Managing projects in IT – an example-based course

MASTER’S THESIS

Bc. Dita Salajková

Brno, 2019
Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during the elaboration of this work are properly cited and listed in complete reference to the due source.

Advisor: RNDr. Jaroslav Ráček, Ph.D.
Acknowledgement

I would like to thank the advisor for his support, professional advice and most of all for his true mentorship through which he has encouraged me to believe in my abilities.
Abstract

This thesis aims to create a course that introduces essential theoretical grounds as well as practical examples of project management in IT. The course is founded on widely used standards of both project management and software development. It focuses on the way these standards can be combined in order to select an approach that is fitting to the project’s initial settings. The example-based part of the course is founded on two project cases. The first project is managed using an agile approach, the second one using a predictive approach. In each of these two examples, a project’s lifecycle is followed, presenting managerial techniques that comply with the selected type of management.
Keywords

IT project management, software development, example-based learning, PRINCE2, PMBOK, IPMA, Unified Process, Scrum
# Contents

1 Introduction .................................................................................................................. 1

2 Motivation and requirements ...................................................................................... 2

3 Project management in IT ............................................................................................ 4
   3.1 General project management and its standards ....................................................... 4
   3.2 Project management in IT and standards of SW development ............................. 12
   3.3 Selecting management approach ......................................................................... 15

4 Project management courses at Czech universities with IT specialization ............ 19
   4.1 Selected courses .................................................................................................... 19
   4.2 Evaluating method ................................................................................................. 23
   4.3 Results and conclusions ......................................................................................... 24

5 Example-based learning ............................................................................................... 26
   5.1 Types of example-based learning .......................................................................... 26
   5.2 Recommendations for designing the examples .................................................... 27

6 The developed course .................................................................................................. 29
   6.1 Course’s form ......................................................................................................... 29
   6.2 Course’s contents .................................................................................................. 40

7 Discussion ..................................................................................................................... 55

8 Conclusion ..................................................................................................................... 58

Bibliography .................................................................................................................... 59

Electronic attachments ..................................................................................................... 62
1 Introduction

Projects are at the heart of each business’ activities. In a limited period, projects aim to create a unique product that will bring value to its stakeholders. In order to succeed, projects need to be suitably managed to stay within the agreed borders of time, budget and scope. IT-based projects and software development projects, in particular, are in a great risk of failure [1], often as a result of erroneous management. A well-managed IT project is based on applying the right methods, techniques, tools and competencies the choice of which depends on the approach selected. Each project needs to be approached in a way that is suitable for its initial settings. For instance, a communication strategy within a large and diverse team cannot be based on face-to-face interactions, or delivering a fast-time-to-market product cannot be planned in an exhaustive and time-consuming way.

My goal within this thesis is to create a course that teaches students the importance of selecting an appropriate managerial approach, as well as the methods, techniques, tools and competencies that such approach requires. The second chapter of this thesis presents the motivation for such course, together with the requirements for its design. The third chapter introduces concepts, knowledge areas, principles and processes of IT project management, the knowledge of which was mostly derived from the widely used standards of project management and software development. It also shows how these standards can be combined in order to approach an IT project of certain initial settings successfully. Fourth chapter analyses the contents of other project management courses taught at Czech faculties with an IT specialization. The fifth chapter introduces example-based learning and the reasoning behind basing the course on two analogical project examples, followed by suggestions for the design of such examples. The sixth chapter describes the developed course, focusing on both how it is structured and what it contains. The seventh chapter reflects on the work done and discusses alternations and improvements.
2 Motivation and requirements

The outcome of this work is a course that will be taught at the Faculty of Informatics at Masaryk University in Brno, Czech Republic. Faculty of Informatics (FI) is an established institution that specializes in computer science education. The study programs here focus on both theoretical computer science and applied informatics. When studying applied informatics, students can specialize in Artificial intelligence, Visual informatics, Computer systems, communication and security, but also Software systems and services management\(^1\). Being able to manage a project is one of the essential competencies of the latter program’s graduate. This is where the need for project management course emerges.

The Project management course that is currently taught at FI requires renovation. According to its lecturer, it does not reflect current trends in project management. It is composed of mainly theory-based lectures, with some references to real-life project management and its techniques\(^2\). A significant focus of the course is on software quality measurement and testing. Software quality, however, has become a primary subject in other courses taught at FI, e.g. Software Quality\(^3\), Software Engineering I\(^4\) and Software Engineering II\(^5\). Over the last years, the course has become more popular, with 83, 103, 147, 120 and 121 students enrolled in the Spring semesters of 2015, 2016, 2017, 2018 and 2019, respectively. Despite this fact, the present allotted time for the course is two hours a week for approximately 12 weeks of the Spring semester. These two hours will be taught in the form of lectures as combining lectures with practical seminars for over 100 students in the allowed 2 hours is considered counter-productive. Practical learning, however, is one of the main requirements for the course design.

The lecturer wishes to introduce students to main theoretical grounds, as well as practical examples of IT project management, reflecting popular standards of pro-

---

\(^1\) More on programs can be found at https://www.fi.muni.cz/admission/.
\(^3\) Software Quality course syllabi: https://is.muni.cz/course/fi/PV260.
\(^4\) Software Engineering I course syllabi: https://is.muni.cz/course/fi/PB007.
\(^5\) Software Engineering II course syllabi: https://is.muni.cz/auth/fi/PA017.
ject management and software development. In consistency with the course’s current emphasis on managing the development of software within a project, the combination of project management and software development standards will be the leitmotiv and a basis for understanding IT project manager’s day to day business. As the course is lecture-based, the required practical examples will be included in the lecture. However, a suggestion for practical seminars consistent with the lecture’s content will also be a part of the outcome. This is in case the allotted time for the course changes in the future. The form and structure of lectures and related study materials will be based on pedagogics’ best-practices. As the course is attended by international students, the study materials will be created in English. Table 2.1 structures and summarizes the requirements.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The course includes lectures on <em>theoretical grounds</em> of project management in IT, reflecting the contents of popular standards of project management and software development.</td>
</tr>
<tr>
<td>2.</td>
<td>The course includes <em>practical examples</em> of the project’s processes, tools and techniques, with a focus on implications of combining project management and software development methods.</td>
</tr>
<tr>
<td>3.</td>
<td>The course is lecture-based. Deliverables include study materials (presentations) in English. The form is based on pedagogical best-practices.</td>
</tr>
<tr>
<td>4.</td>
<td>The course includes an outline and recommendations for practical seminars’ content.</td>
</tr>
</tbody>
</table>
3 Project management in IT

This chapter is dedicated to the domain of IT project management. Through the contents of widely used standards, it introduces main knowledge areas, processes and principles that should be included in any course on project management. Because of the IT specialization of the course, two popular standards of software development are also reviewed. These represent two main approaches to software development and its lifecycle. The last part of this chapter focuses on the importance of choosing the right approach with regards to the initial settings of the project. It shows how the principles of standards of project management and software development can be combined in order to successfully manage such projects.

3.1 General project management and its standards

Projects are a way to temporarily put together the knowledge and means to deliver certain outcome. The International Project Management Association defines project as „unique, temporary, multidisciplinary and organized endeavor to realize agreed deliverables within predefined requirements and constraints [2]”. The temporariness of the project lies in its lifecycle, with typically pre-project, initiation, execution and close phases. For companies, a project is not business as usual. The combination of key factors (e.g. requirements, deliverables, customer, team) makes it unique. However, every project has repetitive elements called processes (e.g. creating a plan, analyzing risks, designating roles and responsibilities). A project is essentially a unique set of such processes that can be repeated and even automated in most projects. This is where the standards for project management come to play.

Whether it’s building a house, landing on a moon or developing an information system, each project can be managed by applying general knowledge of the processes that are common to all projects. This know-how can be found in standardized sources, typically called standards, methods, methodologies or frameworks. For the purpose of this thesis, all these standardized sources will be referred to as standards. They introduce their readers to terminology, knowledge areas, principles, processes
and even competencies required for successful project management. Moreover, adopting appropriate standard can contribute to better results, effective coordination of different activities, raised transparency and strengthened trust with stakeholders [3]. Most companies use standardized project management practices, either in some departments or throughout the entire organization [4]. Using a standard on project management means not having to invent the wheel.

In order to include a widely acceptable content on project management in the course, three popular project management standards have been used as a source: Projects in a Controlled Environment (PRINCE2)\(^6\), Project Management Institute’s Project Management Body of Knowledge (PMBOK)\(^7\) and International Project Management Association’s Individual Competence Baseline for Project, Programme & Portfolio Management (IPMA ICB)\(^8\). Each of these standards is specific in its content and approach to project management and should be adopted with respect to the project’s background and initial settings. The next of this chapter part briefly describes the scope of these standards, together with suggestions for its adoption.

**PRINCE2 [5]**

PRINCE2 (Projects in Controlled Environment) is a widely considered leading project management method [5]. It is mostly popular in the country of its origin – the UK. However, its popularity is increasing in other regions, especially Europe and Australia, with over 1 million PRINCE2 certification exams taken globally since its launch in 1996 [6]. When comparing industries, the highest adopters of PRINCE2 are IT-based projects [6].

---


\(^7\) [https://www.pmi.org/pmbok-guide-standards](https://www.pmi.org/pmbok-guide-standards)

\(^8\) [https://www.ipma.world/individuals/standard/](https://www.ipma.world/individuals/standard/)
PRINCE2 is *process-based* and employs a prescriptive approach, making it a step-by-step formula for project management. Its structure is based on seven principles, themes and processes (Figure 3.1).

The seven principles are guiding obligations and good practices. For instance, the principle “Defined roles and responsibilities” urges the manager to explicitly specify team structure, defining roles and responsibilities. The seven PRINCE2 principles are:

- Continued business justification,
- Learn from experience,
- Defined roles and responsibilities,
- Manage by stages,
- Manage by exception,
- Focus on products,
- Tailor to suit the project.
The seven themes in PRINCE2 are disciplines of the project that must be continuously addressed. For instance, the “Progress” theme directs the manager to constantly monitor and compare actual achievements against planned ones, and to refer any exceptions up next management level. The seven PRINCE2 themes are:

- Business case,
- Organization,
- Quality,
- Plans,
- Risk,
- Change,
- Progress.

The seven PRINCE2 processes map the progression along the project’s lifecycle. Each process provides a set of activities required for a successful project. For instance, the “Initiating a project” process prescribes assembling the Project Initialization Documentation that includes management strategies and project plan. The seven PRINCE2 processes are:

- Starting up a project,
- Directing a project,
- Initiating a project,
- Controlling a stage,
- Managing product delivery,
- Managing a stage boundary,
- Closing a project.

The prescriptive, process-based approach makes PRINCE2 a suitable methodology for „command and control” type of management of a large and diversified team. The larger the team, the more complex communication between its members becomes. PRINCE2’s regular reporting on the current project’s state and exception reports in case a set tolerance has been exceeded contribute to easier control of a large team. PRINCE2 is also suitable for projects where all activities need to be thoroughly documented.

Because of the prescriptive and procedural nature of PRINCE2, adopting other standards should be considered when bureaucratic overload is not welcome. Because of the reporting overload that following PRINCE2 method can generate, its
suitability is questionable for projects with unstable requirements and frequent change requests that need to be fulfilled effortlessly.

**PMBOK [7]**

The Project Management Association’s (PMI) Project Management Body of Knowledge (PMBOK) is an extensive, *process-based* guide to good practices in project management. Unlike PRINCE2, it is not considered a method or methodology. It is a foundation upon which a method can be built [5]. It originated in North America, where it is also most widely practiced [4].

PMBOK’s structure comprises of 49 processes grouped into five process groups and ten knowledge areas (Figure 3.2).

![Figure 3.2: The structure of PMBOK [7].](image-url)
Similarly to PRINCE2’s themes, PMBOK’s knowledge areas are disciplines of a project that have to be addressed constantly. The knowledge areas are:

- Integration,
- Scope,
- Schedule,
- Cost,
- Quality,
- Resource,
- Communications,
- Risk,
- Procurement,
- Stakeholder.

The five process groups follow the project’s lifecycle and are as follows:

- Initiating processes,
- Planning processes,
- Executing processes,
- Monitoring and controlling processes,
- Closing processes.

The five process groups are further broken into 49 processes (e.g. Plan schedule management, Develop schedule, Control schedule). Each process has clearly defined inputs, tools, techniques and outputs.

PMBOK is best used as a handbook or guidance for different knowledge areas, especially when looking for related tools and techniques. It is suitable for both expert and entry-level project managers.

IPMA ICB [2]
International Project Management Association’s Individual Competence Baseline (IPMA ICB) is a competence-based standard for managing a project.

Unlike the other two standards introduced, IPMA ICB it is not a how-to method for managing projects. It does not include step-by-step guidance on the project’s processes. Instead, it directs the manager to the required knowledge, skills and abilities.
he should have. Unlike the other introduced standards, it includes a thorough section about soft skills and their usage. Due to this approach, it can be used beside any other standard.

IPMA ICB’s structure comprises of three competence areas – perspective competencies, people competencies and practice competencies (Figure 3.3). These competence areas are further broken down into 28 competence elements that come with defined purpose, knowledge and skills required, and key competence indicators.

![Figure 3.3: The structure of IPMA ICB [2].](image)

The perspective competence area supports developing rationale for a project. It comprises of knowledge and skills required for successful interaction with the project environment. The perspective competence elements are:

- Strategy,
- Governance, structures and processes,
- Compliance, standards and regulation,
- Power and interest,
- Culture and values.
The *people competence* area is devoted to personal an interpersonal competencies. It includes the following elements:

- Self-reflection and self-management,
- Personal integrity and reliability,
- Personal communication,
- Relationships and engagement,
- Leadership,
- Teamwork,
- Conflict and crisis,
- Resourcefulness,
- Negotiation,
- Results orientation.

The *practice competence* area specifies methods, tools and techniques required for a successful project. It comprises of following elements:

- Project design,
- Requirements and objectives,
- Scope,
- Time,
- Organisation and information,
- Quality,
- Finance,
- Resources,
- Procurement,
- Plan and Control,
- Risk and opportunity,
- Stakeholders,
- Change and transformation.

IPMA ICB can be useful at any time during the project. It is best used as a handbook on different managerial competencies. It is suitable for experienced managers, who can embed IPMA ICB into a management process defined by other standards.
3.2 Project management in IT and standards of SW development

IT project is the same as any other project in its temporariness, uniqueness and in the way it drives change to the business, delivering value to its stakeholders. The difference is that its deliverables are mostly created and operated using information technology. IT projects can include procurement (e.g. selecting and deploying new antivirus software), network and infrastructure improvement (e.g. improving company’s network security), system integration (e.g. integrating new content management system with centralized authentication system), or software development (e.g. building an interactive website, developing a system to track child immunizations). The latter, software development, is the most common type of IT projects [8], [9] and therefore a focus of the designed course.

Nowadays, SW development projects are usually managed within an iterative and incremental, spiral-type lifecycle [9]. The software is developed by following repeating cycles of requirements analysis, design, implementation and testing, with little more added to the product each time, until final deployment. The way the outcome is delivered is not only dependent on the development lifecycle, but also the overall development approach. There are two main approaches to SW development – predictive and agile [10].

The predictive approach is more rigid, with a focus on processes. It is based on thorough upfront planning, with fixed requirements [9]. Agile approach, on the other hand, is a flexible and adaptable approach that focuses on stakeholder’s interactions. It minimizes upfront planning and welcomes regular update of requirements[9]. Both agile and predictive approaches are represented by specific standards. These standards, if applied appropriately, assure the productivity and quality of the project’s deliverables. They define the best-practices in the development process.

For the purpose of the course design, two SW development standards were selected - Unified Process as a predictive development standard and Scrum as an agile development standard. These standards are introduced in the next part of this chapter.

Unified Process is a predictive software development framework. It is use-case-driven, architecture-centric. Its lifecycle comprises of iterations that are grouped into phases – Inception, Elaboration, Construction and Transition (Figure 3.4).

![Figure 3.4: Lifecycle of Unified Process [11].](image)

An iteration of the Unified Process is like a mini-project and should not be more than three months long. It includes six workflows:

- Business modelling,
- Requirements,
- Analysis and design,
- Implementation,
- Tests.

As Figure 3.4 shows, these workflows change in their extent as the project progresses. At the inception and elaboration phase, more emphasis is given to Business modelling and Requirements. As the project progresses, more and more activities are related to analyzing, designing and implementing. The transition phase concentrates on testing.
Because Unified Process is architecture-centric, it is suitable for developing large systems with complex architecture. It requires thorough early-stage planning. A developed robust architecture defines the milestone between elaboration and construction phase. During the whole development process, UML diagrams are used to model the current state of the system and its requirements.

Unified Process is best used for large and complex systems that require thorough and early-stage planning. Due to its emphasis on UML modelling, it is suitable for projects that require comprehensive technical documentation.

**Scrum [12]**

Scrum is the most popular agile framework for developing software [4]. Just like Unified Process, it is iterative and incremental. Its agility is applied through strict procedural rules. Its lifecycle comprises of numerous, time-boxed iterations called Sprints (Figure 3.4).

![Figure 3.4: The lifecycle of Scrum [13].](image)

At the heart of a Scrum process is the Product Backlog. It is the scope of the product that contains independent items in the form of user stories. At the beginning of each Sprint, the team chooses a subset of the Product backlog called the Sprint backlog. At the end of each Sprint, a usable Product Increment is released and reviewed by the customer. Scrum is a framework that welcomes change. Its frequent releases and
customer reviews cause frequent changes to Product Backlog’s scope or priority order, all accordingly to the customer’s current feedback.

Scrum is best used in projects where exact requirements are not available upfront. Its frequent release of Increments makes it ideal for projects where time to market is a key factor. Because it is based on transparency and communication, it is best applied for projects with smaller teams that are strong at cooperation and open communication.

3.3 Selecting management approach

Every project manager that stands in front of an it-based project is in a high risk of failure. According to Brewer [9], software development projects, compared to other project domains, often come with ambiguous and frequently changed requirements. The used technologies can also change rapidly, making the implementation phase chaotic in many cases. Failures within one project can have a cascading effect on other projects in the company’s portfolio [14]. These specifics put IT projects at a higher risk of failure, yet the risk management is often underperformed. This results in many unsuccessful projects [1]. Table 3.2 shows the success rate of SW development projects.

<table>
<thead>
<tr>
<th></th>
<th>2013</th>
<th>2014</th>
<th>2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUCCESSFUL</td>
<td>41%</td>
<td>36%</td>
<td>36%</td>
</tr>
<tr>
<td>CHALLENGED</td>
<td>40%</td>
<td>47%</td>
<td>45%</td>
</tr>
<tr>
<td>FAILED</td>
<td>19%</td>
<td>17%</td>
<td>19%</td>
</tr>
</tbody>
</table>

Most common reasons for an IT project’s failure include personnel shortfalls, unrealistic time and cost estimates, misunderstanding of the requirements, failure to
gain user commitment, late changes to requirements or lack of effective project management skill and approach [15]. The latter - failure to select an appropriate approach, can raise the probability of all the other risks. For example, taking a predictive approach with thorough upfront planning where exact requirements are not known upfront can result in late unwelcome changes to requirements. On the other hand, an agile approach to an extensive and heterogenous project can lead to unrealistic time and cost estimates. Numerous studies have found that project approach needs to be selected with respect to initial project factors, like team size and project domain [16], [17] documenting and reporting policy [18], or the stability of requirements and the need for fast time to market [19], [20].

I have created Table 3.2 to show the compatibility of selected standards with initial project environment. The initial project settings are divided into three categories – managed team, company and project. Each of the categories contains numerous initial settings. The compatibility is shown through colored flags - green flags when the standard’s and setting’s compatibility is ideal, yellow flags for matches that are not optimal, a red flag for conflicting matches that can easily lead to unsuccessful project management.

**Table 3.2: Matching standards to project environment**

<table>
<thead>
<tr>
<th></th>
<th>PROJECT MANAGEMENT</th>
<th>SW DEVELOPMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PRINCE2</td>
<td>PMBOK</td>
</tr>
<tr>
<td>Managed team</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Needs command and control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cooperative, transparent, self-organizing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed across locations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Colocated</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Larger than 15 people</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smaller than 15 people</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Company</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requires detailed reporting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requires detailed documenting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has flat organization structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Has hierarchical organization structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fast time to market</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Needs rapid customer’s feedback</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complex and unpredictable scope</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Large scope with a span over 1 year</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scope realizable within 1 year</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements unstable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements clearly defined upfront</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain - exploratory or innovation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain - safety-critical</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
When teaching project management, putting different approaches into comparison can lead to a better understanding of the subject. In a recent study provided by Castillo et al. [21], teaching agile concepts has been improved by comparing Scrum to more traditional Unified Process. The study has demonstrated the improvement in comprehension of Scrum and agile methods in general when compared to more traditional approaches. These findings and a review of example-based methods of learning that promote analogical reasoning (see Chapter 5) has led me to a decision to base major part of the course on two examples of projects, one approached with traditional management, the other one with agile management.

Teaching project management by showing two examples based on Scrum and Unified Process might not be enough. When approaching an IT project, a manager will most likely need to adopt or consult more than one standard. According to Archer [22], generic project management standards are usually designed to be used in conjunction with other specialist standards, such as standards of software development. Project management standards guide managers on non-technical aspects of the project, like stakeholder management and risk management. The SW development standards, on the other hand, are more dedicated to managing requirements and handling the development process. Examples of combining standards can be found in [22] (combining UP and PRINCE2), [23] (combining UP and PRINCE2), [24] (combining Scrum and PRINCE2) and [25] (combining UP and PRINCE2, Scrum and IPMA ICB). The example-based part of the course will show such combinations in practice with respect to the findings shown in Table 3.2.

Table 3.2 shows that general standards like PRINCE2, PMBOK and IPMA can be combined with both agile and traditional development process. However, as IPMA and PMBOK do not prescribe a lifecycle or heavy reporting and documentation processes, it will be teamed with a lightweight development approach - Scrum. In the example, the Scrum will be used for the lightweight development process, supplying key roles, events and artifacts. IPMA ICB and PMBOK will supply the manager with knowledge on how to create main project documentation, key strategies and how to handle personal and interpersonal competencies. In the second example, the project will be managed more traditionally, using Unified Process and PRINCE2.
standards. Unified Process will guide the manager through the development process, requirements analysis and product documentation. PRINCE2 will supply terminology, roles and guiding principles, leading the manager through thorough up-front planning, documentation and reporting. The full description of the two examples can be found in chapter 6 that presents details of the course’s design.
4 Project management courses at Czech universities with IT specialization

In order to gain insight into how Project management is being taught at other universities, I have analyzed the contents of numerous Project management courses' syllabi. The purpose of this research is to find the most common topics and to develop an awareness of which important topics are missing. This chapter briefly outlines courses that have been searched, the method that has been used to analyze and evaluate their syllabi's content, and the findings and conclusions of the analysis.

4.1 Selected courses

The reviewed courses are dedicated to project management and taught at Czech universities' faculties with IT specialization. The selection of faculties was constricted geographically to the three largest Czech cities, i.e. Prague, Brno and Ostrava. Out of these cities' faculties, the ones that included a reference to Information Technology or Computer Science in the title were selected. The selected faculties are:

- Faculty of Information Technology, Brno University of Technology,
- Faculty of Information Technology, Czech Technical University in Prague,
- Faculty of Engineering and Computer Science, Technical University Ostrava,
- Faculty of Informatics and Statistics, University of Economics, Prague,
- Faculty of Informatics, Masaryk University, Brno.

The selection was then based on the course’s name and specialization related to project management. In order to gain insight into the contents of the course, the publicly available syllabi and course objectives have been examined. This method generated nine project management courses. Following is a list and a brief outline of the courses’ contents.
TheFIT’s course is focused on a process approach to project management. It introduces basic project management terminology and knowledge areas before moving on to the process-driven organization of the project. The teaching of application of processes in the project follows the project’s lifecycle, including the planning process, execute and control process, closure process. The course’s content also includes methods and techniques of project planning and quality measurement and current standards of project and process management.

The same faculty offers their students a course focusing on the project manager’s competencies. The course follows themes and competencies defined in the International Project Management Association’s ICB standard. The contents range from technical and contextual competencies, including risk, change and quality management, standards and regulations, to behavioural competencies and soft skills, including personal communication, teamwork and negotiation.

The course introduces basic project management terminology and principles, together with the project’s lifecycle. A substantial part of the content is dedicated to project management’s themes, with a focus on the business case, planning, scheduling and risks. Part of the course’s focus is on soft skills, particularly communication, motivation and ethics of project management. Practical topics include tools and techniques, including Work Breakdown Structure (WBS), Gantt diagrams and SWOT analysis.

---

9 http://www.fit.vutbr.cz/study/course-l.php.cs?id=12850
10 http://www.fit.vutbr.cz/study/course-l.php.cs?id=12888
11 http://bk.fit.cvut.cz/cz/predmety/00/00/00/00/00/01/80/90/p1809006.html
The course focuses on change management in projects. Major part of the course is dedicated to change management in IT projects, methods and techniques of resolving issues and requests for change. Other topics include project organization, traditional and agile methods of project management, strategic project management, quality and financial management.

The course is dedicated to case studies of project management in IT. The lectures are given by managers ranging from small companies to large corporations. Lectures are devoted to both traditional and agile project management, software testing methods, team organization, soft skills and change management. The course combines theoretical frameworks with practical case-studies, which gives the students insight into real-life project management.

A significant part of the course is dedicated to project management knowledge areas, including lectures on project organization, planning, resources management, change, risk and quality management. Practical tools and techniques taught in this course include WBS, Gantt diagrams, Program evaluation and review technique, Critical path method, and Network analysis. The course includes a lecture on soft skills, where a case study on the conflict at work is presented.

---

12 http://bk.fit.cvut.cz/cz/predmety/00/00/00/00/00/00/04/67/24/p4672406.html
13 http://bk.fit.cvut.cz/cz/predmety/00/00/00/00/00/00/01/80/89/p1808906.html
14 https://edison.sso.vsb.cz/cz.vsb.edison.edu.study.prepare.web/SubjectVersion.faces?version=4604049/02&subjectBlockAssignmentId=334906&studyFormId=1&studyPlanId=20994&locale=cs&back=true
The course includes topics both general project management and IT project management. Introduced general project management standards include PMBOK, PRINCE2 and ISO standards; specialized standards are represented by Multidimensional Management and Development of Information Systems (MMDIS), Control Objectives for Information and Related Technologies (COBIT) and Information Technology Infrastructure Library (ITIL). Other topics include knowledge areas, including Planning, Risk and Organization. This course also lectures students on specialized tools and techniques, including WBS, Program evaluation and review technique and Critical Path Method.

The course introduces both soft and technical skills of project management. Soft skills lecturing includes teamwork and problem-solving. The lectures follow the project’s lifecycle, with a focus on time and cost analysis, quality management and control processes.

The course introduces basic project management terminology and lifecycles. The knowledge areas are represented by a lecture on Risk and Quality management. A large part of the course is dedicated to inspection, testing and metrics of software products. The course also introduces standards of Project management – PMBOK, PRINCE2 and IPMA ICB. The course also presents Project management tools and techniques, including WBS, Gantt diagram and Network diagrams.

https://insis.vse.cz/katalog/syllabus.pl?odkud=;zobrazit_sklad=0;zobrazit_obdobi=0;obdobi=;predmet=146423

MMDIS is an Information systems development method created at the Faculty of Informatics and Statistics at University of Economics, Prague.

https://insis.vse.cz/katalog/syllabus.pl?odkud=;zobrazit_sklad=0;zobrazit_obdobi=0;obdobi=;predmet=146329

https://is.muni.cz/predmet/fi/PA179
4.2 Evaluating method

In order to assess the content of the selected courses’ syllabi, a list of project management topics has been created. This list has been divided into the following five topic categories:

- Basic terminology and concepts,
- Knowledge areas,
- Standards and methodologies,
- Lifecycle and processes,
- Tools and Techniques.

Each category includes related topics that were selected based on their occurrence in the course’s syllabi. For example, the category Knowledge areas include the following topics:

- Business case,
- Organization,
- Quality,
- Plans,
- Risks,
- Change,
- Progress.

Some of the topics in the Standards and methodologies category included:

- PMI PMBOK,
- PRINCE2,
- IPMA ICB,
- Scrum,
- ITIL,
- ISO.
In order to quantify the content, a matrix of values has been designed, where each topic on the list was given points based on their occurrence in a given syllabus. The points scale comprises of three values – 0 points for topics that do not appear in the course content; 0,5 points for topics that appear in the course content, but do not span over more than or 1/12 of the course (a rough proportional equivalent to one lecture); 1 point has been given to topics that appear to the extent of more than 1/12 of the course. Table 4.2 depicts the method of assigning the points.

<table>
<thead>
<tr>
<th>Topic extent in the course content</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
</tr>
<tr>
<td>Up to 1/12 of the course content</td>
<td>0,5</td>
</tr>
<tr>
<td>Exceeding 1/12 of the course content</td>
<td>1</td>
</tr>
</tbody>
</table>

Based on this marking, a maximum number of points that a given topic could acquire was 9, that is one point for every analyzed course.

### 4.3 Results and conclusions

Eight out of the nine courses had an opening lecture on basic terminology and concepts of project management. This is understandable as introducing basic concepts is considered essential in order to get across further knowledge later in the course. The strategic view of project management and its relation to Programme and Portfolio management has appeared in three courses’ syllabi.

Out of the Standards and methodologies, more attention was given to project management standards, with PMI’s PMBOK, PRINCE2 and IPMA ICB leading the list and each appearing in at least 3 out of 9 courses’ syllabi. The SW development standards were only mentioned in few of the course’s syllabi, with Scrum being the most often (1 point).
The scope dedicated to knowledge areas was significant in all of the courses. The most extent was given to Plans, Quality and Organization gaining 5.5, 4 and 3.5 points, respectively. Interestingly, the Risk topic attained the least points of the knowledge areas, scoping 2.5 points.

The project’s lifecycle and its processes were also a topic of almost every course, with most attention given to initiating and delivery processes (3 points each) and less focus given to controlling and closing processes (2.5 points each).

Out of the tools and techniques found in the course’s content syllabi, Soft skills techniques appeared most often (4 points), mostly centered around communication and resolving conflicts. Other tools and techniques at top positions are Assembling a project plan, SW product metrics, Gantt diagrams, WBS, Critical path method, Team roles designating, Cost estimate and Resources control – all appearing in at least four of the courses’ syllabi, scoring at least 2 points. Other tools and techniques scoring at least 1 point included: PERT, Responsibilities matrix, Risk analysis and Risk register and Network diagrams.

The majority of topics found in the analyzed courses is covered by project management standards, whether it be knowledge areas, processes or tools and techniques. Basing the course’s content on the topics of project management standards has therefore proved to be a good idea. The analysis has inspired me to include a bigger, strategy-based picture on project management, with an introduction to Programme and Portfolio management and IT services management. Even though the analyzed courses were all taught at faculties specialized in IT, little content was dedicated to the standards of software development and the way they engage with the standards of project management. This is where the designed course aspires to be different. On two consecutive examples, it will present the difference between managing a project in an agile way and a traditional way, using both standards and lifecycles of software development and project management.
5 Example-based learning

Example-based learning is a popular educational method. According to cognitive research findings [26], providing examples to students in order to improve learning has proven relevant and effective, often reducing knowledge acquisition time [27]. The lecturers, unlike their students, have usually gone through extensive experience in their field and picturing an instance or an example of a lectured concept is much simpler to them. However, this is not the case with students, as their years of experiencing real-life problems are yet to come. Teaching with examples is a way to bridge the experience gap and give students a powerful way of learning. This chapter introduces some basic concepts of example-based learning and research-based suggestions for the design of the examples.

5.1 Types of example-based learning

According to Renkl [28], example-based learning is traditionally researched in the following three fields: observational learning, analogical reasoning and worked examples. Observational learning involves observing another person’s behaviour, encoding and remembering the demonstrated activity. It usually includes instructional explanations, e.g. demonstrating how to build a cube out of a piece of paper. Analogical reasoning relies on examples of familiar cases or domains to explain unknown concepts, e.g. comparing algorithm to a recipe for cooking. As the requirements ask for static lecture materials, further review will focus on the worked example and analogical reasoning methods.

A worked example involves studying problems for which the solution is given. It includes formulated problem, solution steps and the final solution. It is most suitable for beginners in the respective field of study and loses effectiveness with the increase of learner’s expertise [29]. Learning from a worked example allows students to abstract general rules, which helps them to solve similar problems later on [27]. It also relieves student’s cognitive load, supports retention of the knowledge, eases the transfer of information between the lecturer and the student, helps with cognitive load and reduces time spent on learning the concept [27]. Although
widely used by in mathematics and geometry, the worked example model is suitable for other fields as well. As Van Gog and Rummel state: „... several recent studies have shown that worked examples can also be effective with less structured cognitive tasks such as learning how to apply an instructional design model, learning argumentation skills, learning to construct concept maps, learning to recognize designer styles, or learning to reason about legal cases” [27].

Analogical reasoning is fundamental in identifying causal relations where knowledge about one example is used to achieve understanding about another example. Providing analogies of different examples prove the most efficient way of realizing and deeply understanding domain principles. Examples based on analogy help where students have little domain knowledge [28].

5.2 Recommendations for designing the examples

According to Sweller et al. [30], learning by studying already solved examples can be more effective than having to solve the problem personally as the latter often leads to weak problem-solving strategies that are not effective for deep learning of the concept. This is referred to as the “worked example effect”. The successful learning from a worked example, however, considerably relies on the design of the example [31], [26], [32], where the most emphasis should be based on its structure. In the first phase, students should learn the basics of the domain and its principles. In the next part, an example is introduced, with references to the previously explained domain basics. Next, the steps leading to the final solution are presented. According to Atkinson et al. [26] visually isolating these steps positively impacts the learning and gives clarity to using a structure of sub-goals. The final solution should be reached using reasoning based on domain principles. Further suggestions to the design of the example include the use of multiple modalities (aural, visual.. ) and integrating examples into one source. Focusing the lecturer’s attention to one information source, rather than multiple formats, has also proven effective. In analogical reasoning, students should be introduced to the abstract principles in order to understand analog examples. In multiple research findings, the effectiveness of the learning relied on the learner’s ability to self-explain the given example. Ideally, the
student should explain the rationale behind the solution to themselves. This can be fostered by the lecturer giving instructional explanations when laying the example [26].

As the course’s requirements ask for static study materials (lectures slides), the worked example appears to be the best way to illustrate the use of some of project management concepts on the slides. Analogical reasoning will also be a major component of the course’s design, with the example based on two analogical projects, each with different initial settings and different management approaches. Both worked example, and analogical reasoning are suitable for lectures as no time-consuming problem solving is required on the students part. Both methods are also effective and suitable for novice learners. When constructing the examples, the above mentioned recommendations for its design will be taken into account.
6 The developed course

Based on the implications of analytical parts of the work, the iterative design of the course has commenced. The course’s structure was laid out into 12 lectures. Each lecture’s design has been consulted with the lecturer, before creating the multimedia presentation corresponding with the lecture’s content. The lectures were presented to students in a pilot run. The feedback from the lecturer after the lecture has been given was incorporated to improve the quality of the presentations.

This chapter describes the outcome of this process - the developed course. To improve the reader’s understanding of the design, I have divided it into two domains – form and content.

6.1 Course’s form

The form domain is dedicated to the way the course and its lectures are structured and what elements are used to support better understanding and learning. It is the “HOW” part of the course design, focusing on the composition, techniques and media used.

Structure of the course

Because the course is taught on a semester basis, it has been divided into 12 lectures, one lecture for each week of the semester. These 12 lectures are further divided into four parts.

The first part is three lectures long. It is dedicated to the theory and based on selected standards for project management and software development. The inclusion of this part is grounded in the worked-example method that suggests introducing students to basic concepts first, before diving into the example-based part.
The second and third part are each four lectures long and each dedicated to one example of a project. The first project is approached in an agile way, the second one more predictively. Each of these two examples follows the lifecycle of a project from its pre-project phase, through to its close. The four lectures that each project example comprises of are filled with corresponding techniques and tools that are typical for the specific approach. Going through a whole lifecycle of a project, before commencing the other one promotes analogical reasoning and supports a better understanding of the implications of the two different approaches. For instance, when explaining work estimation, the first example introduces the concept of user stories and its estimation through story points, using the planning poker method (Figure 6.1).

![Figure 6.1: Estimating work through user stories in the agile project example.](image-url)
The second example estimates work through PERT (Program Evaluation and Review Technique) (Figure 6.2).

![PERT (Program Evaluation and Review Technique)](image)

**Figure 6.2: Estimating work through PERT in the predictive project example.**

The aim of both approaches is the same – estimate work to be done so that the project can be planned accordingly. Techniques used are different and fit accordingly to the chosen approach.

The examples of tools and techniques given are mostly based on the specific project. As seen in figure 6.1, the user story is not generalized, but specifically related to the given project. Providing real-life examples, similar to what an actual project would be like was one of the main goals of the course’s design.

The *fourth and last part* of the course contains one lecture and summarizes the course and its contents. Its goal is to reflect on all of the lectures and take out important lessons learned.
Structure of each lecture

Each lecture has its corresponding multimedia presentation ("slides"). The slides comprise of text, as well as images, graphs, tables and other visual elements. The use of various media is supported by the worked example method. Also with respect to the worked example method, the information given on each slide is apposite and understandable, giving the lecturer space to further elaborate on the subject orally. The design of the slides is simplistic, and apart from the contents, it contains only a few necessary elements – the title of the slide and where applicable, a logo or a footer that indicates the current subject matter (Figure 6.3).

Managing the triple constraint – „PICK TWO“

- The triangle defines project's boundaries
- Balancing the triangle is project manager's responsibility
- Reducing one side increases pressure on the other two (e.g. reducing cost puts pressure on scope and time)
- The „PICK TWO“ rule means you can have:
  - Fast and cheap, but it won’t be good
  - Good and cheap, but it won’t be fast
  - Good and fast, but it won’t be cheap

Figure 6.3: Example of a slide with an infographic element.
Each presentation has its heading slide that contains the lecture’s title, course’s code, lecturer’s name and the name of the authors. The design of this slide is simplistic, displaying the logo of the Faculty of Informatics, consistent with the university’s new visual identity (Figure 6.4).

![Figure 6.4: The title slide of the first presentation.](image-url)
The second slide of each presentation presents an outline of the given lecture (Figure 6.5).

**Outline of today’s lecture**

1. Main project characteristics
2. Project vs Process
3. Project, Program, Portfolio
4. Project management
5. PRINCE2
6. PMI PMBOK
7. IPMA ICB

*Figure 6.5: Slide with an outline of the first lecture.*

The selected standards are introduced in a composed manner, looking at overall structure first, before diving into the individual concepts and principles (Figure 6.6).

*Figure 6.6: Four slides introducing the PRINCE2 structure.*
Each lecture or a specific part of the lecture ends with a multi-choice quiz that tests the reader’s knowledge of the introduced concepts (Figure 6.7). The last slide of each lecture is dedicated to sources that have been used to create the lecture’s content. Where applicable, the reference’s number also appears on a corresponding slide.

### Quiz * Project management basics

More than one answer can be correct

1. **What are the main characteristics of a project?**
   - A. Temporary, Change-driving, Uncertain, Unique
   - B. Repetitive, Stable, Linear, Event-driven
   - C. Permanent, Simple, Flexible, Task-driven

2. **What are the usual dimensions of project manager’s triple constraint?**
   - A. Time, Cost, Scope
   - B. Risk, Quality, Procurement
   - C. Plan, Change, Progress

3. **What is the aim of Project Portfolio?**
   - A. To add value to business
   - B. To fulfill strategic goals
   - C. To create a unique product

4. **A process is best visualized with:**
   - A. Gantt chart.
   - B. Flowchart.
   - C. Use case diagram.

*Figure 6.7: Slide with a quiz on project management basics.*

More structural elements appear once the course moves on to the project examples. Because the two example projects are presented following their respective lifecycle, two diagrams picturing these lifecycles, together with the deliverables of each stage have been created. At the beginning of each lecture, the student is reminded of where in its lifecycle the project currently is and what are the deliverables examined (Figure 6.8 and 6.9).
In the examples, each slide comes with a footer that informs the student of the current project’s domain (also Figure 6.8 and 6.9).
Other functional elements

When explaining basic concepts and principles of project management, definitions of important concepts frequently appear as text boxes (Figure 6.10).

The uncertainty of project

Projects are generally more risky
  • We have deadlines
  • We are introducing change
  • We have never done this before

*Figure 6.10: Use of an isolated text box to present an important definition of Project risk.*
When introducing techniques or artifacts, the WHAT, WHY, HOW, in some cases WHO form of depiction is frequently used (Figure 6.11). This is a commonly applied and effective form of information gathering, used to get a complete view of a certain subject [33].
In the worked out examples of projects, when presenting specific examples techniques, a double-sided textbox is used. One side of the textbox marked “Theory”, depicts the theory and instructions for using the technique. The other side presents an example based on such instructions (Figure 6.12).

Figure 6.12: Double-sided textbox, presenting both theory and example on assessing risks.
6.2 Course’s contents

The contents part of this chapter is dedicated to the subject matter of the course. It is the “WHAT” part of the course design, focusing on the significance of the selected topics and the context in which they are used.

Contents outline

As already mentioned in this chapter, the course contents comprise of twelve lectures that can be further grouped into four parts.

The first part that spans over three lectures introduces fundamental concepts of project management, together with the basics of each of the selected standards. The second and third part are both example-based and both comprise of four lectures. The first example depicts an agile approach to project management; the second shows a more predictive approach to project management. Both examples take the student through a whole lifecycle of a project, from the initiating and planning activities through to it close. The fourth part summarizes the course and focuses on the differences between the two introduced examples. Figure 6.13 illustrates the course’s outline. The next part of this chapter describes each of these four parts in detail.

Figure 6.13: The contents outline of the course.
IT project management basics

The first lecture starts with introducing the main project characteristics. It explains the difference between project and process, showing examples of both. Understanding processes role in projects is important as two process-based standards are introduced later on. Next part shows the big picture of project management – the relationship between portfolio, program and project management, depicting the importance of strategic objectives, business case and delivering value to customers. An example of portfolio, program and project hierarchy is also given (Figure 6.14). Even though general project management concepts are introduced, the examples shown are IT-based. Showing IT examples instead of generic ones is an important aspect of the whole course.

**Portfolio, program, project – an example**

![Diagram of portfolio, program, and project hierarchy](image)

*Figure 6.14: Slide with an example of portfolio, program and project hierarchy.*

The next part of the lecture defines project management and familiarizes students with the triple constraint [34] and the importance of adopting an appropriate standard. Following is an introduction to three standards – PRINCE2, PMBOK and IPMA ICB. Through the PRINCE2 and PMBOK standards’ description, students are introduced to the main knowledge areas of project management and the processes that need to take place in order for the project to be successful. The students get a grasp of the project’s lifecycle and which managing processes are important in each of its
phases. Describing ICB IPMA standard raises awareness of the competencies a project manager should have, including soft-skills. Each standard’s introduction includes recommendations on which type of projects it is ideally suited (Figure 6.15). Included is also a slide on associated certifications and how to obtain them.

---

**How to use PRINCE2**

- PRINCE2 is best used as a **method to follow** from the beginning of the project to its closure
- It is suitable for experienced managers, as well as **entry-level project managers**
- Consider adopting PRINCE2 if:
  - the company requires comprehensive reporting
  - the company requires complete project documentation
  - the team needs “command and control” type of management
- PRINCE2 method does not address **requirements and budget management**
  - Consider complementing PRINCE2 with other standards to fill this gap
- Following PRINCE2 method **to the letter** can create a substantial document and reporting overhead
  - This should not be encouraged

*Figure 6.15: Slide with recommendations on which type of project is PRINCE2 ideally suited.*
The second lecture starts with the definition of an IT project and IT project specifics. It presents the bigger picture on IT project, focusing on cases when the project’s deliverables become part of services and products that require further management. This reasoning leads to the introduction of the IT Infrastructure Library (ITIL) standard (Figure 6.16), with a brief description of its stages.

**ITIL (IT Infrastructure Library)**

- Best practices for IT service management
- Helpful with „What happens before and after an IT project?“
- Five lifecycle stages
  - Service Strategy
  - Service Design
  - Service Transition
  - Service Operation
  - Continual Service Improvement
- Each stage consists of processes with clearly defined inputs and outputs

Figure 6.16: Slide introducing ITIL.
After giving examples of types of IT project management, an introduction to SW development starts, presenting the lifecycles and main approaches of SW development. The lecture continues with a description of two selected SW development standards – Unified Process and Scrum. Each standard’s description comes with an elaboration on its structure and lifecycle, focusing on the implications of taking a predictive or agile approach. Each development approach and includes a slide on contracting, depicting the advantages and disadvantages of both approaches (Figure 6.17).

Contracting in Agile development

Time and means contract

<table>
<thead>
<tr>
<th><strong>Advantages</strong></th>
<th><strong>Disadvantages</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Flexible contract</td>
<td>• Constant customer involvement requires extra time</td>
</tr>
<tr>
<td>• Scope is not known upfront, we can easily add features and fulfill change requests</td>
<td>• Difficult to predict final budget and deadline</td>
</tr>
<tr>
<td>• Frequent customer supervision drives more trust</td>
<td>• Managing the triple constraint becomes a continuous mission</td>
</tr>
</tbody>
</table>

**Figure 6.17: Slide on contracting in Agile development.**
The third lecture reviews the introduced standards and their typical application. It explains why a combination of standards will most likely need to be adopted in order to successfully manage an IT project. Following are instructions on how to choose an appropriate standard or combination of standards. This idea is supplemented with the table Matching standards to the project environment (Table 3.2 in Chapter 3) that shows the suitability of each standard for given initial settings of the project. The lecture concludes with an introduction to upcoming, example-based lectures (Figure 6.18).

Figure 6.18: Slide with introduction to upcoming, example-based lectures.
Agile example

The first part of the course familiarizes students with main IT project management concepts, including basic principles, knowledge areas, processes and essential project manager’s competencies. In the second and third part, these concepts are shown in practice on two worked-out examples of projects.

The agile example starts with presenting the initial characteristics of the example project. The project manager is fresh out of school with some PM experience. He is dedicated, enthusiastic and confident in his capabilities. The company he works for is a startup with an uncertain future, simple governance and a flat type of hierarchy. The team working on the project is collocated, with eight enthusiastic and independent employees. The customer is the Centre for students with special needs. The project’s objective is to create an Information system for disabled students. The requirements are unstable, with clear demand for some core functions. The initial functionality of the product is needed ASAP. All these characteristics complete a rationale behind choosing an appropriate approach to manage such a project. A table “Comparing standards” is again used, this time with only selected characteristics of the related project and selected standards. Figure 6.19 shows a mismatch of project’s characteristics and standards. Managing this project with PRINCE2 and Unified Process cannot be encouraged.
Figure 6.20 on the other hand, shows the initial settings and its matching standards.

As presented in Figure 6.20, Scrum, IPMA and PMBOK are a more suitable approach to this project’s management. The lecture continues by explaining what these standards are used for when managing the project. IPMA and PMBOK will be used for creating a project charter, developing key strategies and managing personal and interpersonal competencies. The Scrum will provide key principles based on its development lifecycle, roles, events and artefacts.

After the introduction and approach selection, the work on the project can start. The lifecycle (Figure 6.8) starts with the planning phase in which a Project charter, Key strategies and Product backlog is created. The project then continues to several Sprints that start with creating Sprint backlog and conclude in releasable Increment. The closing phase concentrates on the project’s retrospection. Each time a tool or a technique is introduced, it is illustrated on worked example directly related to the project.
Table 6.1 summarizes the contents of the four lectures dedicated to this project and its lifecycle. It shows the tools and techniques taught at each of the phases. Where applicable, the table also shows which standard the use of such tools or technique originated from. It shows that IPMA and PMBOK standards were mainly used to manage the first, planning phase. Once the project got into the Sprinting phase, it was primarily managed by the Scrum principles.

Table 6.1: Tools and techniques included in the first example.

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Phase</th>
<th>Tools and techniques learned</th>
<th>Standard of origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Planning</td>
<td>Selecting the project approach</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Determining project lifecycle</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developing project charter</td>
<td>PMBOK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gantt chart</td>
<td>PMBOK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Precedence Diagramming Method (PDM)</td>
<td>PMBOK</td>
</tr>
<tr>
<td>5</td>
<td>Planning</td>
<td>Developing communication strategy</td>
<td>IPMA/Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developing risk strategy</td>
<td>IPMA/PMBOK/Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Creating Risk register</td>
<td>PMBOK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developing change management strategy</td>
<td>IPMA/Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developing quality management strategy</td>
<td>IPMA/PMBOK/Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Burndown chart</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Calculating team velocity</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Employing Definition of Done</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Creating Product Backlog</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Composing and estimating user stories</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Planning Poker</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Prioritizing user stories with MoSCoW technique</td>
<td>Scrum</td>
</tr>
<tr>
<td>6</td>
<td>Sprinting</td>
<td>Determining Sprint goal</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Creating Sprint backlog</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Daily SCRUM</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sprint Review meeting</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Increments release process</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sprint Retrospective meeting</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updating key strategies</td>
<td>PMBOK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Agile Change requests management</td>
<td>Scrum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Handling conflicts in the team</td>
<td>IPMA</td>
</tr>
<tr>
<td>7</td>
<td>Closing</td>
<td>Handing over documentation</td>
<td>PMBOK</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Project retrospective meeting</td>
<td>Scrum</td>
</tr>
</tbody>
</table>
Predictive example

Analogically to the first project example, the second project example also starts with introducing its initial settings. To elicit a sense of a situation in which the student stands in front of two different projects, the project manager stays the same. This time, however, he works for an established and large software solutions company with a department-based hierarchy. The customer is a local council – stable and traditional institution with complex and rigid governance that requires detailed documentation and reporting. The project delivers a software solution for city information kiosks. It spans over 18 months and requires a large team that is located across different departments and premises. The scope of the project will be defined and signed off upfront. Unlike the other project, this one will need to be heavily administrated, with command and control type of management. The Comparison of standards table is again used to choose an appropriate combination of standards. It clearly shows that Scrum is not suitable for this project (Figure 6.21) and that Unified Process will be a better choice for a large scope and thorough upfront planning. Because of the heavy administration and reporting required, the Unified Process will be best teamed with PRINCE2 (Figure 6.22).

Selecting management approach

---

**Figure 6.21: A mismatch of second project’s initial settings and standards.**
In the second project example, PRINCE2 will be used for basic terminology and guiding principles. Through PRINCE2, the Project plan and other initiating documentation will be developed. PRINCE2’s techniques will also be used for reporting and handling of issues. Unified Process will be used primarily for requirements analysis, iterative development process and creating technical documentation. Figure 6.9 shows the second project’s lifecycle and its deliverables.
Table 6.2 summarizes the contents of the second example’s four lectures. It also shows the tools and techniques taught at each of the phases and related standards.

Table 6.2: Tools and techniques included in the second example.

<table>
<thead>
<tr>
<th>Lecture</th>
<th>Phase</th>
<th>Tools and techniques learned</th>
<th>Standard of origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Starting up</td>
<td>Selecting project approach</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>(Inception)</td>
<td>Determining project lifecycle</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Analyzing high-level requirements</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Feasibility analysis</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developing Project brief</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Appointing Project management team</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Creating Next stage plan</td>
<td>PRINCE2</td>
</tr>
<tr>
<td>9</td>
<td>Initiation</td>
<td>Developing Project Initiation Documentation (PID)</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td>(Elaboration)</td>
<td>Appointing roles and responsibilities</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Detailed requirement analysis</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Product specification</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Developing Project plan</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Work Breakdown Structure</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PERT (Program Evaluation and Review Technique)</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Resource assignment matrix</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Critical Path Method (CPM)</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>End stage report</td>
<td>PRINCE2</td>
</tr>
<tr>
<td>10</td>
<td>Delivery</td>
<td>Organizing teams around architecture</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td>(Construction)</td>
<td>Defining work package</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Highlights report</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monitoring risks and issues</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Escalating risks and issues</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exception report</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exception plan</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Team plan</td>
<td>PRINCE2</td>
</tr>
<tr>
<td>11</td>
<td>Close</td>
<td>Beta testing</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td>(Transition)</td>
<td>Product acceptance testing</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Acceptance record</td>
<td>UP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Service level agreement</td>
<td>ITIL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>End project report</td>
<td>PRINCE2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lessons report</td>
<td>PRINCE2</td>
</tr>
</tbody>
</table>
Course summary
The twelfth lecture reflects on the course, especially on the two examples. The differences in the two approaches are shown by comparing the two project’s team management, customers communication, domains, lifecycle (Figure 6.24) and handling of risk and quality (6.25). The course concludes with main lessons learned and project success factors.

Figures 6.24: Comparing the two project’s lifecycles.

Figures 6.25: Comparing the two project’s risk and quality management.
Seminars
Part of the requirements was to suggest contents for practical seminars. This is in case the allotted time for the course changes in the future, and the students will get a chance to practice techniques presented in the lectures. Table 6.3 shows recommendations for such practical seminars. Their topics correspond with the topics of each lecture.

Table 6.3: Recommendations for practical seminars.

<table>
<thead>
<tr>
<th>Seminar</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Come up with three examples of process and three examples of project. Select one project and define its at least three goals. Think of a program and portfolio the project could fit into in a large company.</td>
</tr>
<tr>
<td>2</td>
<td>Discuss the difference between a software development project and a building construction project. Why is iterative and incremental approach optimal for IT projects? What are the major risks of IT projects and why?</td>
</tr>
<tr>
<td>3</td>
<td>Come up with two examples of a software development project. One compatible with an agile approach, one with a predictive approach. Define what the project’s initial settings (team, customer, product, company’s policies) should be in order to fit into the selected approach. Brainstorm on what would happen if the projects swapped approaches without swapping the initial settings.</td>
</tr>
<tr>
<td>4</td>
<td>Create Project charter for the agile project. Include Business case, description of outcome, identification of external and internal stakeholders and description of the management approach. Define the activities of the first phase of the project and their relationships using PDM.</td>
</tr>
<tr>
<td>5</td>
<td>In your project, identify at least three risks. Assess these risks based on probability and impact and define what would be the consequences. Define responses to these risks. Compose three user stories, each with a defined at least two defined acceptance criteria. Estimate these user stories with story points using planning poker.</td>
</tr>
<tr>
<td>6</td>
<td>In your project, define the first Sprint’s goal. Assign tasks to two user stories from the last seminar. Include responsibilities. In a group of four, simulate Daily Scrum. Describe what you did yesterday, what you plan to do today and if you noticed anything that can be improved. Think of two ways communication in your group could be improved and present them to the other groups.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>7</td>
<td>In a group of four, simulate conflict in a team. Use de-escalation techniques to handle the conflict. Come up with a major change request to your project. What are the consequences of this request? How will it change the Product Backlog?</td>
</tr>
<tr>
<td>8</td>
<td>For the predictive project, describe high-level requirements. In a group of four, describe at least two core functions using two use case diagrams. Create a Work Breakdown Structure for the project.</td>
</tr>
<tr>
<td>9</td>
<td>Estimate 3 work packages from the WBS with PERT. Calculate the expected duration of each work package in work days. Define tasks in one of the work packages and assign their responsibilities using the RACI chart.</td>
</tr>
<tr>
<td>10</td>
<td>In a given example, determine Critical path using CPM. Simulate managing an issue that exceeds set tolerances. Assess the issue. Create an exception report and define causes of the exception, its consequences and suggest possible options.</td>
</tr>
<tr>
<td>11</td>
<td>Create an acceptance record. Define early-life support requirements for your product. Elaborate on what should be included in the manual for operations and maintenance of the product.</td>
</tr>
<tr>
<td>12</td>
<td>Discuss the differences in managing an agile and predictive project.</td>
</tr>
</tbody>
</table>
7 Discussion

This thesis aimed to develop a course on project management in IT. The course’s design is based on and inspired by the following: lecturer’s requirements; standards of project management and software development; contents of project management courses taught at faculties with IT specialization; and pedagogic best-practices.

Requirements 1 and 2 (see Table 2.1) requested a course that included both theoretical grounds and practical examples, reflecting the contents of popular standards of project management and software development and its combination. This was done by dividing the contents into four parts. The first part introduces basic project management terminology and principles, together with widely used standards of project management and software development. Also introduced are knowledge areas, principles and processes of project management, as well as competencies that a project manager should possess. Once this basic knowledge sets in, the second and third part of the course presents two consecutive project examples, each different in its initial background and settings. These initial settings are used to select an appropriate approach that involves both project management and software development standards. These standards define the project’s lifecycle and techniques and tools used to manage it. Each project example spans over four lectures and takes students through all the phases of the project, presenting worked examples of tools and techniques as it goes along.

The course also includes summarizing part that reflects on the two projects and once more reminds the students of the implications of choosing a particular approach to their management. It is important to note that this course does not aspire to show all the possible approaches to IT project management, it does not even indicate that the approaches selected to manage the example projects are the best possible. The main lesson to take out is that the initial project’s background should play an important role in the decision on how to approach the project. Another important lesson to take out is the implications of choosing such approach and what could be the results of a wrong decision (for instance, forcing agile type of communication within a large and diverse team).
In response to the third requirement, twelve multimedia presentations in English were created, one for each lecture of the semester. The design follows the suggestions for an effective worked out example. With regards to the worked example method, all of the 346 slides that the presentations comprise of are carefully and simplistically designed. The contents are well-arranged and multimodal. Apart from text, the presentations include numerous images, tables, graphs, diagrams, quizzes and other visual elements. Each lecture starts with an outline and ends with a quiz on the lecture’s subject matter. All the examples of techniques and tools given are part of the presentations. This is to avoid distraction and to allow the students and the lecturer to focus on one source of information. The way the course progresses and how it introduces theoretical grounds before diving into examples is also inspired by the worked example method. With regards to this method, the effectiveness of learning will be further enhanced by the lecturer by giving instructional explanations when laying the examples. Founding a substantial part of the course on the comparison of two project examples was inspired by another method of example-based learning, the analogical reasoning. Both analog examples manifest the processes that all projects have to go through, like estimating work and identifying stakeholders. The project management know-how that these abstract principles represent is understood better when illustrated on at least two practical examples. More examples can be constructed by the students themselves in seminars. As a part of this thesis, I have suggested the content of such seminars that promotes further practice of given project management techniques.

The course design also reflected on the findings from a content analysis of Project management courses taught at other faculties with IT specialization. Some of these courses focused on the way the delivered product fits into the company’s portfolio and what are the processes that such integration requires. Although I have tried to include this bigger picture into the course’s content, I believe that this topic should be introduced in more depth, if the course’s allotted time allows it in the future. The other Project management course’s content analysis had one important outcome. It helped envision how to make this course more specific for the future IT-specialists by showing how the lifecycle of a software development fits into the processes of project management. The way the course is structured and based on two project examples is what makes it most different than the others, and it hopefully makes it
also more understandable, stimulating and more fitting to the students’ needs. After the pilot run of the course, a collection of feedback is expected to take place in order to show how the course was perceived by the students themselves.
8 Conclusion

The outcome of this diploma thesis has been created similar to the principles it aspires to teach – the activities ranged from collecting requirements, through an analysis that was followed by an iterative design and construction, to the test-run that resulted in further improvements. The result comprises of a coherently structured course that spans over 12 lectures. Each lecture is represented by a set of multimodal slides. The course can be modified by the suggested contents for practical seminars. The multichoice quiz included in each lecture can be used to test the student’s knowledge of the given subject matter.

There are several ways how to create a course on IT Project management. As the content analysis of other project management courses has shown, the subject focus can differ. Most courses concentrate on the lifecycle of the project and its processes. Some courses focus on important knowledge areas, like risk, change and quality. Some courses consist purely of case studies from real companies. All these topics reflect what is important in nowadays Project management. Moreover, as the review of popular standards has shown, all these topics can also be found in numerous guides, frameworks and methodologies. This thesis aims to reflect these popular topics and ways of teaching, but it also takes its own approach. After laying the essential knowledge foundation, it tells a story about two projects. These two projects are different in their initial background and have to be managed accordingly. The chosen management approach influences the whole project and its outcome. The course shows that it is the project manager’s responsibility to decide how these projects are managed as he will be responsible for the outcomes. To model the process and the consequences of such decision, I have chosen two typical approaches to IT project management – one lightweight and agile, one more traditional and predictive. The course shows how these approaches influence the processes that unravel as the project moves along its lifecycle. In a practical and example-based manner, it also shows the tools and techniques that are used to deliver outputs along the way.
Bibliography


Electronic attachments

The multimedia presentations are available as the study materials of the PA179 Project management course taught at the Faculty of Informatics at Masaryk University.