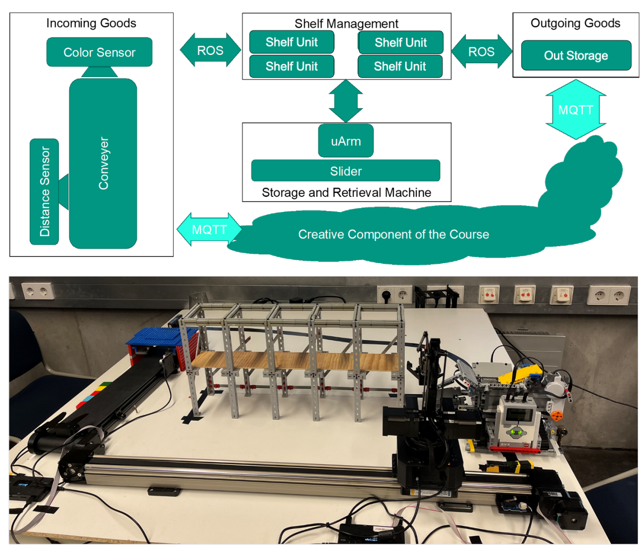

Figure 3: Concept and possible solution of PnP course with student-built LEGO Mindstorms line-following robot.

The *Plug-and-Play Material Handling Lab* focuses on teaching robotics and intralogistics concepts through hands-on use of a system of small scale model hardware of conveyors and robots and a LEGO mindstorm setup for an creative task (see Figure 3). With a total of 4 ECTS in two weeks plus preparation, the course is characterized by a steep learning curve and a high degree of intensity, as can be seen in the course overview in 5. The course is offered to a maximum of 15 students split into 3 Teams. On the technical side, students learn Python, Ubuntu, ROS, and fundamentals of material handling systems. On the soft-skill side, emphasis is placed on project management, teamwork, and critical evaluation of solutions. This course uses a pass/fail grading system.

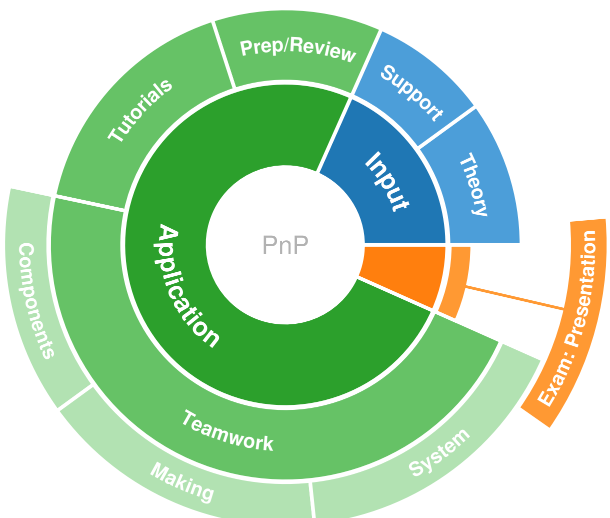

Figure 4: Approximate distribution of the workload for PnP (Input: 18%, Application 75%, Exam: 7%).

Content: The course begins starts with a teacher-centered input sessions. Students are introduced to the programming language python and ROS through live coding sessions and interactive exercises. This is followed by a supervised self-study phase using adapted ROS tutorials. Teams are formed by the instructors based on a self-assessment of prior knowledge. Another input is a high level introduction to the model hardware. By the end of day two, student have all the tools to work on the hardware independently, concluding the near transfer of learned input.

On day three, students are presented with the central assignment: the design and implementation of a functioning logistics system focusing on the interaction of multiple material flow participants. All teams have a uniform task design and must perform a certain amount of transfer effort, as the task develops from simple implementation to dealing with cumulative errors and complex interrelationships. Three Milestones structure the work with the first end of week 1 where teams present the planned system structure, specify planned interactions between material flow participants, and develop a timeline for achieving their goals. Beginning of the second week, the students are introduced to the additional creative task of building a own line-following robot using LEGO mindstorm hardware. An intermediate presentation of the progress and the concept for the creative task Is expected on Milestone two End of day six or mid of day seven. This can be adjusted according to the students progress. For the Final event end of week 2, the teams demonstrate their solutions with a power point presentation and live system demonstration. The performance over the two weeks, within the presentation and followup questions is relevant for the pass/fail grading system. As an additional motivational element, an ungraded competition is held in which teams execute predefined tasks as quickly as possible to achieve the highest semester score.

To support daily progress, each day begins with a short stand-up meeting, during which team representatives outline the day's planned activities. Each day ends with a 15-minute wrap-up, during which progress is presented by different team members each day. There are daily consultation hours after lunch when students can ask questions, and instructors are also available during course hours to help with hardware-related issues. This provides a high level of technical support. The organisational structure is rated as medium in terms of teamwork, since, compared to other lectures, the team roles and presentation structure were decided by the students.

In previous years, the course was also offered in a shortened one-week variant for an equivalent of 2 ECTS. In this version, the introduction and tutorials were condensed to approximately 1.5 days, only two milestones were defined, and the independent construction of a LEGO robot was omitted. The course concluded with a system concept presentation on day three and a final demonstration on day five.

Evaluation: Students rated the course content as demanding and the workload as high. The steep learning curve, particularly for those without prior programming experience, was often cited as the main challenge. The average workload was considered adequate, but in years with higher hardware complications, such as failing motors or bugs in machine code, this was a major point of critique. The practical application was highlighted as a significant benefit of the course in the student evaluation. Some students viewed team assignments, which were determined by instructors, as critical due to heterogeneity in student backgrounds. Here, a low of organizational structure provided by the instructors led to some creative milestone presentations, but also meant that some students participated less than others. Additionally, some participants found working with the unfamiliar Ubuntu operating system challenging.

From the instructors' perspective, the compact two-week format requires considerable effort and supervision. Certain problems, particularly hardware issues, could not be solved independently by students and required direct sup-

DOI: 10.2195/lj_proc_enke_en_202510_01

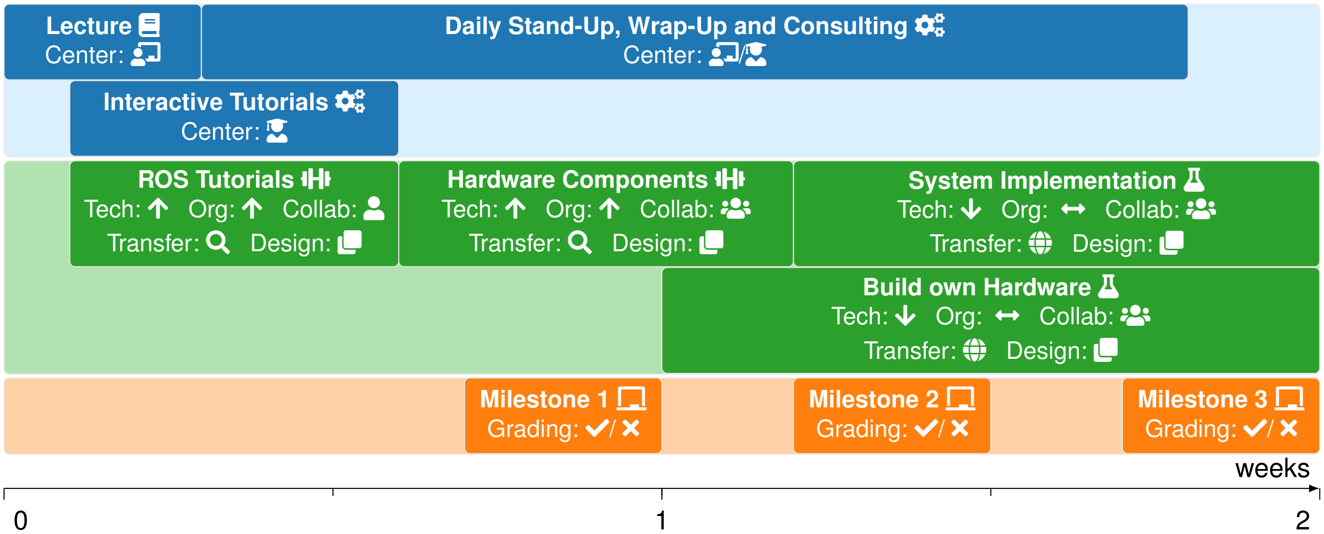

Figure 5: Schedule and logical sequence of building blocks of the Plug and Play Material Handling Lab with Input (blue), Application (green), and Exams (orange). Characteristics according to Table 1.

port from instructors. In the one-week crash course variant, the compressed schedule further increased the required instructor presence.

4.2 Seamless Engineering

Seamless Engineering is an interdisciplinary course that teaches mechatronics students technical skills required to collaboratively solve complex, software-oriented tasks. It has a scope of 9 ECTS and is offered for up to 6 groups of up to 6 students each. The expected distribution of work is shown in Figure 6. The course aims to provide a two-level knowledge gain for students: 2) Students are taught high-level methodological aspects in two domains: intralogistics and systems engineering. 1) It conveys practical problem solving skills as well as soft skills for interdisciplinary group tasks in software engineering applied to a logistics-motivated robotics task. This level also fundamentalizes the theoretical aspects taught in level 1).

These levels are connected through an industry-oriented automation workshop where students need to design, implement and test a software system for a small-scale model warehouse.

The workshop employs the concepts of cyber-physical systems through tight integration of real-world hardware into digital networks and integration of simulation into the system design and deployment.

Students solve this task in groups of five to six, splitting subcomponents ranging over different abstraction levels such as control of robotics hardware (articulated robots, conveyors, and autonomous mobile robots) to order management. This task facilitates direct application of the abstract concepts taught in level 1). Level 2) is incorporated via the inherent interdisciplinary task itself as well as the teamwork required to solve it.

Content: The course is split into two components – corresponding to the two levels of learning objectives – a lecture block in the first half of the semester and a workshop that starts in the first half of the semester and ends with a demonstration at the end of the semester (see Figure 8).

The lecture component is organized into six learning blocks, each with multiple teacher-centered and student-centered components. Teachers first provide theoretical inputs, which are then directly reinforced during the lecture in short exercise sessions and through presentation of the exercise results to the plenum. Three learning blocks are dedicated to systems engineering for mechatronic systems, introducing the students to important concepts like requirements engineering, testing and validation, and software architectures. Three learning blocks are focused on fundamentals of modeling and describing intralogistics systems. First, we teach the Modular Material Handling meta model [23], which provides a rigorous model to describe, connect, and control material handling modules. The students can directly apply the learned model to standardize software interfaces of different modules in the warehouse automation task. Furthermore, the model can be used to flexibly plan and optimize material transport tasks. Students also learn important methods for planning and optimizing material flow systems. This aspect is also applied in the workshop as students need to characterize the performance of different material handling modules and optimize the overall system performance.

DOI: 10.2195/lj_proc_enke_en_202510_01

After the six lecture blocks, the learning progress is evaluated using a written exam at semester half-time.

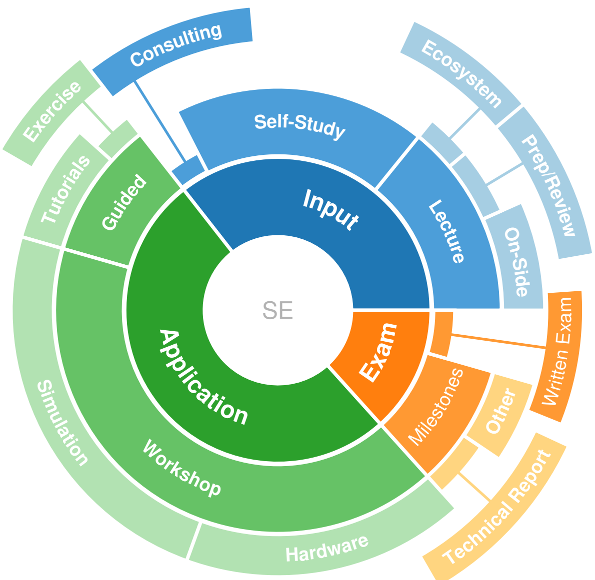

Figure 6: Approximate distribution of the workload for SE (Input: 35%, Application 50%, Exam: 15%).

The second component, the workshop, starts alongside the lecture blocks with a technical deepdive workshop and interlocks thematically with them. The workshop is student-centered and provides learners with the possibility to gain skills beyond the theoretical aspects of the course via project-based learning. Students are guided via a shared online forum, where they can ask and answer questions, and regular tutor consultation hours.

The warehouse automation task aligns with problems occurring in real-world intralogistics systems. We provide the students with a model warehouse setup, as depicted in 7. The setup includes different ready-to-use hardware modules and a ROS code base which facilitates interfacing with the hardware. Furthermore, the students have access to a ROS/Gazebo simulation. The students' task is to then design and implement the software to connect these modules and facilitate material flow through the warehouse.

Students solve the same workshop task in groups (uniform task, mixed individual and group work). As the course is graded individually, concrete subtasks need to be assigned to individuals. For this, we require the specification of roles inside the teams, so that each member takes ownership of a specific component of the system. These components are selected by teachers beforehand and hence provide a technical structure to solve the task. Furthermore, we provide guidance in form of the design process model. Here, students are first tasked to solve the problem inside the simulation environment, where the transfer challenge is medium. Groups

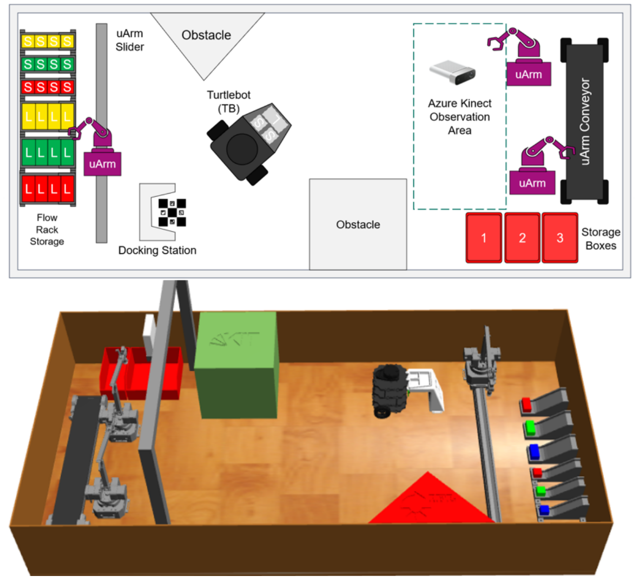

Figure 7: Concept and simulation environment of an internal logistics system for the SE course.

then transfer their software system to the real-hardware (with encouraged iterations back to the simulation environment). This process roughly follows the V-model.

However, besides the ownership definitions,the software ecosystem, and the two-tier design approach (first simulation, then real hardware) no further technical structure is enforced (medium technical structure).

The workshop provides a high organizational structure to guide the students while fostering technical creativity. The organizational structure is aligned with modern project management concepts like Agile. This structure aims to facilitate direct transfer and skill gains relevant for application in real-world engineering projects. The workshop instructors act as stakeholders in this structure. Via this role, we inject an external scaffold which helps students to plan and structure their project. The stakeholders convey this structure via two channels. First, students are provided with a list of system-level requirements which need to be fulfilled. Second, the project evolution is controlled using three milestones with associated expectations. During the first milestone, groups need to provide a written design specification of their system alongside the associated roles for each team member. The second milestone evaluates the system design applied inside the simulation environment. Again, groups hand-in a technical documentation with regards to the performance of their system. In addition, their codebased is analyzed for quality, test coverage, and other metrics via a continuous integration pipeline, which is transparently shared with the students and also acts as a continuous self-check of project progress. The same evaluation is car-

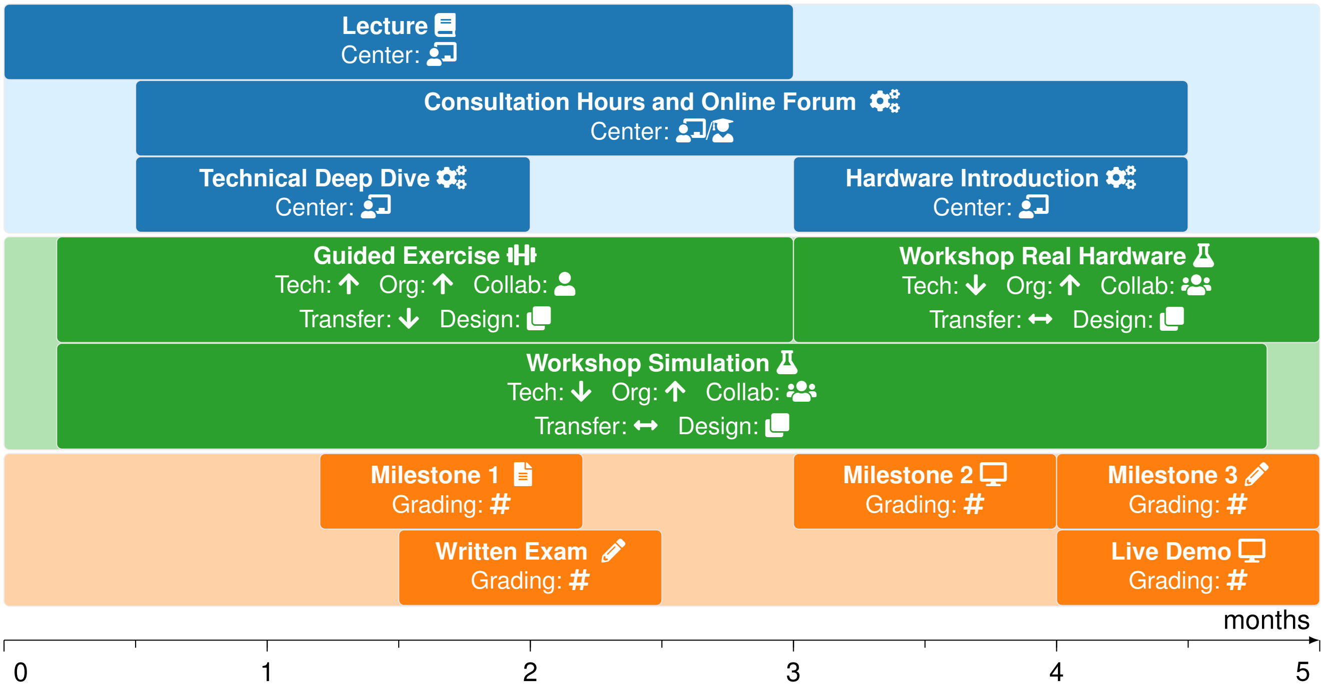

Figure 8: Schedule and logical sequence of building blocks of the Seamless Engineering course with Input (blue), Application (green), and Examination (orange). Characteristics according to Table 1.

ried out for the real hardware in the last milestone, which also incorporates a live demonstration of the students' solutions. The demonstration provides a comparative evaluation, as all groups are evaluated based on their systems overall performance. The demonstration is further used as a closing event, retrospective, and feedback opportunity for the students and teachers. To provide a further individual evaluation, students are required to each showcase a part of their work in form of a guided presentation. These presentations are part of milestones two and three.

Evaluation: Most students are at the end of their master's degree when taking the lecture. The overall feedback from students was mostly positive for each of the four years this course has been offered. While students recall the workload as high, it is still within an adequate level for the number of credits. Most importantly, student motivation for the course is high, as they understand and report the importance of the course for further studies and the transition into industry. They also report that they learned a lot during the course, both theoretical and practical aspects. Further student feedback positively addresses the interaction mechanisms between students, tutors and teachers. The availability of an online forum fosters a low-threshold student-to-student and student-to-tutor knowledge exchange. Similar observations were made during the tutor consultation hours and lectures. Overall, students felt that their questions and concerns were very strongly addressed by lecturers.

Besides the positive feedback, students also highlighted some key challenges with this type of course. Interconnections between the theoretical lectures and the practical workshop were recognized but underrepresented within the workload required during the workshop. This relates to the high effort students need to invest to familiarize themselves with the software and hardware ecosystem, especially if the hardware is not easy-to-use and robust. This effect occurred more often in the earlier instances of the course, where hardware needed to be adapted and improved, requiring patience and collaboration with the students. Similarly, the instructor's practical input given during the tech deep dive had to be condensed which resulted in a steep learning curve.

Another challenge is individual grading in combination with group work. While clear individuality is provided by the grading instruments exam and presentation, a significant share of the knowledge gains happens within the system design and implementation. Here, students felt that in heterogeneous teams, inequalities in participation, workload and attendance sometimes did not reflect in the overall grade.

From the instructors' perspective, the two-level the interconnection between the theoretical and practical components of the course resulted in better overall learning performance and more sustainable skill gains. This could be observed via the amount of students who later joined the institute for theses and as student assistants. Here, a significant increase in overall problem solving and application of course contents was observed. This positive feedback loop provides benefits apart from the core teaching objective and reinforces the motivation to provide such high-effort courses.

4.3 Industrial Mobile Robotics Lab

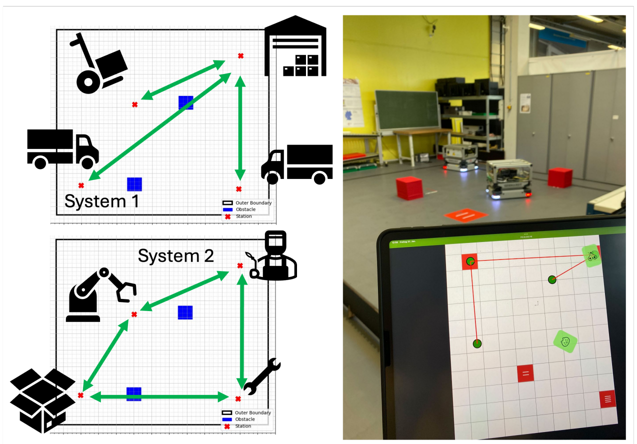

Figure 9: Concept and laboratory environment of the IMRL course. Mobile robot positions and VDA5050 transport orders being visualized in real-time.

The course Industrial Mobile Robotics Lab (IMRL) aims to give students first experiences in the field of mobile robotics within an intralogistics context. Students are given the task to develop a simplified and small-scale version of an intralogistics system made up of two industry-grade mobile robots and a central fleet management system that orchestrates the robots to perform transport tasks. Being designed for master's students, the course poses a great technical as well as organizational challenge as students are to develop a functioning and collaborating systems with several interfaces in a mostly self-organized manner.

The students are taught the basics of control, navigation and fleet management of mobile robots and most importantly the fundamentals of the VDA5050, a widely used industry recommendation interface for the communication between mobile robots and a fleet manager.

Firstly working individually with a simulation model, the students are then put into group developing either the control of a mobile robot or a master control system orchestrating the two robots.

Content: The course extends over a period of roughly two months and is organized around three milestones that structure the learning process and serve as formative performance checks. Because of the intensive supervision required and the limited availability of hardware, participation is restricted to twelve students admitted through a competitive application process.

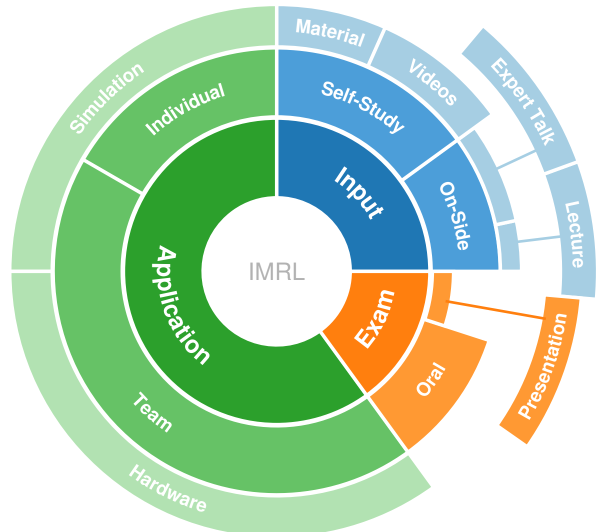

Figure 10: Approximate distribution of the workload for IMRL (Input: 25%, Application 60%, Exam: 15%).

The course begins with an introductory session outlining objectives, organization, and expectations. Students then enter a self-study phase, where video lectures provide an overview of the mobile robotics industry and the theoretical foundations needed for the practical component. This puts the student at the center of teaching as all materials are available from the outset and the students are free to structure their learning individually.

The next stage introduces the application phase, which focus on two key aspects: robot motion control and fleet management. In the individual work phase, each student is required to complete initial exercises in both domains within a simulation environment. Here a highly organizational structure in the form of code skeletons and pre-built code modules are provided to guide development, but students are encouraged to adapt, expand, or rewrite them as needed. In addition, tasks are described to guide the students through their first tasks in the domain. The sequence and depth of these exercises remain up to the students, giving room for autonomy while maintaining a clear framework. This phase concludes with the first milestone, a 30-minute individual evaluation. As the course is pass/fail, the assessment is not based on fixed criteria but verifies engagement with the material and provides a basis for the fair formation of teams in the next phase.

After, teams of three are formed, with two teams dedicated to robot control and two to fleet management. The group work builds directly on the outcomes of the individual tasks, ensuring continuity and progression. With each student meant to work on both technical aspects as well as presentations and interface definition, the student self-assign themselves the responsibility for one of these three aspects. First the groups continue working in the simulation environment tackling more complex tasks. Soon after the students are introduced to the hardware. From then on, the students have continuous access to the laboratory environment during regular working hours, allowing them to iteratively test and refine their code on real robotic platforms.

Two further milestones mark the group phase. After the first group work phase, the teams present their design concepts during the second milestone, receiving feedback from peers and instructors to refine their approaches. After the final stretch of implementation the third and final milestone requires a demonstration of the implemented systems: the robot control teams must show how precise the robot can follow trajectories under their control, while the fleet management teams demonstrate the correct allocation of transport tasks to multiple robots. The detailed time table of the IMRL can be seen in Figure 11.

To support learning, students benefit from regular consultation hours, an actively monitored online forum, and the on-site presence of instructors during the group work phase for immediate guidance. Moreover, expert lectures on mobile robotics are integrated at the beginning of the course and in connection with the second and third milestones, linking the practical work with current research and industrial applications.

Evaluation: One of the main challenges in designing the course was to identify a suitable time frame that would neither overlap with the examination period nor require students to invest excessive time during the lecture period, allowing them to keep pace with their other courses. To balance these considerations, while still providing sufficient time for students to work through the assignments, the course was scheduled as an intensive one-month block. Figure 10 illustrates the distribution of the allocated time across the main building blocks of the IMRL. Students consistently confirmed the considerable workload and time investment, but most did not perceive this as negative. On the contrary, reported motivation remained high throughout the course. This can be attributed both to the selective admission process, which favors highly motivated participants, and to the close supervision provided by the teaching team. Frequent consultation hours, an actively monitored forum, and permanent access to the lab during the group phase gave students the sense that their efforts were supported and that they were engaged in a relevant and meaningful course.

The course's industry-oriented and technically complex assignment was another source of motivation. Students highlighted that they expanded their skills in programming, working with robotic systems, transferring solutions from simulation to hardware, and using MQTT and standardized communication interfaces such as VDA 5050. They also emphasized the value of teamwork: dividing tasks within their group, jointly refining interface definitions, and integrating different solutions into a common system. Many participants explicitly noted that this collaboration mirrored challenges they expect to encounter in their future careers.

The integration of industry talks by invited experts was particularly well received. These sessions not only provided insights into the current state of mobile robotics but also underscored the industrial relevance of the students' own work. The strong connection between the lab assignments and real-world applications gave participants the impression that they were contributing to authentic engineering problems rather than abstract exercises.

Another highly valued element was the combination of individual and group work. During the individual phase, students were able to explore the topics at their own pace and engage with both motion control and fleet management tasks to the extent they felt necessary. This allowed for individual learning paths and accommodated differences in technical competence and interest. In the subsequent group phase, these differences became an asset: students learned to combine diverse solution approaches into a coherent system. The structured milestones reinforced this process by requiring students to present and defend their work to a technical audience and to incorporate feedback from instructors and peers.

Overall, students reported that the course better prepared them for the professional environment of an engineer. Working on a complex technical task within a short time frame, organizing themselves in interdisciplinary teams, and deploying solutions on industry-grade hardware gave them valuable experience for future endeavors. The examination format was also well received. Individual evaluation sessions allowed students to present their work at their own pace, while the milestone presentations provided a platform for pitching their concepts and demonstrating progress in a professional setting.

Finally, the course structure—starting with simulation-based development, followed by adaptation to real-world hardware—exposed students to the realities of engineering practice. They encountered and learned to manage practical challenges such as sensor noise, hardware imperfections, interface inconsistencies, and differing interpretations of task requirements. This direct confrontation with real-world complexity was consistently highlighted as one of the most valuable learning experiences of the course.

DOI: 10.2195/lj_proc_enke_en_202510_01

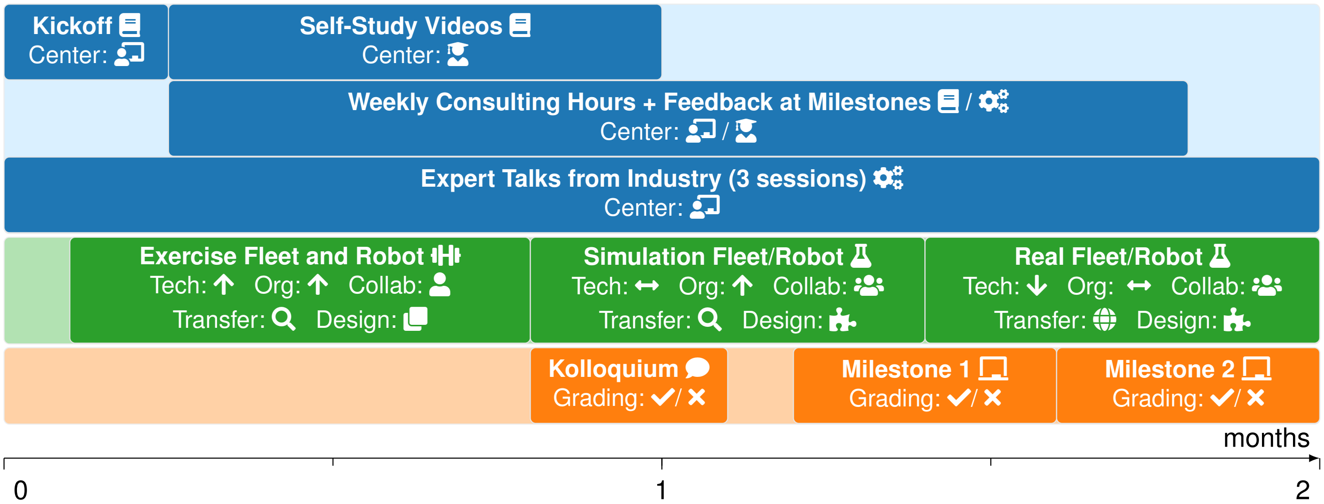

Figure 11: Schedule and logical sequence of building blocks of the Industrial Mobile Robotics Lab with Input (blue), Application (green), and Exams (orange). Characteristics according to Table 1.

4.4 Further fields of application oriented learning

4.4.1 Production Techniques Laboratory

The Production Techniques Laboratory (PTL) provides students with hands-on insights into production engineering through a series of weekly experiments that combine theoretical preparation with immediate practical application.

Content: PTL is jointly organized by several institutes at KIT and comprises 12 experiments across logistics, supply chain, manufacturing, and human factors. Students prepare with provided course materials and are evaluated in short pass/fail colloquium at the end of each experiment, of which at least 11 must be passed. Our institute contributes three laboratories on intralogistics: *Storage and Order Picking Technology*, focusing on throughput and inventory; *Identification Systems*, covering barcode and RFID technologies; and *Industry 4.0 Material Flow*, addressing fleet management of mobile robots. There is a strong focus on practical application, with students performing order picking themselves, examining various barcode and RFID technologies, or simulating the function of a fleet of mobile robots by simulating a guidance system and mobile robots on a playing field.

Evaluation: More than half of the participants are early in their studies, yet feedback is consistently positive. Students highlighted the value of comprehensive materials, close tutor support, and the clarity gained from combining theory with practice. Instructors view the integration as effective for heterogeneous backgrounds, though balancing theoretical input with practical application remains a challenge.

4.4.2 Theses

Thesis work represents the final step in training to become an engineer in which students consolidate and demonstrate their engineering knowledge. The process requires them to formulate a research question, select an appropriate methodology, execute it in a structured manner, and communicate the results scientifically. Beyond mastering disciplinary content, students practice critical thinking, project management, and academic writing. These abilities are directly transferable to professional engineering practice and to subsequent academic research, making the thesis both a demonstration of academic maturity and a preparation for lifelong learning.

Content: Thesis work at the institute follows a structured yet student-centered process. Preparation begins with a mandatory workshop on scientific research and writing, in which students revisit the foundations of academic work. The workshop clarifies expectations regarding structure, style, and ethical standards, such as correct citation practices, and introduces relevant tools for literature management and text production. This common orientation lowers entry barriers, establishes a shared understanding of academic standards, and ensures that assessment criteria are transparent from the outset.

At the beginning of the thesis, students write a concise research exposee that defines the problem statement, specifies objectives and expected outcomes. This serves to promote early, in-depth engagement with the research topic and to prompt students to articulate the full conceptual trajectory of their work from the beginning. A preliminary chapter outline is submitted in parallel to secure a coherent logical structure and to guide subsequent research and writing phases. During the writing phase, weekly student-initiated supervision meetings provide opportunities for questions and feedback. This continuous interaction mitigates isolation and supports an iterative, student-driven refinement of both the argumentation and the research design. Final evaluation comprises two components, the written thesis and an oral presentation of results, with primary weight placed on the written document. Assessment criteria, communicated transparently at the outset of the thesis process, guide both components to ensure fairness and alignment with the intended learning outcomes.

Evaluation: Student feedback underscores the importance of early clarity and continuous supervision. The preparatory workshop reduces entry barriers and clarifies academic expectations, while the exposee encourages thorough planning and a well-defined methodological approach. Weekly meetings are considered essential for maintaining progress and addressing conceptual or technical challenges in a timely manner. The provision of evaluation criteria is seen as a way to a fair and transparent evaluation.

Supervisors report that the structured framework enhances thesis quality and comparability, supports fair and transparent assessment. Transparent criteria and regular dialogue also help balance autonomy with guidance, enabling students to develop self-regulation and a strong sense of ownership over their work.

5 Discussion

The case studies highlight how different teaching formats for application-oriented learning balance input, application, and examination while addressing varying levels of student autonomy and course structure. In this chapter, we discuss the interplay of course design characteristics with teaching outcomes, the role of preparation and mentoring in practice-oriented learning, and key lessons for practitioners aiming to implement similar formats.

5.1 Interplay between the characteristics of the building blocks and teaching success

Compared to the exemplary, classic course presented in chapter 3, the focus of the presented courses lies on applications. This can be seen in the distribution of the expected work, where applications account for more than 50% of the required time investment by the students. The following section examines how the various components affected the evaluation of the course by students and instructors.

Across the courses, a recurring observation is that a high degree of freedom in the input phase requires a correspondingly strong structure in knowledge assessment, preferably using continuous assessments to keep students engaged. In the IMRL course, some students reported that they underestimated the importance of the self-study videos and therefore did not engage with the content until later, when the relevance became more apparent whilst working with real systems. This delay led to avoidable uncertainties and repeated questions. The colloquium partly compensated for this effect by focusing questions on important topics. This underlines that self-study formats are effective only when combined with structured assessments that request students to address the material in due time.

By contrast, in courses such as PnP and Seamless Engineering, essential concepts are presented at the beginning of the lecture and clarified immediately through interaction. This direct focus on relevant content motivates students, as the practical applicability is visible from the outset. In SE, the written mid-semester exam ensured a common knowledge base for all students before entering later stages of teamwork. Over all courses it could be observed, that when theoretical input is streamlined to core concepts and complemented with references for further study, students report that they can concentrate more effectively and see the purpose of their learning. Conversely, when input is too broad or insufficiently structured, both students and instructors emphasize the need for stronger scaffolding through assignments and transparent assessment criteria.

At the same time, students requested more structure in terms of example code, practical demonstrations, and clearer guidance on complex tools such as Python (concept of object-oriented coding), Git, and MQTT. These findings suggest that practical freedom is appreciated, but requires careful instructional design to avoid overburdening students with peripheral challenges.

A further recurring challenge lies in the heterogeneous composition of student teams, especially regarding prior knowledge and willingness to contribute. While interdisciplinary diversity is valuable for authentic project settings, unequal distributions of technical expertise and motivation can lead to imbalances in learning opportunities and workload. This issue points to the importance of deliberate team formation strategies and structured mentoring to ensure that the benefits of collaborative learning are realized without creating systematic disadvantages.

In the PnP and the beginning of the IMRL course, the instructors noticed that some students had difficulty distributing work fairly. For this reason, the instructors presented team roles like Coding-Supervisor, Communication-Leader and Team-Manager in various courses, which the students were expected to assign themselves. This increased level of organizational structure led to positive results, as each team member took responsibility for a specific area. It is important to emphasize that responsibilities are only assigned for areas such as coding, which does not mean that all the code is written by one person, but only that this person must distribute tasks accordingly. This structure reaches its limits when the skills within the team are too diverse and, for example, one team member lacks coding skills completely, making it difficult to distribute tasks fairly. In this case, either increased supervision must be provided or admission requirements must be implemented. These can come in the form of more in-detail required application information or even in-person exercises to verify certain capabilities.

If students only have to pass the course it can lead to lower motivation, as they (especially highly-motivated students) often prefer to invest their time in courses that are graded. At the same time this can encourage team work, since there is no competition and pressure within the teams. Individual assessment is difficult in practical courses, especially in teamwork. An example on how to deal with this is shown in Seamless Engineering where individual tasks are introduced. Also, oral colloquia, as done for the IMRL, can be a way to achieve individual assessment within a primarily group-based course.

5.2 Learnings and Practices for Practitioners

The preparation and execution of interactive, practice-oriented teaching formats is highly demanding and requires careful attention to both technical and organizational aspects. From the field studies, several recurring practices and lessons learned can be derived that may support practitioners when designing and conducting such courses.

Hardware preparation should not be underestimated. Experience shows that most student questions and difficulties occur at the very beginning of the course, when systems are first encountered. A thorough pre-test of the entire setup by the instructors, combined with a hands-on introduction and opportunities for immediate live testing, allows students to gain familiarity and helps to resolve initial uncertainties early. Continuous technical support during the course remains essential, as unforeseen failures of hardware components are difficult rule out completely.

Software preparation is equally critical. Systems should be tested not only in standard configurations but also in more complex scenarios in order to eliminate bugs and bottlenecks before they appear in the classroom. To lower the entry barrier for students, software should be intuitive to install and restricted to the relevant functionalities needed for the task. Providing comprehensive examples in a running state ensures that all students can start from a working baseline rather than struggling with setup problems as it can decrease motivation at an early stage. At the same time, reliance on locally installed software brings challenges: installation on heterogeneous systems is nearly impossible without substantial support. A pragmatic solution is to rely on a uniform operating system environment (e.g., Linux), or to distribute pre-installed environments on bootable USB-drives. While this approach has proven successful, it requires that student hardware supports USB booting. Alternatives such as WSL2, virtual machines, centrally hosted systems, or containerized solutions (e.g., Docker) offer potential but introduce new complications, particularly with respect to communication performance and support effort.

The maintenance and updating of course content is an ongoing requirement. Course materials and software environments should be updated immediately after the end of each course to preserve relevance and to incorporate improvements based on feedback. This is particularly demanding when relying on evolving software packages, as the transition between major versions (e.g., from ROS to ROS 2) can significantly increase preparation effort. Nevertheless, maintaining state-of-the-art content is essential for both credibility and learning effectiveness.

Task preparation benefits from a fine-grained and clearly structured design. At the beginning of the course, tasks should leave students little room for ambiguity, as excessive freedom can lead to frustration or misaligned priorities. Defining milestones and providing intermediate deliverables has been shown to help students remain on track. As students gain confidence, more freedom can be introduced progressively. At the same time, team conflicts are almost inevitable in group projects. This requires instructors to demonstrate soft skills in conflict mediation and facilitation, as team dynamics strongly influence learning outcomes.

Team assignment remains a major challenge for application-oriented courses. Various approaches can be chosen here. A random team assignment approach can be fair, but carries the risk of creating unequal teams due to a lack of skills. Purely self-assigned teams can lead to exclusion and also to unequal teams. Self-assessment by students is a good approach especially for short courses. Albeit, it can be distorted by students underestimating or overestimating themselves, at the same time self-reflection in encouraged. The best results were achieved through targeted colloquiums and subsequent assignment based on the assessment of the instructors. However, this is time-consuming and may be perceived as unfair by the students.

5.3 Challenges and Future Directions

A high workload is seen as the biggest challenge of application-oriented learning, by both students and instructors. However, even though the students in the example courses consistently described the workload as high, they also recognized its necessity and appropriateness for the level of learning achieved. More than two-thirds of the participants said they took the course for personal interest. Students emphasized the course's perceived relevance to their studies and the substantial learning progress they achieved. For IMRL the students also assessed their motivation retrospective after the course. Motivation was reported to be highest during practical phases involving robotic hardware, when students had the opportunity to apply their knowledge directly. The situation becomes critical when important successes fail to materialize. In PnP, supervisors provide close support and observe that frustration sets in halfway through the course as students perceive the system as very complex. During this period, it is crucial to offer assistance to promptly resolve any hardware or software issues and maintain student engagement. In the past, motivation has always been regained after this challenging period, and participants have worked on the topics with even more enthusiasm.

The temporal design strongly affects student learning and motivation, posing a major challenge for instructors. Very short formats, such as the one- to two-week PnP lab, require intensive engagement and leave little room for students to overcome initial difficulties or consolidate knowledge if they fall behind. On the other hand, extending the duration of courses, as in the case of the IMRL change from one month to two, results in phases of reduced focus and delayed engagement. Based on student feedback and instructor observations, simply increasing the available time does not guarantee improved outcomes. A balanced course structure that combines compactness with clearly defined checkpoints appears to be the most effective approach. Providing preparatory material in advance can support students, but experience indicates that more than one week of lead time is rarely used effectively. Milestones placed one-quarter or one-third of the way through the course help sustain motivation, provide opportunities for feedback, and allow instructors to take timely corrective action.

Finally, changing teaching is requirements due to the rise of AI. With the increasing availability of generative AI tools such as ChatGPT we observed a transformed way students approach programming and problem-solving tasks. In particular, coding assignments with clearly defined interfaces and processes can often be solved effectively with AI assistance. This development requires new teaching methods that both challenge students beyond what AI can deliver and deliberately integrate the use of such tools into course structures. Simply prohibiting AI use is neither enforceable nor reflective of the realities of modern engineering practice.

Experiences from the field studies illustrate different strategies to address this challenge. In the PnP Lab, a creative component was introduced in which students had to design and build their own robot, ensuring that physical implementation and design decisions could not be outsourced to AI. In IMRL, emphasis was placed on inter-team communication: students had to negotiate interfaces and adapt their code to the solutions of other groups, a process that requires mutual understanding and cannot be automated. Similarly, in SE and thesis work, oral presentations and subsequent questioning provided opportunities to test students' depth of understanding and to detect excessive use of AI without understanding.

Looking ahead, it will be important for practical courses to strengthen students' AI literacy and design tasks that surpass the current capabilities of AI. Effective strategies include evaluating alternative technical solutions for presented problems, critically questioning generated outputs, and physically building and testing prototypes.

6 Conclusion

This paper has presented and analyzed three application-oriented courses using a classification framework along the dimensions of input, application, and examination. Beyond describing individual cases, the contribution lies in demonstrating how this framework supports systematic evaluation of course design and teaching success. For practitioners, it provides a structured view to reflect on their own courses and to identify critical points of intervention.

Across the courses, a recurring principle emerges: freedom in learning must be balanced with structure in assessment and task design. Providing too much openness at the beginning often leads to unfocused effort, while well-defined milestones, transparent requirements, and supportive team structures foster both motivation and achievement. At the same time, high workload remains a central challenge. A time frame that is too short hinders deep learning, whereas overly long formats risk a loss of momentum. Milestone meetings, functioning infrastructure, and constructive feedback loops are effective to maintain motivation and ensure timely progress.

For instructors, the orchestration of technical infrastructure, software environments, and team setup remains demanding and resource-intensive. There is no single solution, but experiences show that lowering technical entry barriers, offering structured yet flexible tasks, and adopting transparent approaches to team formation are key to both learning effectiveness and student satisfaction. Maintaining course content at the state of the art, for example during the transition from ROS to ROS 2, further underscores the long-term investment required to keep such formats sustainable.

Looking forward, one of the most pressing challenges will be the role of artificial intelligence. Students already rely heavily on AI tools, particularly for programming, which can accelerate progress but also reduce engagement with the underlying concepts. Rather than ignoring this trend, future teaching formats must actively integrate AI into course design and assessment, ensuring that tools support rather than replace deep learning.

Application-oriented learning demonstrates its potential to inspire students, to connect theoretical knowledge with practice, and to strengthen teamwork and problem-solving skills that are crucial for engineering education. The experiences presented here highlight that, despite the high preparation effort, such courses generate substantial benefits for students and teachers alike. Practitioners and researchers are therefore encouraged to expand and refine these approaches to meet the evolving demands of modern engineering education.

References

[1] C. Enke, J.-F. Klein, M. Sperling, B. Zhou, and K. Furmans, “Development of an experimental environment to study the challenges in cyber-physical intralogistics systems,” *Logistics Journal: Proceedings*, no. 18, 2022.

[2] M. Prince, “Does active learning work? A review of the research,” *Journal of Engineering Education*, vol. 93, no. 3, pp. 223–231, 2004.

[3] J. Biggs and C. Tang, *Teaching for quality learning at university: what the student does*, 4th ed. Maidenhead: Open University Press, 2011.

[4] A. King, “From sage on the stage to guide on the side,” *College Teaching*, vol. 41, no. 1, pp. 30–35, 1993.

[5] J. Bergmann and A. Sams, *Flip your classroom: reach every student in every class every day*, 1st ed. Washington DC: International Society for Technology in Education, 2012.

[6] H. S. Barrows, “A taxonomy of problem-based learning methods,” *Medical Education*, vol. 20, no. 6, pp. 481–486, 1986.

[7] J. W. Thomas, “A review on research on project-based learning,” 2000.

[8] D. A. Kolb, *Experimental learning: experience as the source of learning and development*. Englewood Cliffs, N.J: Prentice-Hall, 1984.

[9] D. S. Domain, “A review of laboratory instruction styles,” *Journal of Chemical Education*, vol. 76, pp. 543–547, 1999.

[10] A. Gorbunova, J. J. G. Van Merrienboer, and J. Costley, “Are inductive teaching methods compatible with cognitive load theory?” *Educational Psychology Review*, vol. 35, p. 111, 2023.

[11] J. Bouwma-Gearhart, Y. H. Choi, C. A. Lenhart, I. Villanueva, L. S. Nadelson, and E. Soto, “Undergraduate students becoming engineers: the affordances of university-based makerspaces,” *Sustainability*, vol. 13, no. 4, p. 1670, 2021.

[12] H. Taş and M. B. Minaz, “The effects of learning style-based differentiated instructional activities on academic achievement and learning retention in the social studies course,” *Sage Open*, vol. 14, no. 2, 2024.

[13] R. Ghannam and C. Chan, “Teaching undergraduate students to think like real-world systems engineers: a technology-based hybrid learning approach,” *Systems Engineering*, vol. 26, no. 6, pp. 728–741, 2023.

[14] D. W. Johnson, R. T. Johnson, and K. A. Smith, “Cooperative learning: improving university instruction by basing practice on validated theory,” *Journal on Excellence in College Teaching*, vol. 25, no. 3-4, pp. 85–118, 2014.

[15] M. Laal, L. Geranpaye, and M. Daemi, “Individual accountability in collaborative learning,” *Procedia - Social and Behavioral Sciences*, vol. 93, pp. 286–289, 2013.

[16] S. M. Barnett and S. J. Ceci, “When and where do we apply what we learn? A taxonomy for far transfer.” *Psychological Bulletin*, vol. 128, no. 4, pp. 612–637, 2002.

[17] S. D. Blum, *Ungrading: why rating students undermines learning (and what to do instead)*, 1st ed., ser. Teaching and learning in higher education. Morgantown: West Virginia University Press, 2020.

[18] N. Falchikov, *Improving assessment through student involvement: practical solutions for aiding learning in higher and further education*. New York: Routledge, 2005.

[19] S. French, A. Dickerson, and R. A. Mulder, “A review of the benefits and drawbacks of high-stakes final examinations in higher education,” *Higher Education*, vol. 88, no. 3, pp. 893–918, 2024.

[20] A. Fenton, “Reconsidering the use of oral exams and assessments: an old way to move into a new future,” *Educational Researcher*, 2025.

[21] J. Cain, M. Medina, F. Romanelli, and A. Persky, “Deficiencies of traditional grading systems and recommendations for the future,” *American Journal of Pharmaceutical Education*, vol. 86, no. 7, p. 8850, 2022.

[22] T. Paloposki, V. Virtanen, and M. Clavert, “From a final exam to continuous assessment on a large Bachelor level engineering course,” *European Journal of Engineering Education*, pp. 1–14, 2024.

[23] K. Furmans and K. R. Gue, “A framework for modeling material handling with decentralized control,” *15th IMHRC Proceedings (Savannah, Georgia. USA – 2018)*, p. 8, 2018.