Startup-student collaboration in software engineering student projects:

The introduction of a design thinking workshop

Francis Behnen

Delft University of Technology

TU Delft

Startup-student collaboration in software engineering student projects:

The introduction of a design thinking workshop

by

Francis Behnen

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday April 26, 2024 at 15:00.

Student number:	4498534	
Project duration:	November 2022 – April 2024	
Thesis committee:	Prof. dr. ir. D.M. van Solingen,	TU Delft, first supervisor
	Dr. ir. E. Aivaloglou,	TU Delft, second supervisor
	Ir. T.A.R. Overklift Vaupel Klein,	TU Delft
	Ir. O.W. Visser,	TU Delft

Cover: DALL·E generated cartoon with roughly the prompt "create an image of owls doing a design thinking workshop"
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Summary

Projects in a software engineering (SE) curriculum have many advantages for students. For example, students gain teamwork and project management skills. Involving an external stakeholder can enhance the learning experience further. For instance, students have the opportunity to practice stakeholder management. Moreover, working on a project with a real external stakeholder can help them make informed career choices. This external stakeholder can be a company, a government institution, or an organization from civil society, called 'client' from here onwards. In this work, we focus on startups to investigate the collaboration between startups and students in SE projects. Furthermore, we developed and introduced a design thinking workshop.

We did the research in a project course at Delft University of Technology. It is the capstone course near the end of the major in Computer Science & Engineering. In this course, students take on a project for a 'client'. This external stakeholder brings in the project's objective. Every student group has a unique project from different clients.

We started by analysing these projects from the client's perspective. We interviewed several startups that previously participated in the course. Additionally, we sent a survey to all clients from the previous two years. We incorporated the findings of the interviews and survey with literature. From this, we concluded that startups potentially suffer from a lack of design upfront. This can lead to the creation of an undesirable product. To address this, we developed and introduced a design thinking workshop to our course. Design Thinking is a method for problem analysis and problem-solving. We evaluated the workshop by interviewing teaching assistants who supervised student groups that had participated in the workshop. Additionally, we surveyed the students who participated in the workshop and teaching assistants who joined a test run of the workshop.

From the results of the interviews with startups and survey of previous clients, we synthesized dimensions of success for projects with students. We found three dimensions in which startups measure the success of a project. The output of a project can be important to them in terms of the 1) *product* that is developed or what was 2) *learned* during the project. Additionally, whether students are successfully 3) *recruited* after the project can also be a consideration when assessing success. Furthermore, we synthesized dimensions of product success and preconditions to success in these projects. We found five dimensions of product success and five preconditions to success.

The design thinking workshop was given to students in the first week they started their project. It constituted half a day during which the students learned to apply the Design Sprint method, a method for operationalizing design thinking in a structured way.

After the projects ended, we evaluated the impact of the workshop. The students and their teaching assistants were both generally positive about it. The teaching assistants, who also supervised non-workshop groups, noticed an improved early process. Furthermore, they noticed that some of their non-workshop groups fell into traps that could have possibly been prevented if the group had joined the workshop and applied the tools. This leads us to believe that the workshop should be extended to all students in the course.

Implication for startups Spend time thoroughly understanding the problem before engineering a costly and time-consuming prototype. A design sprint is a low-cost way to discover if the problem is sufficiently understood.

Implications for companies involved in SE project courses If properly motivated, students can create a useful product or prototype, the project does not have to be just a recruitment exercise.

Implications for teachers (of SE project courses with external stakeholders) Add a design sprint workshop for *all* students at the start of the course. This thesis shows that adding such a workshop is feasible and improves students' early process.

Implications for students Do a design sprint before formulating requirements; this will deepen the understanding of the problem to solve and improve the process, so that it yields a product that is more aligned with the user's needs.

Preface

This thesis has been an interesting journey towards finding new knowledge. When I started in the background section, I wondered what my original motivation was for doing this project. I went back to the first email I sent Rini to see if I mentioned anything about my motivation and research directions that I was thinking about at the time.

In that e-mail, I found that my original idea for this thesis was to learn about software engineering processes in small startups beyond creating a MoSCoW requirements list. While leading a small team at an early-stage startup, I noticed I did not have a structure in my software engineering process after we established the requirements. Full-on scrum did not make sense to me, because we were all working part-time, so small, we knew exactly what the others were doing, and it was a greenfield project. I think I reached this learning goal, although the answers are a bit underwhelming. It turns out that start-ups use whatever the tech lead knows, but rarely a “heavy” rigid system. Whenever a one-size-fits-all type of system is introduced, it quickly degrades to something that barely satisfies the requirements of the company for their engineering system.

In that sense, I think my original intuition that scrum was “too much” was justified. Unfortunately, I never found *the* system for software development in (early-stage) startups during my quest...

What I found was much more interesting. It turns out that developing a new software product does not start with a list of requirements. Rather, it is the result of a design process in which you first explore the problem space, then the solution space, and then you zoom in on one solution for which you create requirements.

There are a number of people without whom this thesis would not have existed. First of all, I would like to thank Rini for his enthusiasm when I first approached him and his continued support throughout the process. My priorities were not always with the thesis, but you always managed to pull me back and renew my focus, for which I am grateful. Furthermore, I would like to thank Fenia for joining as second supervisor while we were already underway. Your suggestion of conducting interviews feels like a positive turning point important for the academic quality of this thesis. I greatly appreciated your precise explanation of the process and your elaborate feedback on my many thesis drafts. In addition, Thomas and Otto deserve a big thank you. Without your trust in me to create and give a useful workshop to your students, this thesis would not have been half as interesting.

Privately, I also have some people to thank. In the first place, my girlfriend Corlien. You supported me at any time, even when I tried to take two courses, my driving licence, and do my thesis all at once. Moreover, at the end of the process, you proofread my thesis cover to cover twice. In addition, I would like to thank my other proofreaders: Xiaoling and my mum. Without you, I could not have reached this level of writing! Furthermore, I am thankful for my flatmates. Nander, Jesse, and Sven, it was always great coming home to talk about anything but computer science. I think this was instrumental in keeping my sanity.

Francis Behnen
Delft, April 2024

Contents

Summary i

Preface ii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research aim & setup . . . 2
- 1.3 Background . . . 4
- 1.4 Contributions . . . 5

2 Related work 6
- 2.1 SE processes in startups . . . 6
- 2.2 Projects in SE education . . . 7
  - 2.2.1 SE Education projects with companies from a student perspective . . . 7
  - 2.2.2 SE Education projects from a company perspective . . . 7
  - 2.2.3 Joint perspective on SE Education projects with companies . . . 7
- 2.3 Project failure . . . 9
- 2.4 Design Thinking definitions . . . 9
- 2.5 Integrating Design Thinking (DT), Lean Startup (LS) and Agile Methodologies (AM) . . . 9

3 Method 10
- 3.1 Research Design . . . 10
- 3.2 Context . . . 12
- 3.3 Data collection . . . 12
  - 3.3.1 Exploratory literature study . . . 12
  - 3.3.2 Software startup Interviews . . . 14
  - 3.3.3 Company survey . . . 14
  - 3.3.4 Workshop development . . . 15
  - 3.3.5 Teaching assistant survey . . . 16
  - 3.3.6 Student survey . . . 16
  - 3.3.7 Teaching assistant interviews . . . 16
- 3.4 Data analysis . . . 17
  - 3.4.1 Data analysis literature . . . 17
  - 3.4.2 Client survey . . . 17
  - 3.4.3 Student survey . . . 18
  - 3.4.4 Teaching assistant survey . . . 18
  - 3.4.5 Teaching assistant interviews . . . 18

4 Results 19
- 4.1 RQ1: What are the dimensions of success in SE student projects at software startups? 21
  - 4.1.1 Software engineering success in literature . . . 21
  - 4.1.2 Survey results . . . 22
- 4.2 RQ2: Which preconditions make SE student projects in software startups successful (or not)? . . . 22
- 4.3 RQ3: What are the dimensions of product success in SE student projects? . . . 23
- 4.4 RQ4: To which extent can a Design Sprint workshop increase the (product) success of software engineering student projects for startups? . . . 24
  - 4.4.1 Teaching assistant survey . . . 24
  - 4.4.2 Student survey . . . 25
  - 4.4.3 TA interviews . . . 26
- 4.5 Update educational material . . . 29

5 Conclusion & Discussion **30**
5.1 RQ1: What are the dimensions of success in SE student projects at software startups? 31
5.2 RQ2: Which preconditions make SE student projects in software startups successful (or not)? . . . 31
5.3 RQ3: What are the dimensions of product success in SE student projects? . . . 32
5.4 RQ4: To which extent can a Design Sprint workshop increase the (product) success of software engineering student projects for startups? . . . 32
5.5 Discussion . . . 33
5.6 Limitations . . . 35

6 Future work **37**

References **38**

A Startup interview questions **42**
A.1 Company overview . . . 42
A.2 Project Overview . . . 43
A.3 Supervision . . . 43
A.4 Result . . . 44
A.5 Evaluation . . . 44

B Company survey Software Project **45**
B.1 Introduction and organizational features . . . 45
B.2 Project . . . 46
B.3 Supervision . . . 47
B.4 Result . . . 48
B.5 Evaluation . . . 49

C Software Project expectations TAs **50**

D Student evaluation questions **53**

E Teaching assistant interview questions **55**
E.1 Organizational . . . 57
E.2 Workshop evaluation . . . 57
E.2.1 Did a design sprint . . . 58
E.2.2 Didn't do a design sprint . . . 58
E.3 Workshop iteration . . . 59
E.4 Personal information . . . 59

F Brightspace self-learning module **60**

G Brightspace facilitator module **71**

1 Introduction

This thesis concerns collaboration between startups and students in software engineering (SE) student projects. The research was performed around our bachelor's capstone project course. The projects are proposed and supervised by a client. This client can be a company, a government institution, or an organization from civil society. We interviewed startups and surveyed organizations that previously participated as client in the course. Furthermore, we developed and introduced a design thinking workshop.

Our interest in startups is two-fold. Startups are increasingly a driving force of innovation in the Dutch economy. Furthermore, as an industry, they grow consistently faster than the Dutch economy as a whole[1], growing more than twice as fast as average in most years. In addition to the economic factor, startups are also an important tool in the valorisation process of universities. Universities are increasingly focused on valorising their research. Bringing the research results outside the university confines and into the world is often done by spinning off a startup that brings some intellectual property of the university to the market. In short, startups are increasingly pivotal for universities and the broader society.

This work focuses, in particular, on software startups. At some universities with a Computer Science (CS), Software Engineering (SE) or related programme, university and companies work together in a (capstone) project course. Such projects can be particularly interesting for early-stage startups, as they usually lack workforce and other resources. For students, it is usually a course where all the other subjects of the programme come together. A place where they can apply all their newly acquired skills in practice. This thesis seeks to improve the collaboration between startups and students in such projects.

1.1. Motivation

For students, there are many advantages to a project course in their curriculum. Working together with their peers, they learn to work in a team [1]–[5] and improve their communication skills [2], [6]. They also learn how to plan and manage such a software project [1], [5], [6]. When the project is full-time, with no classes running parallel to it, the size of the project can be substantial and yield learning beyond that of other courses. During such a project, students will inadvertently encounter issues they have not seen before. They will face challenges that are hard to learn from a book or require knowledge not taught in their curriculum. These challenges foster students' problem-solving skills [3], [6], [7].

In this work, we focus on project courses with external stakeholders. This external stakeholder brings in their project idea and acts like a 'client' or 'customer' to the team. The 'client' can be a company, a government institution, or an organisation from civil society.

Involving an external stakeholder provides additional benefits. For example, it has been reported to increase student motivation [4], [8]. Additionally, it allows students to practice communicating with an external stakeholder. Furthermore, because it is a 'client' with real-life needs, it provides a more realistic experience of practical elements of a software engineering project [9]. The collaboration can

[1] *Netherlands Startup Employment 2022 Report*, en. [Online]. Available: `https://www.techleap.nl/reports/netherlands-startup-employment-2022-report/(visitedon10/09/2023)`.

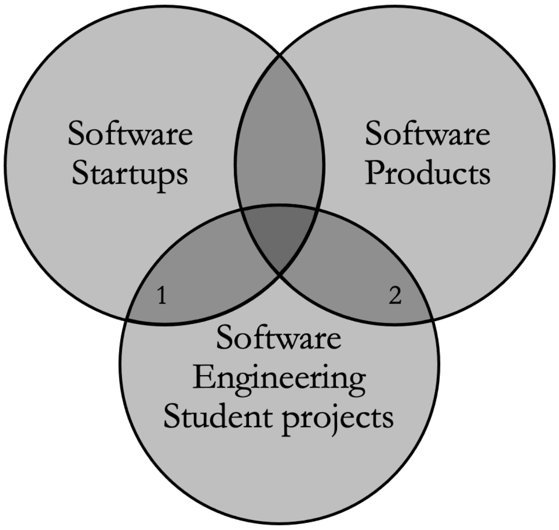


Figure 1.1: Mental model of research area with areas of interest

also serve as a qualification for future employment [1], [5], [10]. This is a benefit that does not directly increase the programme's educational level but does increase its usefulness to the student.

Interviews with clients of such a project course confirm this. Most clients mention recruitment as the most important reason to participate [8]. Developing the software product is only a secondary goal for most of them. This may not be a surprising result. On the other hand, a team of SE students working full-time at the end of their degree should be able to create something substantial. Interestingly, for small companies, the primary and secondary goals are reversed in [8]. First and foremost, the software product is important for them, and then, whether they can hire the students. The third reason clients participate in university SE project courses is to pilot new technologies, according to [8]. The fourth overall priority clients mention is to gather new ideas.

This client perspective in the collaboration with a university is a neglected research area. To the best of our knowledge, only three papers exist that discuss it [8], [11], [12]. It remains unclear why. Specifically with startups, it could be challenging to find startups willing to cooperate [13]. This could be a broader theme. However, in the case of a university course that has been running for a few years, there is already an established collaboration. In our experience, most of the supervisors on the client side were not difficult to reach and were eager to share their perspectives. They did not need to be convinced that reflecting and evaluating the collaboration would be beneficial.

1.2. Research aim & setup

Figure 1.1 depicts the mental model we used to frame our research. It shows that we are interested in the area where SE student projects, software startups and software products overlap. Our interest in SE student projects and software startups is introduced above. Many aspects of this could be researched. For example, one could try to find out how to optimize the hiring rate of the students after a project at a company. This is not something we are interested in for this work. The area we are interested in has to do with the software products themselves. We want to optimize for output that is useful to the company, in line with the priorities of smaller companies found in literature [8] and in our own interviews with startup supervisors. In our mental model, we introduced the 'Software Products' area to avoid aims like optimizing the hiring rate or other goals unrelated to the products themselves.

Our first research questions are in area 1 of the mental model, where SE student projects and software startups overlap. Similarly, in area 2, we are interested in the overlap of SE student projects and software products.

In area 1, where SE student projects and software startups overlap, we formulated the following research questions (RQs):

- RQ 1: *What are the dimensions of success in SE student projects at software startups?*

  We are interested in knowing how the success of a project is defined at startups. A project is not necessarily successful only if a fully functional product is delivered. Due to the strict time limits on courses, they have to follow the university calendar after all, there could be other success dimensions. This research question intends to find those different dimensions of success.
- RQ 2: *Which preconditions make SE student projects in software startups successful (or not)?*

  After finding out how success is defined in RQ 1, RQ 2 intends to determine which preconditions and contextual factors are important to make a project successful.

In area 2, where student projects and software products overlap with a focus on startups, we formulate the following research question. Mirroring RQ 1, we intend to find out how product success is defined and reached in SE student projects:

- RQ 3: *What are the dimensions of product success in SE student projects?*

  In this question, we are interested in the dimensions that define a successful product in SE student projects, especially from the startup's point of view. Intuitively, a fully functional production-ready app is not necessarily the only definition of success, as time is limited. The difference from RQ 1 is that this question examines the product specifically, whereas RQ 1 examines the project as a whole.

In the middle of our mental model, where SE engineering student projects, software startups and software products come together, we have a fourth research question. This question requires more introduction.

During the research for the first three research questions, we found that short-term success goals are often met in our sample. However, by digging deeper into startups' individual projects, we uncovered a few relevant points for every type of stakeholder. These startups usually are still figuring out what their product should be, their user base, and their business model. Getting this right is vital to the survival of the company. The literature shows that a product without market need is one of the main reasons for software project failure [14]. This is a real risk for technical teams and stakeholders in the software project. [15] suggests, based on Steve Tobak[2], that a team too focused on technical aspects and entrapped in their product vision can build products that do not solve real problems; a product without market need. According to Ximenes, Alves, and Araújo [15], Aulet [16] put it even stronger: "[it] explains the major pitfall in the teams' inability to put themselves in the users' shoes, truly understanding the way they think and what they need" [16] via [15].

There are several reasons to believe that the student teams in the course could be too focused on technical aspects. At the start of the project, the students have to hand in a project plan. This report has a compulsory structure from the course. It requires students to complete the entire problem analysis in one chapter and spend five chapters on feasibility, development methodology, and other aspects of the technical project's execution. This stems from the report structure's origin as a technical report. Additionally, the course teachers have told us that students only use feasibility frameworks like scrum, waterfall, or spiral during their projects. A limitation of scrum repeatedly mentioned in the literature according to [17] is a lack of attention to design. This, in turn, can lead to project failure later, according to [18] citing [19].

To ensure long-term project success, the project's desirability (user) aspect should also be considered. This is relevant to startups and every other type of client in the Software Project course. Design thinking can be a framework for this in a professional setting [18]. Fortunately, undergraduate CS students can also do design thinking work [15]. Multiple frameworks integrate design thinking and scrum by starting a design thinking process that morphs into scrum during the project [15], [18], [20]. Ultimately, the 'Design Sprint' framework was chosen.

In summary, short-term success goals are often met, but the literature suggests these projects could fail in the long term. Technical teams are prone to get trapped in their vision of a product without

[2] S. Tobak, *9 reasons why most startups fail*, Jan. 2014. [Online]. Available: `https://www.entrepreneur.com/leadership/9-reasons-why-most-startups-fail/231129`.

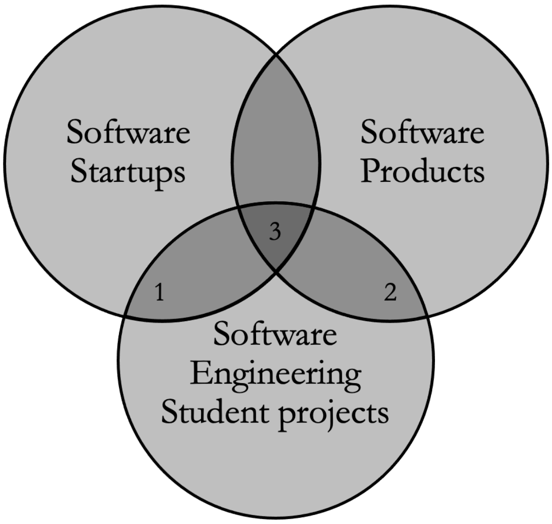


Figure 1.2: Mental model of research area with an additional area of interest

sufficiently considering the user. To prevent this, the team should take a step back, think about the intended users, and investigate their most pressing problems. The student group can do this as a first step in their project. To support them, a Design Sprint workshop was developed to teach them this framework at the start of their project. This resulted in the following research question:

- RQ 4: *To which extent can a Design Sprint workshop increase the (product) success of software engineering student projects for startups?*

  This research question aims to determine if a design sprint workshop leads to higher product success. We will look at product success directly and the factors that influence product success. Additionally, we will evaluate the workshop in general.

This research question sits in the middle of our mental model. The workshop is an intervention in SE student projects that aims to improve the product success of software startups. This is visualized in figure 1.2; RQ 4 is in area 3, where SE student projects, software startups, and software products overlap.

With the word ‘product’ in the research question, we intend to indicate the artefact produced by the students during the project. This is not necessarily always a product in the sense that it is an application that can be rolled out to users. It can also be simulation or modelling software where the company is interested in the simulation result, not necessarily in developing the code further and releasing it as a product. Therefore, instead of ‘product’, a more correct term would be ‘artefact’. We used the word product anyway because we think it enhances the readability. The produced artefact is also a product in most projects, so the difference is only semantic in those cases. Only in a limited number of projects an artefact is produced that we would not call a ‘product’.

1.3. Background

Collaborating in a university project course can be a great opportunity for software startups to improve their progress and productivity. However, guiding such a project group of young, inexperienced students is not trivial. There are only a handful of articles in the literature that describe the company’s perspective on such projects. To the best of our knowledge, none of them focus on startups specifically.

At Delft University of Technology, the bachelor of Computer Science and Engineering[3] has a capstone-like full-time 10-week course in the second year of the curriculum that constitutes a software project at a client. This provides a great opportunity to study the phenomenon described above, as some of those clients are startups.

[3] Although it is not obvious from the name the bachelor is mainly Software Engineering. Computer Science and Software Engineering is arguably a better name.

1.4. Contributions

This work makes the following contributions:

- It contributes a design sprint workshop that provides structure to the problem analysis phase of software engineering projects and makes students more aware of the user's perspective. The workshop explains the design sprint framework and provides students with tools to apply it independently (on their project). The workshop was updated based on the results of the first iteration.
- It refines the understanding of SE student projects at startups by contributing dimensions of success from the startup perspective and listing preconditions to make a project successful.
- The results indicate that SE students and software startups can possibly benefit from design thinking, conceivably extending to SE projects in general.
- It shows that a project course with industry is a viable path to reach startups for scientific research. Contributing a new method to mitigate the issue of too few startups willing to cooperate, signalled by [13].

2
Related work

This chapter discusses existing software engineering (SE) literature in startups and SE education projects.

2.1. SE processes in startups

To understand the context in which students will work in startups, this subsection discusses the SE processes – or lack thereof – used in startups.

In small organizations, “one-size-fits-all” software processes have difficulty penetrating [21]. It turns out that most startups use only light SE processes or have no structure at all. However, [21] also suggests this may not be a feature exclusive to startups.

When rigid processes are introduced, startups experience process erosion. So even when “heavier processes” were tried, the startup often reverted to a lighter process, removing all elements from the original process seen as unnecessary [22]. The experience of the co-founding team and technical lead mainly influences the choice of process. There are no resources to explore the best way to develop software, so they take what they know and support their immediate business objective [22], [23].

This sentiment is echoed in the Greenfield Startup Model by Giardino *et al.* [24]. They researched startup engineering strategies and built a model describing software development in early-stage startups. One of their conclusions in the model and while developing it is that everything in a startup revolves around the need to speed up development. This is hypothesized to be due to a severe lack of resources.

However, this view of startups under constant resource pressure with limited formal processes is challenged. Klotins *et al.* [25] set out a survey among many startups to identify goals, challenges, and practices in startups at different stages of their life cycle. One of the things they expected to find in the survey results is a constant lack of resources, especially in the early stages of the life cycle. However, only a few startups responded that resources or time were a continuous issue. This counters the prevailing wisdom in (computer science) literature that one of the defining features of a startup is constant time and resource pressure. Klotins *et al.* furthermore note that the trade-off between project scope and budget, regardless of the experienced pressure, is also not a specific issue for startups. Instead, it is a feature of project management in general. Therefore, most studies exploring software project success factors report it as a concern for software projects in general [26]–[29].

In follow-up work, Klotins concludes that the assumption in the existing literature that startups are somehow “unique” and therefore need different engineering practices is wrong [30]. Rather, the rapid evolution of the product and organization and conflicting stakeholder objectives could add additional complexity to the engineering process. In addition, although there is no constant resource pressure, there is still limited room for error in terms of budget and time overruns. For a large organization, this would mean one failed project; in the case of a startup, it could mean the end of the company as a whole. Klotins concludes this means startups should be more, not less, focused on using the best engineering processes and practices [30].

2.2. Projects in SE education

Extensive research exists on the benefits for students of projects in software engineering education. Stahl, Sandahl, and Buffoni recently published a thoroughly researched overview of the benefits reported in the literature. On the other hand, hardly any article describes the benefits of participating in such a project for a company or another type of external client [12].

This section discusses the benefits for students first, followed by the benefits reported for clients, and the last subsection is about the model developed by Stahl, Sandahl, and Buffoni that takes a holistic view by joining both perspectives and looking at it as an ecosystem [12].

2.2.1. SE Education projects with companies from a student perspective

Most of the reported types of skill development are soft skills. For example, teamwork is reported several times [1]–[5], as well as project management [1], [5], [6]. In addition to that, some publications mention problem-solving and critical thinking [3], [6], [7], and others find that self-directed learning skills are developed [1], [31]. In addition, internal and external communication between teams has been reported to improve in some publications [2], [6].

In addition to that, it is associated with:

- more positive attitude towards the SE discipline [7],
- supporting students in making career decisions [1], [3], [10],
- entrepreneurship [4],
- scrum [8],
- increase students' motivation [4], [8], and
- a project course can serve as a qualification for future employment [1], [5], [10].

2.2.2. SE Education projects from a company perspective

In contrast to the well-researched student perspectives, to our knowledge, only two articles discuss the company perspective. Host, Feldt, and Luders [11] looked into the master theses at their faculty that were done in collaboration with industry, conducting interviews with students, industrial supervisors and faculty staff. They find the classic tension between the industry supervisor, who wants concrete, usable results, and the academic supervisor, who is more focused on comparing techniques and building new theories [11]. It would be interesting to see how this difference in stakeholder objectives plays out in courses earlier in the curriculum. One can imagine, for example, that a project course with learning objectives that are not related to the project outcome, e.g., soft skills, is more compatible with an industrial client who would like to see results.

The only paper that takes an explicit industry perspective is "Collaborating with Industrial Customers in a Capstone Project Course" by Paasivaara, Vanhanen, and Lassenius [8]. To better understand the needs of industry partners for their capstone course, they interviewed all the clients from one of the editions of their course. For their customer companies, in order of importance and frequency, the goals or reasons for having a student team from the course are: 1. recruiting, 2. develop the software, 3. research and pilot new technologies, and 4. fresh ideas.. Advice about the project setup from the interviewed companies for other companies: 1. [have a clear] vision, 2. suitably important, 3. realistic with a challenge, 4. motivating, 5. illustrates the work in the company.

They also have practical advice for companies. The companies interviewed emphasized that spending enough time actively collaborating with students is essential, especially in the beginning. Most companies achieved this by inviting students to their offices at least once a week. Some companies that did not do this reflected that this would have helped in the collaboration and that they would ensure there was space for students in the next edition of the course.

2.2.3. Joint perspective on SE Education projects with companies

While the benefits of a larger project in the SE curriculum are well described, there has been little attention in the scientific literature for the client perspective. This is curious since such courses cannot exist without the client's projects. Besides that, if the project is executed well, they stand to gain significant value if a team of end-bachelor SE students comes working for them for multiple months. A well-executed project, in turn, can inspire the client to provide a new project the following year.

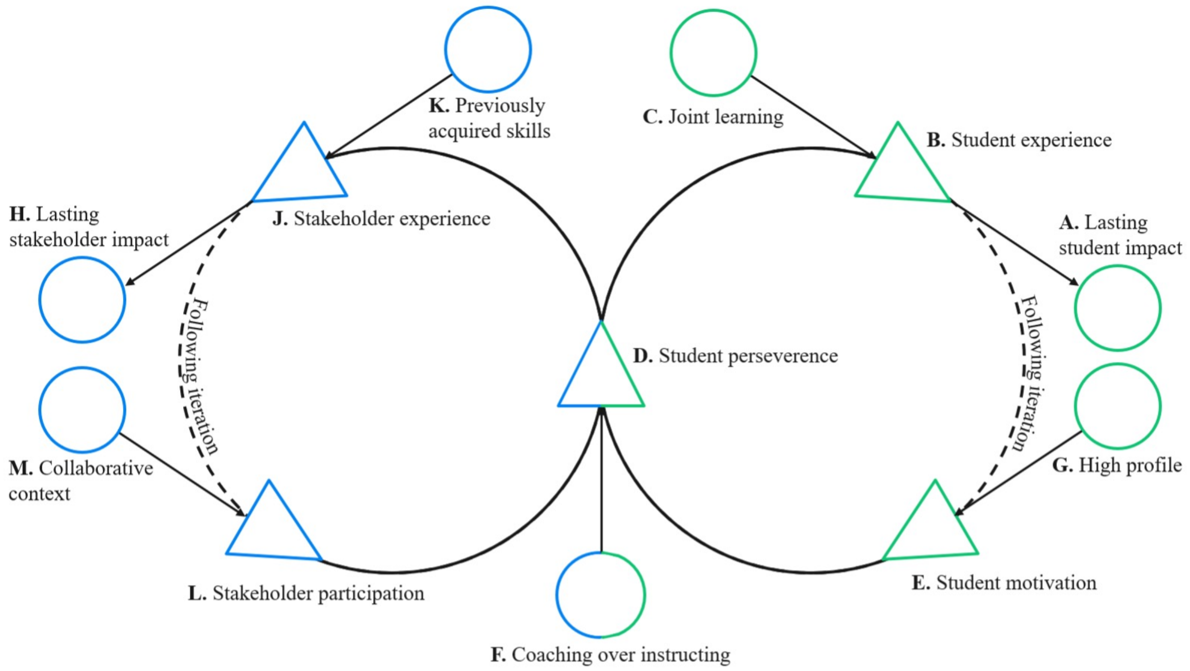


Figure 2.1: Reinforcing loops in the student-stakeholder-university ecosystem by Stahl, Sandahl, and Buffoni, *taken from [12]*. Their definition of 'stakeholder' is similar to what we call 'client' in this thesis.

Stahl, Sandahl, and Buffoni [12] take an explicit joint perspective of both the student and the client. They refer to it as the stakeholder perspective. Their model centres around student perseverance. When the student group perseveres, they will create a positive experience for themselves and their client. Students will persevere when motivated by the project. This will happen in two ways:

1. a positive experience from last year's cohort, and
2. a high profile of the course in the curriculum [12].

When a large majority of students have a positive experience with a course, this will create a "buzz" around the course and will have students looking forward to doing this course. [12] found that this "buzz" around their project course with clients contributed to the motivation of the students to achieve a good performance in the project. In this way, a positive experience for the students is vital for success in the next year.

The authors also found that the fact that the course stood out from the rest of the curriculum contributed to the student's motivation. The course has a high profile because it occupies an entire block without any other courses next to it. Next to that, it is together with an actual client, unlike most, if not all, other courses.

There are two other inputs to student perseverance according to this model. The fact that the teachers do not instruct but coach motivates the students to take ownership of their problem. Teachers are not there to tell students how to do the project. Instead, they make suggestions on how to approach a problem. In this way, the students must independently figure out what to do. This makes them the owner of the problem and the solution. With this responsibility, there is no one to turn to but themselves. This makes them persevere in the face of adversity.

In addition to coaching instruction and student motivation, there is a third input to student perseverance. This is client participation. In a way, this client's side mirrors the student's side. [12] found two things that lead to (more) client participation:

1. a positive experience in the previous year, and
2. a collaborative context.

Clients who had a good experience the previous time are more likely to invest energy into the next project. This can translate into more participation in the project.

In addition, in the case of [12], there is a broader context of collaboration between the university and the local industry. For example, some participating companies also serve on university committees. The authors find that this biases the companies towards participation, too.

The positive experiences by both students and clients lead to reinforcing loops, according to [12]. In addition, they found other factors that influence the experience of both groups. They synthesized the model depicted in figure 2.1 from these findings. This shows a holistic view of the project course and views it as an ecosystem where both student experience *and* client experience (impact) are important.

2.3. Project failure

A systematic literature review by Dingsøyr, Nerur, Balijepally, *et al.* [17] finds that a limitation of scrum repeatedly mentioned in the literature is a lack of attention to design. This can lead to the launch of the "wrong" product or poor market reception [19]. Furthermore, "no market need" is attributed as the most prominent software project failure, according to [15] based on an article by Erin Griffith.[1] [15] attributes the most likely underlying reason to teams becoming too focused on technical aspects and caught in their product vision based on Steve Tobak.[2] A peer-reviewed study of 214 postmortems [14] also observed "a high focus on the product or service by the management and founders, but insufficient attention to commercial development" behind many of the startup failures.

2.4. Design Thinking definitions

Design thinking has many definitions. Traditionally, it was viewed as an approach to solving complex or wicked problems [32]–[35]. In this view, the approach achieves greater project success by successfully tackling these problems. A recent study counteracts this view. It followed 160 students in innovation projects for companies and suggests that design thinking is not necessarily a good approach to complex or wicked problems. Instead, success in design thinking projects should be attributed to the facilitation of motivation and empowerment [36].

Tim Brown, long-time chair of IDEO and author of the book that popularised design thinking 'Change by Design', defines design thinking as "a set of principles that can be applied by diverse people to a wide range of problems" in his book. The book 'Design Thinking' [37], edited by Hasso Plattner, gives a definition based on science. "Design thinking is a creative, individual-level process influenced by social-level factors (that is, high inspiration by others, high user-centricity, high prototyping, and low criticism by others), which includes attention, memory, and learning and leads to an aesthetically appealing object." [38]

2.5. Integrating Design Thinking (DT), Lean Startup (LS) and Agile Methodologies (AM)

Several studies have aimed to integrate DT, LS, and AM [15], [18], [20], [39], [40]. "[They] converged when observing that design thinking supports lean startup with moments of empathy and user understanding" [41]. The motivation for combining the different frameworks often aligns with the previous section on project failure; using only agile or scrum will lead to undesirable products. [15] and [20] show that undergraduate software engineering students can execute such an integrated framework.

[42] criticizes the lack of inclusion of design thinking in academic programmes outside design courses.

[1] E. Griffith, *Why startups fail, according to their founders*, Sep. 2014. [Online]. Available: `https://fortune.com/2014/09/25/why-startups-fail-according-to-their-founders/`.

[2] S. Tobak, emph9 reasons why most startups fail, Jan. 2014. [Online]. Available: `https://www.entrepreneur.com/leadership/9-reasons-why-most-startups-fail/231129`.

3 Method

3.1. Research Design

This thesis aimed to investigate the collaboration between startups and students in software engineering (SE) student projects. To do this, we chose the course in our own bachelor Computer Science & Engineering at Delft University of Technology. This is a capstone course in the curriculum. It is a full-time course and placed after all mandatory non-elective courses, the “cornerstone courses”. More information about the course is given in the context section (3.2).

We employed a design cycle method consisting of five stages: define the problem, understand the design space, ideate, build, learn, and repeat. Depicted in figure 3.1. Several research methods built towards the first iteration of the design, and the remaining methods were to evaluate the first design.

To *define the problem*, first scientific literature was consulted. After that, an exploratory set of interviews was held with startups that had previously been clients of the Software Project course. To quantify the ideas gathered from the interviews, a survey was developed and sent to all ex-clients of the previous two years of the course.

From this analysis, we concluded that teaching students essential design thinking elements could lead to more valuable results for startups and other clients. The first iteration of the design thinking educational material was a workshop. This workshop was tested by giving it to several student groups in the next edition of the course. The educational material was evaluated, and the evaluation was used as input for a second iteration of the educational material.

The educational material was evaluated in three ways.

1. Before the workshop was given to students, it was tested with the course’s teaching assistants. After this session, they were asked to complete a survey on the differences they expected to see between groups that would and groups that would not participate in the workshop.
2. After the course ended, all groups that participated in the workshop were asked to complete a survey about the workshop and its usefulness in the course.
3. Next to that, their teaching assistants were invited for an interview to talk about the differences they observed between workshop and non-workshop groups.

The timeline of the different data-gathering activities is summarized in Table 3.1 together with the workshop. Next to these data activities, several different literature categories were explored:

- Software engineering in software startups
- Projects in software engineering education
- Project failure and startup failure
- Agile and design thinking integration frameworks

The first two categories informed the exploratory interviews and the survey of companies. The other two categories of literature provided the theoretical underpinnings for the workshop.

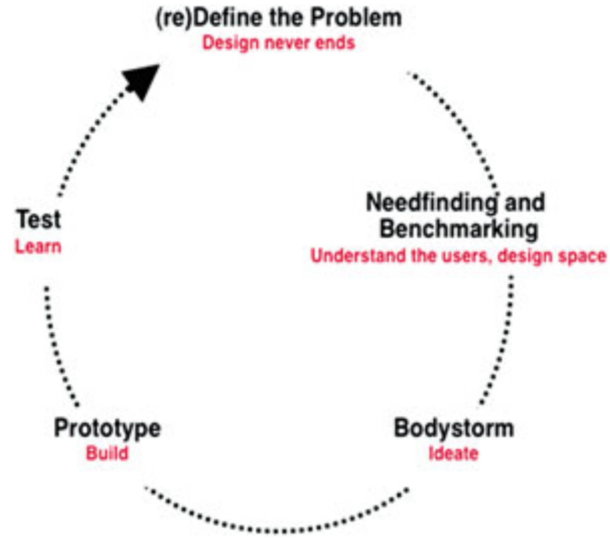


Figure 3.1: Design cycle by Plattner, Meinel, and Weinberg [37].

Timespan	Activity
Dec. 2022 - Jan. 2023	Exploratory interviews startups (pre-workshop)
3 Feb. 2023	Survey to companies (pre-workshop)
21 Apr. 2023	TA survey (at workshop test run)
25 - 27 April 2023	Workshops
July 2023	Student survey (post-workshop)
Nov. 2023	TA interviews (post-workshop)

Table 3.1: Timeline of data gathering activities together with the workshop.

3.2. Context

Most of the research was conducted around the Software Project (SP) course in the Bachelor of Computer Science and Engineering[1] at Delft University of Technology. This bachelor's degree is nominally three years long and has a mandatory minor in semester 1 (S1) of the third year (Y3). It ends with an individual research project in quarter 4 (Q4) of that same year. SP is a full-time course (15 ECTS[2] credits) in the fourth quarter of the second year (Y2Q4). Due to this placement, students have completed all of their shared subject courses, the cornerstone courses of the curriculum. Only three specialisation courses remain in Y3Q3. For this reason, the SP is comparable to a capstone project course at the end of a bachelor's curriculum. It serves the same function, allowing students to apply all their newly discovered knowledge at the end of their major.

Furthermore, [43] describes a software engineering capstone course in the following way, based on the ACM/IEEE Curriculum Guidelines for Software Engineering (SE) Degree Programme[3]: "They can be characterised as long and substantial projects that should preferably be completed in a team. Projects should have customers for whom students are expected to deliver some form of real implementation at the end of the course. Students should therefore engage in real software development activities and not just complete simple, theory based assignments provided by the teacher. The evaluation of the project results should focus not only on the fact that the project 'works', but also on the deliverables on how well they have been completed. Finally, the focus of the course and its assessment should be on software engineering practices and processes, and students should give adequate opportunities to reflect on the experience."

An external stakeholder always commissions the project. The stakeholder can be a company, public institution, or organisation from civil society. In exceptional cases, the stakeholder can be internally from the department, but that rarely happens. In this work, we refer to this external stakeholder as the 'client' of the students, of the student group, or *in* the course.

3.3. Data collection

This section describes the data collection process of every research activity where data was collected. Furthermore, it describes the workshop development process. The research activities were:

1. exploratory literature study,
2. software startup interviews,
3. client survey,
4. workshop development,
5. teaching assistant survey,
6. student survey, and
7. teaching assistant interviews.

Table 3.2 in the data analysis section relates these research methods to the research questions it helped answer.

3.3.1. Exploratory literature study

In the exploratory literature study, two different themes were explored to learn more about the current understanding of the literature on these two topics.

- Software startup literature.
- SE project education literature.

Both themes were investigated using the forward snowballing and backward snowballing technique.

[1] The name of the bachelor is a bit of a misnomer, implying that it is mainly about computer science and computer engineering, which is incorrect. Most of the courses are in computer science or software engineering

[2] European Credit Transfer and Accumulation System

[3] ACM/IEEE, ACM/IEEE joint task force on computing curricula: Software Engineering 2014: Curriculum guidelines for undergraduate degree programmes in software engineering, 2014, URL `https://ieeecs-media.computer.org/assets/pdf/se2014.pdf`.

Software startup literature was read to understand how much the scientific community generally understands software development in startups. To this end, we were interested in models describing software engineering in startups and surveys presenting an overview of the status quo of software startup literature.

The study started by searching for relevant papers on the Scopus and Web of Science indexing databases. To start the snowballing process, a set of initial papers was selected from the Scopus and Web of Sciences databases according to the following process. We searched for "software startup model" within the "computer science" subject area and further limited the results to articles including the keywords "software startup", "software startups", "startup", or "startups". The exact query on Scopus was:

```
TITLE-ABS-KEY ( "software" AND "startup" AND "model" ) AND ( LIMIT-TO ( SUBJAREA , "COMP" ) ) AND ( LIMIT-TO ( EXACTKEYWORD , "Software Startups" ) OR LIMIT-TO ( EXACTKEYWORD , "Software Startup" ) OR LIMIT-TO ( EXACTKEYWORD , "Startup" ) OR LIMIT-TO ( EXACTKEYWORD , "Startups" ) ).
```

The exact query on Web of Science was:

```
(((AB=(software startup model)) OR TI=(software startup model) OR KP=(software startup model))) AND (QMTS=("Software startup") AND QMTS=("Software startups") AND QMTS=("startup") AND QMTS=("Startups") AND TASCA==("COMPUTER SCIENCE SOFTWARE ENGINEERING" OR "COMPUTER SCIENCE THEORY METHODS")).
```

The same procedure was followed to find surveys of the software startup research area. The search query used was 'software startup (survey OR "mapping study")', again within computer science and limited to the same keywords supplemented with 'survey', 'surveys', and 'systematic mapping study'. The exact query on Scopus was:

```
TITLE-ABS-KEY ( software AND startup AND ( survey OR "mapping study" ) ) AND ( LIMIT-TO ( SUBJAREA , "COMP" ) ) AND ( LIMIT-TO ( EXACTKEYWORD , "Surveys" ) OR LIMIT-TO ( EXACTKEYWORD , "Survey" ) OR LIMIT-TO ( EXACTKEYWORD , "Systematic Mapping Studies" ) OR LIMIT-TO ( EXACTKEYWORD , "Software Startups" ) OR LIMIT-TO ( EXACTKEYWORD , "Software Startup" ) OR LIMIT-TO ( EXACTKEYWORD , "Startup" ) OR LIMIT-TO ( EXACTKEYWORD , "Startups" ) ).
```

The exact query on Web of Science was:

```
(((AB=(software startup (survey OR "mapping study"))) OR TI=(software startup (survey OR "mapping study")) OR KP=(software startup (survey OR "mapping study"))) AND ((QMTS=("Software startups")) OR (QMTS=("startup")) OR (QMTS=("Startups")) OR (QMTS=("Software startup")) AND TASCA==("COMPUTER SCIENCE SOFTWARE ENGINEERING" OR "COMPUTER SCIENCE THEORY METHODS")) AND ((QMTS=("Survey")) OR (QMTS=("mapping study")))).
```

The lists were further refined by reading the articles' titles and abstracts and excluding papers that were not models or surveys of software startup literature. The key papers while snowballing were:

- "Software Startups - A Research Agenda" by Unterkalmsteiner, Abrahamsson, Wang, *et al.* [13],
- "Software Development in Startup Companies: The Greenfield Startup Model" by Giardino, Paternoster, Unterkalmsteiner, *et al.* [24], and
- "Software startup engineering: A systematic mapping study" by Berg, Birkeland, Nguyen-Duc, *et al.* [44].

SE project education literature was used to find project courses similar to the one researched here. We intended to find companies' definitions of success in those articles and compare them with our survey and interview results. Furthermore, we were looking for success definitions of student projects.

The set of initial papers for this theme was found by searching for 'software engineering project course client' in the computer science subject area limited to the last ten years. The exact query on Scopus was:

```
TITLE-ABS-KEY ( software AND engineering AND project AND course AND client ) AND PUBYEAR > 2012 AND PUBYEAR < 2024 AND ( LIMIT-TO ( SUBJAREA , "COMP" ) ).
```

The exact query on Web of Science was:

```
(AB=(software engineering project course client) OR TI=(software engineering project course client) OR KP=(software engineering project course client)) and Computer Science Software Engineering (Web of Science Categories) and 2023 or 2022 or 2021 or 2019 or 2018 or 2017 or 2016 or 2015 or 2014 or 2013 (Publication Years).
```

The list was further refined by reading titles and abstracts and excluding papers that were not about software project courses with an external client. Key papers were:

- "Software Engineering Project Courses with Industrial Clients" by Bruegge, Krusche, and Alperowitz [2],
- "Involving external stakeholders in project courses" by Steghöfer, Burden, Hebig, *et al.* [45], and
- "An Eco-System Approach to Project-Based Learning in Software Engineering Education" by Stahl, Sandahl, and Buffoni [12].

3.3.2. Software startup Interviews

Exploratory interviews with software companies were conducted to understand the current collaboration between startups and students. The protocol is in appendix A. The interview consisted of questions on five topics. 1) The interviews started with questions about the company to check if it was really a startup. After that, there were questions about 2) the project itself, 3) how supervision was organised from the company's side, 4) the result of the project, and 5) questions to reflect on the project. The project questions allowed the interviewee to explain what the students did. In the supervision section, we could get into the collaboration with the students. Questions on the result were asked to gauge the company's satisfaction with the project and the delivered artefact. The questions were a mix of open, multiple choice, and Likert scale questions.

Participant selection & recruitment All companies that could be startups that participated in the 2022 edition of the software project course were considered. This amounted to around fourteen. As definition for startup, we used the definition of [25], except that a company could sell multiple products/services and still be considered a startup.

Of this, one project was discarded because the author of this thesis was the supervisor from the startup's side. Another two were discarded because they turned out to be large organisations, more akin to scale-ups.

The remaining eleven companies were invited for an interview by the course staff. Eight companies accepted the invitation.

Interview process The interviews were semi-structured; the interview protocol can be found in appendix A.Except for the first, they were conducted online. The first three interviews were used to refine the protocol. All online interviews are recorded in audio.

3.3.3. Company survey

A survey was conducted with past course participants to confirm the insights obtained from the initial interviews. The purpose of the survey was to validate the information gathered from the interviews for a larger group. The questions can be found in appendix B or at `https://forms.office.com/e/RK6EUFwXRb`. The survey included questions about collaboration with the students relevant to this research. Furthermore, the survey included questions to evaluate the course more broadly from a company perspective. The results of these questions were shared with the course staff for their evaluation.

The survey used the same structure as the interviews and, to a degree, the same questions. It started with questions about the client to check the type of institution. After that, there were questions about the project's subject, how supervision was organised from the company's side, the project's results, and questions to reflect on the project. From the project questions, we wanted to get an idea of the types of projects that were being done. In the supervision section, we asked about the day-to-day of the project and the background of the (technical) supervisor. For instance, we wanted to know if there was office space available to the students and how often they would be there. The result section contained questions that asked how the result of the project was used afterwards and whether the client thought the output made the project worthwhile.

Participant selection & recruitment All clients of the previous two years were selected. This allowed for a broad range and, at the same time, recent perspectives.

The questionnaire was sent along with a call for new projects to all participating companies in the previous two years. The course administrators sent this email.

Twelve companies responded to the questionnaire. The form can be found in appendix B or at `https://forms.office.com/e/RK6EUFwXRb`.

3.3.4. Workshop development

This section describes how the design sprint workshop was developed. The introduction chapter details the reasoning behind doing this workshop. The self-learning module students should work through before attending the workshop is added as appendix F. Furthermore, a facilitator document explains the idea and learning goals behind the workshop in more depth. This document is in appendix G.

Workshop design For the workshop, a design thinking framework was chosen that originates in the software engineering industry: the design sprint [46][4]. In addition to its origins in the software engineering industry, it is highly structured and time-bound. These are attractive characteristics for a framework to teach a foreign concept. While developing the workshop, the goal was to teach the framework in half a day and have the students conduct a design sprint on their project in the following days. To do this, we sought a 'toy project' that could be completed in minutes but with enough design space to design and illustrate the different steps of the framework. Finding something to illustrate the phase of understanding the design space well was especially tricky. For instance, having the students design a paper plane was considered but ultimately dismissed because the goal of a paper plane is always to fly, with limited variety. Ultimately, we decided to have the students build a paper 'tower' for which they had to discover its uses themselves. Since the goal of the 'tower' was unclear beforehand, the students had to use the framework's design tools to investigate the design space.

Furthermore, we wanted to give the students something to bring home to help or guide them in planning their design sprint. This became a set of two canvases. The first canvas had space to place sticky notes per phase of the design sprint. Each sticky note would contain a method that could be used in that phase. This canvas would be filled during the workshop, while the students learnt the phases of the framework and methods to fulfil them. The second canvas had the days of the week on it. At the end of the workshop, the students could plan their design sprint by physically moving the sticky notes from the first canvas to the second.

Workshop evaluation The workshop was evaluated in three ways. First of all, a test run of the workshop was held with the course's teaching assistants. They completed a survey about the workshop directly after the test run ended. Additionally, a survey was sent to the students who attended the workshop after their project ended. Third, the teaching assistants who supervised one or more groups that attended the workshop were interviewed after the students' projects ended.

This evaluation method was chosen because the teaching assistants have the best perspective on the workshop's impact. We considered asking the students to complete a second survey before or after the workshop in addition to the one sent out at the end of the project. Scientifically, this would have been interesting because we could have measured a delta between the students' expectations of the workshop's impact and the actual impact. However, in practice, this means that we ask the students to oversee their own learning path beforehand, which is difficult for any learner. Therefore, we did not think this would yield significant, meaningful results, so we dropped this idea and opted for the teaching assistants instead.

Workshop design second iteration The workshop was updated based on the evaluation. The changes are described in detail after the evaluation results in section 4.5. This section provides a summary of what was changed.

The main issue that had to be solved to implement the workshop in the course beyond the pilot was to find a way to limit the overhead for the coordinating teachers. Therefore, appointing the coordinating teachers as the workshop facilitators was not an option. The solution was found in appointing the teaching assistants as the facilitators.

[4]Helpful resource to get started: `https://designsprintkit.withgoogle.com/`. The book introducing the design sprint: `https://www.thesprintbook.com/`.

Several changes were made to the workshop to support the teaching assistants in giving the workshop.

- All theory presented in the workshop is moved to the online learning environment.Students are now expected to work through the self-learning module before attending the workshop.
- The self-learning module also explains the idea and set-up of the workshop so that students already know what they can expect.
- A facilitation module for the teaching assistants is added to the online learning environment. This explains the idea and learning goals of the workshop in more depth for the facilitator.

The self-learning module for the students can be found in appendix F. The facilitator module is in appendix G.

3.3.5. Teaching assistant survey

Before the workshop was given to the student groups, it was tested with some teaching assistants. Afterwards, they filled in a survey about what they think students who do the workshop would do differently. The survey questions are in appendix C and at `https://forms.office.com/e/s7VK4k354x`. This survey was intended to record beforehand what everyone expects will be the workshop's impact. This could allow us to check which expectations materialised and if other surprising changes happened.

The survey contained questions about the capabilities of the student teams that join the course and how the workshop would influence this. Furthermore, it asked the TAs to estimate how non-workshop groups would allocate their time in the project's first two weeks and how workshop groups would do this. The survey included open, closed, and Likert scale questions.

3.3.5.1. Participant selection & recruitment

All the teaching assistants who did the workshop were asked to complete the questionnaire. All teaching assistants of the course were invited to the workshop. Of the eight participants, seven completed the questionnaire, and one completed it twice. This results in a response rate of 87.5%. The survey questions are in appendix C and at `https://forms.office.com/e/s7VK4k354x`.

3.3.6. Student survey

After the software project ended, the students who completed the workshop in their first week of the project were asked to complete a questionnaire. The questions can be found in the Appendix D or at `https://forms.office.com/e/rfXfbCRX1F`. The survey asked the students how the workshop had helped them in their project. Furthermore, it contained questions about their satisfaction with the content and whether they had carried out a design sprint independently after the workshop. It contained a mix of open, closed, and Likert scale questions.

3.3.6.1. Participant selection & recruitment

Thirteen groups signed up for the workshop, ten of whom attended. However, some sent a delegation and did not attend with the entire group. Five of these ten groups responded to the questionnaire (eight students). The questionnaire was sent out on 3 Jul. 2023. The survey can be found in Appendix D or at `https://forms.office.com/e/rfXfbCRX1F`.

3.3.7. Teaching assistant interviews

After the projects ended, the teaching assistants of groups that had participated in the workshop were interviewed to determine the workshop's impact. We were interested in whether they observed differences between groups that did and those that did not do the workshop. The interview protocol can be found in appendix E.

The interviews contained questions about the satisfaction of the clients with the delivered artefact and about differences between workshop and non-workshop groups. Before the workshop started, the TA was asked to complete two Likert scale questions about the time allocation of workshop and non-workshop groups. The results of these scales were not used directly but served as a conversation starter for the interview. It primed the interviewee to consider the differences between workshop and non-workshop groups. The questions in the interview itself were open.

3.3.7.1. Participant selection & recruitment

Out of 84 student groups, twelve participated in the workshop; therefore, not all teaching assistants had

	RQ1	RQ2	RQ3	RQ4
Literature	X	X		
Startup interviews		X	X	
Company survey	X	X	X	
TA survey				X
Student survey				X
TA interviews				X

Table 3.2: Table that relates the methods employed with the research questions they helped answer.
RQ 1: *What are the dimensions of success in SE student projects at software startups?*
RQ 2: *Which preconditions make SE student projects in software startups successful (or not)?*
RQ 3: *What are the dimensions of product success in SE student projects?*
RQ 4: *To which extent can a Design Sprint workshop increase the (product) success of software engineering student projects for startups?*

a group that followed the workshop. All the teaching assistants who had a group that did the workshop were selected. Four teaching assistants responded to the interview invitation in time. Three of these four teaching assistants attended the workshop test run, so they were familiar with the content taught to their students. The interviews were held five months after the projects ended.

3.3.7.2. Interview process

The interviews were semi-structured; the interview protocol can be found in appendix E. Three interviews were conducted in person, and one was conducted using Microsoft Teams. The first interview was also used to refine the interview protocol. All interviews were audio recorded.

3.4. Data analysis

Table 3.2 summarizes the relation between the employed methods and the research questions.

3.4.1. Data analysis literature

During the literature study several different sub-themes were identified as useful to answer the research questions. These themes were:

- Success in SE projects to answer RQ1
- SE student projects at companies from a company perspective to answer RQ1 and RQ2
- Startup success to answer RQ1

The papers in these sub-themes are presented in the result chapter.

3.4.2. Client survey

The client survey was qualitatively analysed. Descriptive statistics are provided for closed questions. For the open questions, the results are summarized, and indicative quotes are provided.

3.4.2.1. Limitations

Due to the low response rate, there is an inherent risk that the responses are not representative of the entire population. The percentage of clients who plan to participate again may be higher among the respondents than among all clients because the survey was sent together with a call for new projects. Clients who were not interested in participating again could have dismissed the email before seeing the survey. Furthermore, they might be less interested in providing feedback to improve the course. This could have led to a bias toward positive experiences. After all, with a positive experience, one is more inclined to participate again.

3.4.3. Student survey

The student survey was short. It was to briefly check what areas the students thought the workshop helped them. Furthermore, it was to check if there were any negative experiences or other student objections to recommend the workshop for the following year. The survey was qualitatively analysed. Descriptive statistics are provided for closed questions. For the open questions, the results are summarized, and indicative quotes are provided.

3.4.3.1. Limitations

Due to the low response rate, there is an inherent risk that responses are not representative of the entire population. This could be a positive bias from students who were enthusiastic about the workshop and would gladly take the time to provide feedback. On the other hand, it could also be a negative bias from students who thought it was a waste of time.

3.4.4. Teaching assistant survey

The original idea was to do a mirrored survey after the projects ended. This way, we could compare what the teaching assistants thought would happen with what they actually observed. After this survey was conducted, it turned out that the number of responses was too low to perform meaningful statistical analysis. Therefore, we switched to interviews as an analysis method after the projects ended. However, the results of this survey were still used. It was qualitatively analysed, and the answers were still compared to the observations from the interviews. Descriptive statistics are provided for closed questions. For the open questions, the results are summarised, and indicative quotes are provided.

3.4.4.1. Limitations

Almost all (87.5%) of the teaching assistants who attended the workshop test run filled in the survey. Attending the test run was voluntary, though, and not all TAs attended. This creates the risk of a bias in the attending population. For instance, the attending TAs could already be more interested in the subject or otherwise have a positive association with it. This was partly mitigated by compensating the TAs with their normal TA tariff.

3.4.5. Teaching assistant interviews

The interviews with the teaching assistants were audio recorded and transcribed. The transcription was done with automatic transcription software, and corrections were needed.

On the transcriptions, a thematic analysis was performed [47]. A partly theory-driven, partly data-driven approach was used. The themes were directly derived from the research interest, and the labels within these themes were generated from the labels [47], [48]. The interviews were analysed in three steps.

First, four themes were established based on the research questions: Product (artefact) success, Influencing factors, Workshop influences, and Personal information. Next, the labels within the themes were developed. The first interview was thematically categorised. Labels were generated from the content of each theme. For instance, in the theme “influencing factors” the label “motivation” was generated. The second interview was analysed using the same procedure to develop appropriate labels further.

After this, the other interviews were coded using the labels generated from the first two interviews. Lastly, the information was combined for each theme across participants. Some labels were merged in this step for readability.

3.4.5.1. Limitations

The interviews with teaching assistants pose several risks to the validity of this research. First of all, the interviews were conducted with only half of the TAs who supervised a workshop group. The experiences and observations of these TAs may differ from the other half. Furthermore, the TAs are not professional educators. If trained teachers had been available as teaching assistants, this could have yielded different results. Thirdly, as with all interviews, there could be a social desirability bias. Interviewees might have felt a social pressure to give desirable answers. This risk is heightened by the fact that the interviewees knew the interviewer also developed and gave the workshop. The risk of this was attempted to be minimized by talking factually about the workshop and about the student groups as much as possible. Still, the interviewees might have answered differently with a different interviewer.

4
Results

This chapter presents the results from the different research methods employed. Relevant results are presented per research question. Three tools were used to answer the first three research questions.

- Literature was reviewed.
- Exploratory interviews with startups were held.
- A survey was set out among previous clients of the software project course.

Furthermore, also three tools were used to answer the last research question.

- A survey was distributed among teaching assistants who attended the workshop test run.
- A survey was set out among students who attended the workshop.
- Interviews with teaching assistants who supervised those students were held.

Twelve clients responded to the client survey. The survey's questions are in appendix B. Five were startups, three non-startup SMEs, one large enterprise, and three public institutions. 8/12 had software development as one of their core activities (Question 5 of the survey).

The project's intention for most companies was to develop a product to use or develop a prototype to investigate or develop an idea (Q11). Both categories were ticked five times in the survey. Six projects were to create data-dominant software (e.g. an app), three were for systems software, and computation-dominant software was developed in the last three projects (Q12).

In most projects (10/12), the students started from scratch (Q13). In addition, none of the projects relied on internal software that still had to be developed (Q14). However, two companies indicated that there was a dependency on existing internal software (Q15). Most projects' primary quality goal (Q16) was functionality (7/11), but some focused first on usability (3/11), and for one project, maintainability was the primary concern. No one indicated time to market as their primary quality goal. We expected that going to market faster would be one of the main reasons startups would participate in the course.

In a large majority of the projects, the technical point of contact of the students had at least a bachelor's degree in computer science[1] or higher (9/12, Q18) and at least two years of full-time working experience as a developer (10/12, Q19). Slightly more than half of the groups would work in the client's office for some days (7/12, Q21), four clients did not have available office space, and one group did not use available office space (Q20&Q21). Furthermore, most respondents thought that the overhead of supervising the students was low and motivating, as shown in figure 4.1.

[1]At least a bachelor from a university of applied sciences (HBO)

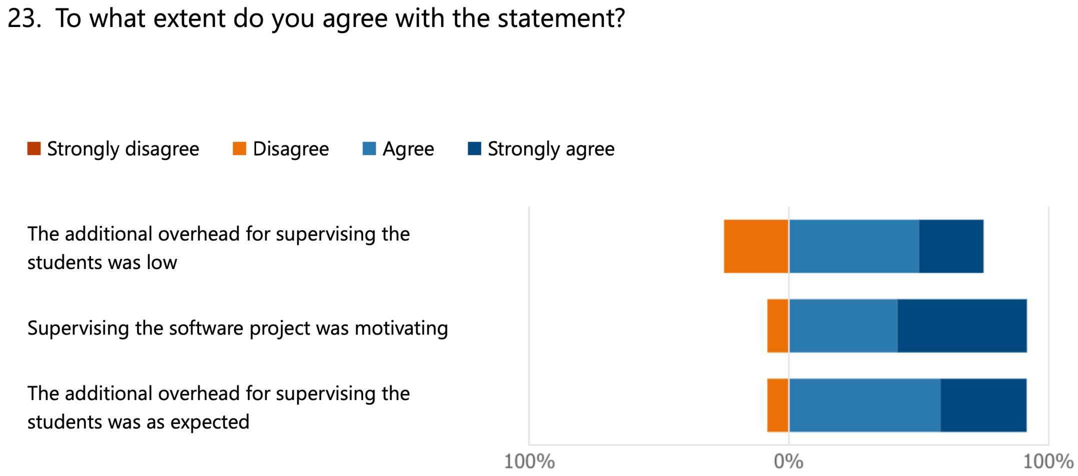


Figure 4.1: Result of Q23 of the company survey in which the respondents were asked if it was motivating to work with the students.

In a majority of the projects, the product could be used (Q27). The results were as follows.

- Three products were put into use immediately.
- Five were developed further and have or will be put into use.
- One product was a prototype.
- One was simulation software that was done after the course ended.
- Two were not developed further at the time of the survey.
- One couldn't be used at all.[2]

This resulted in the unanimous opinion that the Software Project as a whole was worth the effort, as seen in figure 4.2.

One client commented here (Q30) that the developed product is not necessarily the motivation for them to do the project. It is more about seeing their (the students') talents and staying on top of the latest technologies. Another saw "*a conflict between the need for students to have something presentable fast and producing a reliable mergeable output.*"

Developing the product further was often done with the same students. Seven clients took on one or more students as employees after the project ended (Q33). Three more offered one or multiple students a part-time job, but they declined (Q33).

[2] This brings the total to 13 because one client described two projects in this part of the survey.

Figure 4.2: Result of Q29 of the company survey that asked to which degree the project was worthwhile.

Dimension	Meaning
Stakeholders	Everyone impacted by a project, including developers, management, clients, customers and the public
Project Efficiency	The ratio of scope delivered to resources consumed and its relationship to plans, schedules, budgets, goals and contracts
Artifact Quality	Properties of the design and artifact independent of performance in specific contexts, e.g., cohesion, coupling, stability, ease of use
Market Performance	The extent to which an artifact is adopted, used, profitable and defeats competing artifacts
Time	Success-understanding varies across time, i.e., a project may appear successful before a catastrophic failure occurs.

Figure 4.3: Dimensions of Software Engineering Success from [49].

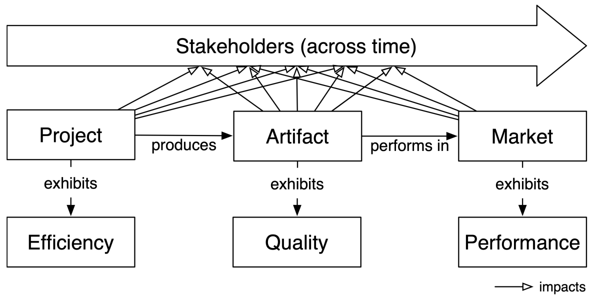


Figure 4.4: Software engineering success framework from [49].

4.1. RQ1: What are the dimensions of success in SE student projects at software startups?

Before we can answer this question, a common understanding of success in SE projects is needed. The framework presented in [49] is used for this. It is described in the next section.

In line with our research goals presented in the introduction, we investigated the dimensions of success from the startups' perspective. Students and teachers can have different priorities and, therefore, measure success in other dimensions. The related work chapter contains a section on the benefits of project education in SE programmes to cover the student and teacher perspectives. To our knowledge, two articles discuss SE student projects at companies from a company perspective, [12] and [8]. Both are based on a limited set of interviews from companies involved in one course.

Success of startups is not yet understood well [50], which limits what we can say about the project's success. This is because the two are closely connected in a startup context. Startups usually have little room for error [30]. Project failure can mean the failure of the company and vice versa.

4.1.1. Software engineering success in literature

Software engineering success is a complex, multifaceted variable [49]. Ralph and Kelly [49] interviewed 191 design professionals to investigate its dimensions. This resulted in a comprehensive theory on the dimensions of success and their relations.

The theory poses that success depends on one's perspective and when it is measured. A project can be simultaneously successful and unsuccessful. Consider employees who work overtime to deliver a project on time. From a client's perspective, the project can be a huge success when it is delivered before the deadline, but the employees can be far less delighted because of the long nights they had to make this happen. The same client can also consider the project unsuccessful a year later when all kinds of defects have come to light while using the software. Or in reverse. Consider a software project that is over time and far exceeding budget. It can be considered a failure directly after it is finally delivered. However, when, ten years later, it is still making money, the project can be considered an enormous success.

[49] synthesizes five dimensions of success: impact on stakeholders, project efficiency, product quality, market performance, and time. They are summarized in table 4.3. However, they are not all equivalent. Instead, they suggest that the following triple defines software engineering success.

$$\text{“}SES = \{Net\ Impact, Stakeholder, Time\}$$

That is, software engineering success is the cumulative effect of a software engineering initiative on a particular stakeholder at a particular time. Meanwhile, project efficiency, product quality, and market success are theorized as the primary contributors to Net Impact.” This is graphically represented in figure 4.4.

This success framework allows us to see the broader context of the projects under investigation. In our research, we cannot consider the market performance dimension. This is for two reasons:

1. student projects are inherently short, and
2. our evaluation time is relatively short after the projects are done.

On the other hand, the framework does not seem complete for our case either. The literature shows that talent scouting could also be a goal with the project [8]. Therefore, we expect to find dimensions related to the software engineering success framework and dimensions specific to the dynamic of student projects with companies.

The startup context of our research question further complicates the stakeholder impact dimension. Startups have little room for error [30]. It is vital to continuously work on something that contributes to the company's survival. However, software startup success is not understood well yet [50]. Therefore, judging if a project is a good idea is hard. The ultimate business goal of a project in a startup is the survival of the company, for which startup success has to be understood.

4.1.2. Survey results

In the survey set out among clients, the following four goals were given as the main goal for the client to participate in the software project (Q11):

- develop software to use or deploy,
- develop a prototype/demo to investigate or develop an idea,
- gain insight/knowledge, and
- recruitment.

Furthermore, the following secondary goals were mentioned in the survey and the interviews:

- keep up to date with the latest technologies (Q30), and
- fresh ideas (Q35).

Furthermore, most companies indicated that they had recruited one or more of the students after the project (Q33). This makes it plausible that recruitment was a secondary goal for a number of these companies as well.

To the best of our knowledge, one paper describes similar research. Paasivaara, Vanhanen, and Lassenius [8] also interviewed clients of their capstone project course. In their interviews, they found the following reasons for participation in the course:

- recruitment,
- develop the software,
- research and pilot new technologies, and
- fresh ideas.

We will synthesize dimensions based on these goals in the Conclusion & Discussion (chapter 5).

4.2. RQ2: Which preconditions make SE student projects in software startups successful (or not)?

The project's success is highly dependent on the student group's performance. Students will have to persevere when they face obstacles. This is the central driver of project success according to [12].For this, the students have to be motivated to continue the project.

The degree to which a student is intrinsically motivated and a group can motivate itself will vary from group to group. We have seen that highly motivated groups can deliver outstanding results without external motivation. Therefore, we think "the preconditions" for success do not exist. Still, however, most successful projects exhibit some of the same features.

Our research found three clear conditions that were the same for most projects. First, none of the projects were dependent on internal software still under development (Q14), and most of the projects were not dependent on internal software at all (Q15). Related to that, most of the projects were greenfield projects where the students could start from scratch with a new codebase (Q13). Lastly, most of the projects had an office available to the students, which they used (Q20Q21).

Interestingly, some failed projects we interviewed and in [8] reflected that they should have had the students at the office (more). This would have helped to signal the lack of motivation and commitment to the project earlier. Furthermore, we spoke with a company that stopped participating in the software project after a year when both projects failed. The company had been a course client for years. However, always with the same codebase. The company reflected that this was likely the reason for the decreased useful output over the years and the failure of the projects in the last participating year. The codebase had become too complex for students to understand in 9 weeks *and* contribute something significant. Another company that had their group work on an existing project remarked, "*There is a conflict between the need of students to have something presentable fast and producing a reliable mergeable output*".

These findings agree with the results of [8]. However, the advice for future clients in that paper is broader. We will not recite the whole paper, but a very interesting suggestion is to kick off the project with a team-building activity. This seamlessly fits into the narrative of Stahl, Sandahl, and Buffoni that motivation of the students is critical to a successful project [12]. Furthermore, it can lower the barrier to asking questions and give the students the feeling that they are part of the company's team.

Another interesting finding was that projects that indicated their main objective was recruitment did not obtain helpful information or a useful product from the project (Q11&Q30 and interviews). It appears that aiming for success in the product or process dimension is also a precondition to achieving success in those dimensions. [8] found a similar result. They say that students will recognize that a project is important for the company's future. This will work motivating.

To conclude, we found the following preconditions.

- *No internal dependencies*. The projects do not depend on internal software under development. Preferably, a project does not depend on any internal software.
- *From scratch or greenfield project*. The students can start from scratch. There is no existing code base that they have to develop further.
- *Students at the office*. The client's office has workspaces for the students, and they use them.
- *Real and challenging project*. If a project is demonstrably important to the company and sufficiently challenging, this can be an important motivator for the students. A 'toy' project or prototype that will not be deployed has to be motivating in another way.
- *Aim for success*. Aiming for a successful product and committing resources to make it successful will yield a successful product. Clients that regard successful output secondary to, e.g. recruitment, do not end up with a successful product. Whereas most clients with a successful product could develop it further with a part of the student group, i.e. they recruited successfully (given they had financial resources to hire those students).

They are not preconditions in the sense that a project will be unsuccessful if they are not met. Rather, these conditions are meant to be good indicators of a successful project. That is, if these conditions are met, chances are higher that the project will be successful.

4.3. RQ3: What are the dimensions of product success in SE student projects?

In our interviews and survey, we found five dimensions of product success. For most companies, it is crucial that the product is *developable*; that it can be developed further, as indicated by the results of Q27 of the survey. They don't expect the product to be finished by the end of the nine weeks but would like to continue working on it so that it can be used later. Some companies emphasized that the product is already *usable* in 9 weeks. This was mentioned in one of the interviews and by some companies in Q27 and Q30. It will likely also have to be developable so that updates can be done later, so the first dimension also plays a role. However, the other way around is not always the case. Companies can emphasize developability without usability. The third dimension we saw was *deployability*. This became very apparent in one of the interviews with a startup that ended up with software that could only run locally after ten weeks. This can also be modelled as a subdimension of usability, as a product that has to be deployed is only usable when it is deployable. However, the degree to which deployability matters to a project varies greatly. Therefore, it is an important (sub)dimension to consider. Furthermore, some companies use the project as an elaborate prototype (Q11&Q27). In this case, the product must be *useful as example*. Lastly, some companies aren't necessarily interested in the product itself, but rather the results it produces. For instance, when the product models some process. In this case, it is crucial that the product produces *useful insights* (Q11, Q27).

In short, we found five dimensions.

- *Developability*. The extend to which the product can be developed on further.
- *Usability*. The extent to which the product is usable when the project ends.
- *Deployability*. The extend to which the product is deployable in relevant context. For example, does it run only locally while it should ultimately run in the cloud?
- *Useful as example (prototype)*. A product does not have to be developed further but can be used as a prototype for a new project. In this case, it becomes relevant to what extent the product is helpful as example.
- *Useful insights from the product*. Not all products are products to be used by users. A product can also be a simulation of a process. In this case, the results from the product are essential. The term 'artefact' would be more correct here. There is no intention of packaging the code as a product and releasing it to users. The client is only interested in the results produced by the code.

4.4. RQ4: To which extent can a Design Sprint workshop increase the (product) success of software engineering student projects for startups?

The following three sections present the results of surveys and interviews carried out to evaluate the workshop. The first survey was conducted among the teaching assistants (TAs) who attended the workshop test run. The following section presents the results of the survey sent to students who attended the workshop. Lastly, the results of the interviews with the TAs are presented.

4.4.1. Teaching assistant survey

8 respondents, 7 of whom served as TA on the software project for the first time, for the other one it was the second time. For 3 it was also their first year of serving as a TA, 3 already did it a year, for 1 it was the third year, and the last one it was the fourth year. The TAs highlighted teamwork, internal and external communication, ambition and determination, and planning as crucial to team success (Q3). All teaching assistants believe that student teams in the Software Project have the competencies to perform well in problem-solving, but are on average neutral on whether these teams have the competences to conduct a good problem analysis, when asked on a likert scale (Q4). About a third disagrees that the students can do this, a third is neutral on the matter, and a third actually thinks that the students have the capabilities to carry out a good problem analysis. Interestingly, when asked how the student teams will approach problem analysis (Q5), the majority of the TAs expect no serious problem analysis. Some think this is because the client provides a problem statement or requirements that is seen as good enough by the students. One said: "*[they will] write the problem analysis section at the end when they realize it is a required part of the report*". Only three TAs said something about problem-solving. They were positive that the students were able to do this.

In Q6, the teaching assistants perceive the most significant challenges faced by students in problem analysis and problem-solving during the Software Project to be a lack of understanding of the importance of defining problems, followed closely by a lack of structure in their problem-solving approach and inadequate consideration of user needs and feedback. Challenges related to assignment ambiguity and limited creativity in solutions were also noted, albeit to a lesser extent. Furthermore, all TAs agree that a *structured approach* to problem analysis and problem-solving is important, see figure 4.5 (Q7).

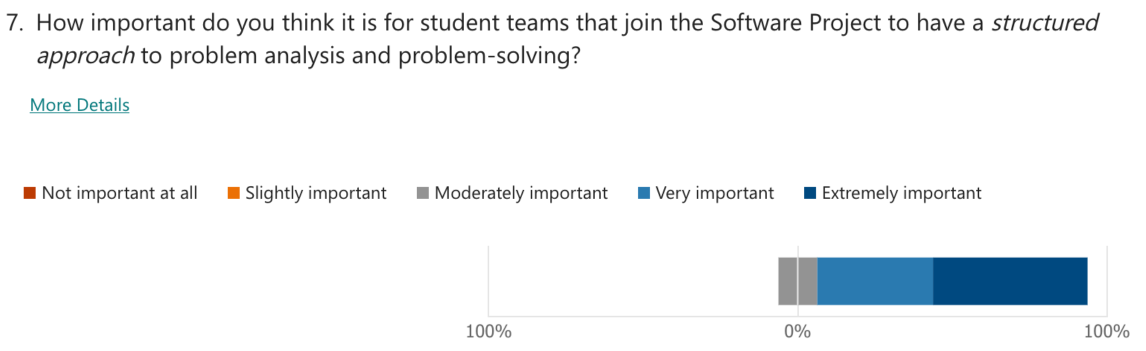


Figure 4.5: Result of question to teaching assistants after participating in the workshop test-run. All of them thought that having a structured approach to problem analysis and problem-solving is at least somewhat important for the student teams.

After discussing the challenges the students face, the TAs were asked how they expect student teams that *would* do the workshop would approach problem analysis and problem-solving (Q8). They expect that these students will likely have a more structured approach to the Software Project, with a stronger focus on problem analysis. These teams are expected to be more user-centric, take the design phase seriously, and invest time in understanding the underlying problem before jumping into solutions. As one TA put it: "*I think they will have a better structure, understanding of the problem and better idea of the next steps, which will ultimately lead to a product that actually solves a real-world problem*". The TAs hope that workshop participants will prioritize brainstorming, research, and communication with potential users, leading to more innovative ideas and an increased awareness of design decisions, ultimately resulting in products that effectively address real-world problems.

Furthermore, the TAs were asked to estimate how workshop groups would spend their first two weeks versus non-workshop groups (Q9&10). On average the TAs thought the amount of time spent on brainstorming would stay the same, but much more time would be spent on defining the problem. On the other hand less time would be spent on discussing requirements and low-fidelity prototyping.

At the end of the survey, one TA responded in the comments box of any other (Q11): "*I think it should be done during the OOPP (workshop)*".

4.4.2. Student survey

All the students that signed up for the design sprint workshop were sent an evaluation form. Five groups of the ten that attended the workshop responded. They were represented by eight different students. One group indicated that they did a design sprint, three groups said that they used some concepts from the workshop but did not do a full design sprint, and one group did not use anything from the workshop (Q6).

The feedback from the workshop attendees about their experience with the software project overall (Q3&4) highlights a diverse range of experiences. Common threads of appreciation are the importance of teamwork, early initiative, and client interaction. Positive outcomes were frequently attributed to strong team dynamics, effective communication within groups, and proactive engagement with clients and coaches, leading to the development of products that met or exceeded expectations. Many groups emphasized the value of starting early, particularly with low-fidelity prototyping, and the benefit of having a clear understanding of the problem they were solving. However, challenges were also prevalent, particularly in academic justification of project decisions, communication with clients and teaching assistants, and articulating project requirements. Several groups encountered difficulties with technology, such as unfamiliarity with Unity, which delayed project progress. Additionally, issues with internal team dynamics and task management were mentioned, indicating areas for improvement.

From Q5 we learn that overall, the workshop is perceived positively, with the strongest consensus on its usefulness and its role in improving the problem analysis. There's also a good level of agreement on its contributions to project planning, outcomes, and structure in the beginning of the project. Students by majority disagree that it had any effect on their grade or requirements development. Furthermore, Q7 shows that an overwhelming majority is very likely to use techniques from the workshop again. The same majority is at least somewhat likely to recommend the workshop to other students and to conduct a design sprint in future projects.

To the question in which areas the workshop helped (Q8), the students responded positively to:

- brainstorming,
- desk research (researching the problem context),
- defining the problem,
- interviews with the client,
- low-fidelity prototyping,
- teambuilding activities, and
- requirements.

They didn't think the workshop helped with:

- interviews with end users,
- testing prototypes,
- deciding on languages or frameworks, or
- setting up technical infrastructure.

On the transcriptions a thematic analysis was performed [47]. A partly theory-driven, partly data-driven approach was used. The themes were directly derived from the research interest, and the labels within these themes were generated from the labels [47], [48]. The interviews were analysed in three steps.

First, four themes were established based on the research questions: Product (artefact) success, Influencing factors, Workshop influences, and Personal information. Next, the labels within the themes were developed. The first interview was thematically categorised. Labels were generated from the content of each theme. For instance, in the theme "*influencing factors*" the label "*motivation*" was generated. The second interview was analysed using the same procedure to develop appropriate labels further.

4.4.3. TA interviews

As explained previously in section 3.4.5 of the methods chapter, four TAs were interviewed. Together they supervised seventeen groups. Eight of those groups did the design sprint workshop. The interviews were analysed per theme and are also presented like that. As described in the methods chapter, the themes were derived from the research interests, and labels were generated from the data. In the *product success* theme, we were looking if the workshop influenced the success of the product delivered by the students. The *workshop influences* theme, was for labels on other influences than product success. *Influencing factors* holds labels on other factors than our workshop that may have changed the outcomes of projects. In summary, the identified themes were:

- product success,
- workshop influences, and
- influencing factors.

The remainder of this section presents the results of the interviews categorized per theme.
Theme: ***Product success***

4.4.3.1. Client satisfaction

All clients were satisfied with the result they got. For some it was not more than that, they received the minimum that was agreed upon. However, a majority of the clients were extremely satisfied with the result of the project. Their student groups managed to build the minimum and extend the project with extra features or otherwise additional useful work. For one of the groups that only did the minimum, the TA explicitly said that the relative lack of performance was not related to the workshop. The other TAs did not explicitly say the relative lack of performance was or was not related to the workshop.

4.4.3.2. Product

For most of the groups, the TAs had difficulty pointing to things in the product that were influenced by the workshop, but for some, the workshop may have improved the process, they said. P4 specifically pointed out that without the workshop *“not as many prototypes and not as many iterations ... might have [been] produced”*. This included a number of surveys and play tests to improve the next prototype. “*Something that not every team did, you know, like actual surveys.*”

For one group P3 very clearly identified that they may have focused on the UI design too much, neglecting the functional side of the application. Their application was not functional at all because *“... they didn’t manage to connect with the database”*.

Theme: ***Workshop influences***

4.4.3.3. Design sprint

The idea of the workshop was that the students would carry out a design sprint on their own afterwards. We found no evidence in the interviews that any of the groups carried out a formal design sprint.

4.4.3.4. Problem iteration – general

Most of the workshop groups implemented the idea of problem iteration before the implementation process. Whereas *“other groups really just start coding right away”* [P1] and without the workshop P4 thinks *“they would have rushed more into ... programming”*. P2 identified this too, but expressed this was *“a waste of time”*. *“They spent a lot of time in the first weeks to understand what the client wanted.”* *“I think for the application in a real world it isn’t a waste of time, but when you have a time frame ... of seven weeks [coding] ... thinking about what they want you to do for two weeks is a little bit long.”*

4.4.3.5. Problem iteration – defining the problem

Most if not all workshop groups spent time defining the problem they were solving before thinking about the solution. Whereas for non-workshop groups *“finding out what the ... company really wants is something they didn’t spend a lot of time on sometimes”* according to P1. This sentiment was echoed by P4. While talking about unprepared clients, mostly clients participating for the first time, P2 said that *“students are not really prepared for that and we do not really expect them to be prepared for that”*. However, for clients that already have experience in the software project *“I would say of course they have the capabilities to explore what the client actually wanted by talking to the client, by exploring”*.

4.4.3.6. Problem iteration – user interface

The student groups made UI designs early in their projects to discuss with the client what they wanted. This was not seen in all the non-workshop groups, but said P2 *“I think all the differences are more related to the groups and the problems that they got”*. None of the TAs saw any of their groups ask potential users of the software ask for feedback early in the project, so not as part of the problem iteration.

4.4.3.7. User tests & problem iteration – user interviews

The same picture emerges here as with UI testing. Exactly one group interviewed a potential user that was not their client in the first two weeks of the project. Some groups did user tests or interviews in the second half of the project, but about half tested the software with their client only. To the dismay of some of the TAs *“What I would have rather seen as a teaching assistant from a course perspective is that they would have stepped out of that bubble at the company”*. The user tests never resulted in major overhauls, as P2 put it *“I wouldn’t say that ... it’s changed like globally now, but some feels here and there changed”*.

Theme: ***Influencing factors***

4.4.3.8. Company preparation

One of the TAs indicated that to their idea the degree to which students think about the problem before doing is highly dependent on the company’s preparation. If a client can clearly describe what they want students don’t “*bother*” digging deeper. If a client is vague about the solution, or even vague about the problem, this will force the students to think about it more than their peers in a group where the client can describe exactly what they want. This effect was seen with workshop and non-workshop groups.

"for the group ... that attended the workshop, the problem was really straightforward, so they didn't really have to think about it for two weeks, they just defined it and went from that point really quickly." P2

"Yeah, I think this group ... did actually a lot of brainstorming because of the fact that ... no lead was given ... [by] the company. they were just OK. This is the task. This is more or less what we want you to do. But then they had to brainstorm and look into how other companies do it. What exactly is this?" P3

4.4.3.9. Experience & Team dynamic

Multiple TAs pointed to the importance of previous experience in the success of the student projects. As one of them put it *"It wasn't very new for them"*. For some experienced non-workshop groups *"it was like they were at the workshop"*. In these cases it's hard to separate the effect of the workshop from the experience they brought. Next to that one TA pointed to the team dynamic as another important factor with outsized influence on the success of a project. It matters whether a team contains some strong characters or are all team players.

4.4.3.10. Non-workshop groups would have benefitted more

Another dynamic the TAs pointed out was that to them it seems that the groups that didn't attend the workshop would have benefitted more from the workshop than the groups that did. A majority of the workshop groups already had experience from side jobs and summer internships and were highly motivated to get good grades. On the other hand most of the groups with less experience didn't go to the workshop. This situation could arise because the workshop was not mandatory. Thus, the less experienced groups missed an opportunity to learn concepts their peers with working experience already had.

"If they (the non-workshop groups) would have had the workshop, I think it would have had a greater impact." P1

4.4.3.11. Motivation & experienced non-workshop groups would *not* have benefitted

Motivation of the student group seems to be a big influencing factor in the success of a project. Together with motivation, the interviewees would often mention the experience of a group. All of them had at least one motivated and experienced group that did very well (grades in the range 9 – 9.5 out of 10). Some of those did the workshop, others didn't. When asked if the workshop had any effect on the motivated, experienced workshop group, P1 responded: *"I think less than the effect would have been on the other groups"*. P2 went a step further and said the workshop has likely not made a difference with regards to the grade received. About a motivated, experienced non-workshop group P4 mentioned *"It was like they were at the workshop"*, and P2 said *"they knew basically how to do the project"*.

Most motivated non-experienced groups in the sample went to the workshop, so it's hard to say based on these interviews what would happen if a group was highly motivated, without internship experience doesn't go to the workshop. What the TAs felt very strongly about however, was that the workshop was or would have been very useful to the non-experienced groups. As P4 put it: *"I think that team would have had a great project, but with the workshop they had a perfect project process"*. About groups that didn't attend P1 said: *"If they (the non-workshop groups) would have had the workshop, I think it would have had a greater impact [than on the experienced group]."*

For the groups that did less well the TAs pointed to a lack of motivation or other team dynamics that likely caused this. None of them pointed to the workshop as a cause for the lower performance, P3 even made it explicit: *"I don't say it was because of the workshop, because they're not related".*

4.4.3.12. Receptiveness

A small but important factor all four TAs pointed to is how open a group is to external feedback. P3: *"if a group doesn't wanna do it, they will not do it"*. P1 said a group may not have benefitted as much from the workshop as they could, because *"they might have not been very open to it"*. The phenomenon was best summarized by P4: *"Some some people are more receptive to ... external feedback, and ... they take that feedback and implement it into the process. [Team X] did that because I think they were a good team before and they took the workshop and ... turned out great. But I think for ... [team Y] it ... was time lost.... It didn't feel like any improvement."*

4.5. Update educational material

Based on the results above and the conclusions in the next chapter, it was decided that we would attempt to continue teaching students design thinking for the following year. This was done in two steps.

1. Find a suitable format with limited overhead for the coordinators of the course.
2. Incorporate the learnings from the first iteration.

Step one was needed because the author of this thesis gave the workshops and is not available in subsequent editions of the course. Leaving the workshop to the course coordinators was deemed unfeasible next to their regular coordination role of the course. Furthermore, we wished to be able to teach the material to all students, instead of a select group like in the voluntary setting. Online material with explainer videos on the virtual learning environment (VLE)[3] was considered. This was not deemed a full replacement of the workshop, because of the lack of interaction. Furthermore, according to the teaching assistants interviewed, students have a negative connotation with online learning modules.

The solution was found in having the teaching assistants give the workshop in a modified form. The theory previously taught during the workshop was moved to the online module with the help of online videos and websites. The students are expected to work through this module before joining the workshop. This way the teaching assistants can fully focus on the workshop part of the workshop, and leave the theory to the students themselves. Moving the workshop to the teaching assistants as facilitators resolves the issue with getting the material to all students, the issue with just online material, and has limited overhead for the course coordinators.

This idea was also presented to the teaching assistants, who were interviewed to determine whether they considered it viable. They indicated that they thought it was a great idea because it solves the issues listed above. Furthermore, they said that they would be willing to give the workshop themselves, and thought most other TAs would be willing to do the same.

In the second step the results of this thesis were used to update the content of the educational material (the workshop and the online module together). From the interviews, it became clear that only one group talked to end-users in the early stages of their project (the first two weeks). Furthermore, only one group did user tests in the first half of their project. Therefore, more emphasis was placed on the importance of early user interviews and testing in the facilitator material. Future work on the workshop could add a simulation of a user interview to the workshop.

The self-learning module on the VLE is added as appendix F. The facilitator material was also added as a module on the VLE, hidden for the students. This module is separately added to this thesis in appendix G.

[3] Brightspace

5
Conclusion & Discussion

This thesis investigates the collaboration between startups and students in software engineering (SE) student projects. The research was performed around our bachelor's capstone project course. The projects are proposed and supervised by a client. We interviewed startups and surveyed organizations that participated as client in the course in the past two years. Furthermore, we developed and introduced a design thinking workshop.

Figure 5.1 depicts the mental model we used to frame our research. RQ1 and RQ2 investigated area 1 in the figure; the overlap between SE student projects and software startups. RQ3 researched area 2; the overlap between SE student projects and software products. In the middle, where all three areas come together, we situated our last research question, RQ4.

The research questions were:

- RQ 1: *What are the dimensions of success in SE student projects at software startups?* (Area 1)
- RQ 2: *Which preconditions make SE student projects in software startups successful (or not)?* (Area 1)
- RQ 3: *What are the dimensions of product success in SE student projects?* (Area 2)
- RQ 4: *To which extent can a Design Sprint workshop increase the (product) success of software engineering student projects for startups?* (Area 3)

The next four sections will answer every research question one by one.

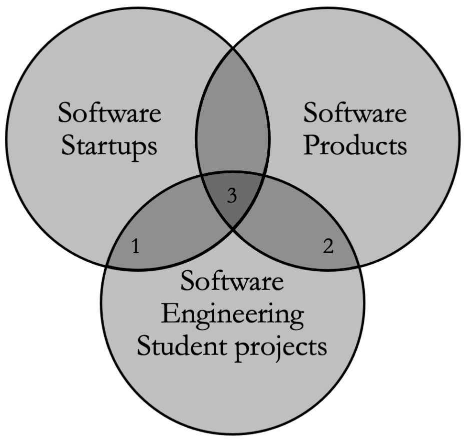


Figure 5.1: Mental model of research area with areas of interest.

5.1. RQ1: What are the dimensions of success in SE student projects at software startups?

We synthesized three dimensions of success in SE student projects at startups from our data. These dimensions are based on the goals presented in the results:

1. *Product*. If success is measured along this dimension, a client looks at what is achieved with the product that was constructed in the project.
2. *Learning*. When a client measures success along this dimension, they examine what has been learned from the project.
3. *Recruitment*. If a client measures success in this dimension, they look at whether they were successful in attracting some of the students for their company after the project.

Whether a project ultimately will be designated as a success or a failure will likely be a (non-linear) combination of these three dimensions and possibly more. We came to these dimensions based on the goals presented in the results chapter.

The main goals and secondary goals presented in that chapter are listed here in table 5.1. The table also lists the corresponding dimension. The first two goals are related to the product under construction, and the others are related to the process. However, the recruitment goal does not seem to 'fit in' with the other three process goals. The other three are all about learning something useful for the company or getting new ideas. Therefore, we concluded that these point to two dimensions: *learning* and *recruitment*.

Goal	Dimension
Develop software to use or deploy	Product
Develop a prototype/demo to investigate or develop an idea	Product
Gain insight/knowledge	Learning
Recruitment	Recruitment
Keep up to date with the latest technologies	Learning
Fresh ideas	Learning

Table 5.1: Goals found in RQ1 with their corresponding dimension.

5.2. RQ2: Which preconditions make SE student projects in software startups successful (or not)?

Our results point to the existence of at least five preconditions for success.

- *No internal dependencies*. The projects do not depend on internal software under development. Preferably, a project does not depend on any internal software.
- *From scratch or greenfield project*. The students can start from scratch. There is no existing code base that they have to develop further.
- *Students at the office*. The client's office has workspaces for the students, and they are using them.
- *Real and challenging project*. If a project is demonstrably important to the company and sufficiently challenging, this can be an important motivator for the students. A 'toy' project or prototype that will not be deployed has to be motivating in another way.
- *Aim for success*. Aiming for a successful product and committing resources to make it successful will yield a successful product. This is in contrast to clients that regard successful output as secondary to, for example, recruitment. It appears as though they do not end up with a successful product. Moreover, most clients with successful products could develop them further with a part of the student group, i.e., they recruited successfully (given they had financial resources to hire those students).

They are not preconditions in the sense that a project will fail if they are not met. Rather, these conditions are good indicators of a successful project. If these conditions are met, chances are higher that the project will be successful.

5.3. RQ3: What are the dimensions of product success in SE student projects?

We found five dimensions of product success:

- *Developability*. The extent to which the product can be developed further.
- *Usability*. The extent to which the product is usable when the project ends.
- *Deployability*. The extent to which the product is deployable in a relevant context. For example, does it run only locally while it should ultimately run in the cloud?
- *Useful as example (prototype)*. A product does not always have to be developed further but can be used as a prototype for a new project. In this case, it becomes relevant to which extent the product is helpful as an example.
- *Useful insights from the product*. Not all products are products to be used by users. A product can also be a simulation of a process. In this case, the results from the product are important. The term 'artefact' would be more correct here. There is no intention to package the code as a product and release it to users. The client is only interested in the results produced by the code.

5.4. RQ4: To which extent can a Design Sprint workshop increase the (product) success of software engineering student projects for startups?

The general picture that emerges is that the workshop positively impacted some of the groups that joined and did not negatively impact any of the groups. For student groups with no or limited side-job experience, the impact was most significant.

- *No (measured improved) product success*. We could not prove that the workshop had led to greater product success. However, we were able to show a positive impact on other aspects of the project.
- *Early process success.* The workshop improved the early process of the groups by providing structure to the first weeks of the project.
- *Problem iteration success.* The workshop helped students with their problem analysis by iterating on the problem to solve before diving into development.
- *Problem definition success.* Iterating on the problem allowed the students to properly define the problem they were going to tackle.
- *Solution iteration success.* All groups made UI designs early in the project. This means that aspects missing in the solution can be signalled early by visual communication with the client.
- *Project plan success.* The students indicated that the workshop helped to make a better project plan.

Furthermore, our results and conclusions lead to a number of recommendations for teachers who wish to add a design thinking workshop to their course.

- *Capacity for all students.* Give the workshop to all students. Our results suggest the workshop benefits students who do not join voluntarily the most.
- *Early.* The workshop improves the students' early process. Therefore, it makes sense to hold the workshop as early as possible during the course.
- *Appropriate project plan format.* A technical report structure may push students towards thinking about technical feasibility quickly. This is undesirable in a project in the idea phase. Start with a chapter on, e.g. the problem space or a description of the intended user.

5.5. Discussion

RQ1 The dimensions found in RQ1 partly fit the software engineering success (SES) framework [49] described in the results[1]. The product dimension we found fits into the artefact dimension of this framework. With the product dimension, we refer to the product or artefact that is developed in the project. This fits the description of the artefact dimension of the SES framework. Furthermore, our learning dimension can be seen as the project's impact on one of the stakeholders, so it fits under the stakeholders dimension. This seems a bit awkward, so further investigation is needed to determine if our learning dimension falls under the stakeholder dimension of the SES framework. Moreover, our recruitment dimension does not seem to fit the framework at all. From this, we can conclude that it is a dimension specific to our context. This may also imply an omission in the SES framework, although it should be noted that our context may not be the framework's intended use case.

RQ2 When we devised RQ2 we hypothesized that we would find data-dominant software and a software company as client as preconditions. We could not confirm these two preconditions. First, data-dominant software was considered the most straightforward project to develop and most suitable for the student's skills. However, half of the clients that responded to the survey did not create a data-dominant software project. Although the success of these projects is lower in our sample compared to the data-dominant software projects, the difference and sample is so small that we cannot say if this would hold for all the projects. Further research would have to determine if the students are better at developing data-dominant software. Second, whether the client is a software company does not seem to matter. We expected them to be more likely to succeed because they could support the students in their software development. This was not the case. However, developing software for a non-software company, especially if the company does not have software engineers, raises questions about software maintenance.

RQ3 The results of RQ3 show that artefact quality is not mentioned directly. This is an interesting addition to the literature because it broadens the way artefact success is understood. Previous research seems to interpret artefact dimension more narrowly as artefact quality [49].

One could argue that developability is a part of quality. In that way, artefact quality is indirectly mentioned. However, artefact quality is broader than developability, and our other dimensions do not include artefact quality. Therefore, this does not change the claim.

RQ4 The workshop did not impact all the groups that joined because some were already familiar with the concepts through internships or other work experience. This section first discusses this and other group dynamics. After that, other conclusions regarding the workshop that are less directly related to the research question are discussed.

Almost by definition, the groups that joined the workshop were motivated above average because joining the workshop was voluntary. This dynamic was also very clearly identified by the interviewees[2]. Their motivation could also be the reason for their better performance. In addition, a significant number of the groups had experience in an internship or other work experience. The workshop did not seem to make any difference for those groups[3]. On top of that, all experienced groups also seemed to be highly motivated[4]. According to the teaching assistants (TAs), the groups for whom the workshop really would have made a difference are the less motivated groups with little to no work experience. The TAs saw them making mistakes they think the workshop touched upon, and they did not observe in workshop groups[5]. This leads us to believe expanding the workshop to all groups is worthwhile.

However, this is not to say it will help all those groups. As TAs pointed out[6], groups and students should also be open to external feedback. Some students or groups are so set in their ways that they do not take advice from anyone.

Based on the results, additional conclusions can be drawn regarding the workshop, in addition to directly addressing the research question. By looking at the less experienced groups, the experience

[1] As presented in section 4.1.1
[2] as presented in the label motivation 4.4.3.11
[3] as presented in the label experience 4.4.3.9
[4] as presented in both the experience label 4.4.3.9 and the motivation label 4.4.3.11
[5] as presented in the label about non-workshop groups 4.4.3.10
[6] as presented in the label receptiveness 4.4.3.12

dynamic is left out. In that case, the result more or less matches the TAs' expectations before the evaluation. Some TAs participated in a test run of the workshop before it was given to students. Directly after this, they were asked to tell what they expected workshop groups to do differently. Most TAs expected some degree of awareness toward problem analysis. They implicitly and sometimes explicitly said that groups that would not do the workshop and the students in previous years were unaware of the importance of analysing the problem at hand. Two TAs went one step further. They not only expected a heightened awareness of problem analysis and design but also expected the workshop groups to carry this out in a structured way. Taking the time to understand the problem, ideate, and only jump into the programming afterwards.

In fact, the TAs observed that the less experienced workshop groups spent time iterating on the problem[7]. More specifically, they spent time defining the problem to solve. This was expected by the TAs[8], reported by the students[9] and reported by the TAs[10]. In addition, they all made UI designs early in their project to discuss with the client, so this is a helpful problem iteration tool. However, except for one group, this did not result in early user tests or otherwise user feedback that led to fundamental changes in the solution[11]. This seems like a missed opportunity. This is confirmed in the student evaluation. It indicates the workshop did not help with user interviews[12]. Interviewing users in the second half of the project is suitable for evaluating the user experience of the software, changing colours here and there or changing some user flow. However, it does not check whether the idea being worked on is a good idea at all. For this, talking to potential users should happen much earlier before investing time in a solution nobody wants. In summary, the students who did the workshop spent time defining the problem they had to solve; this is positive. On the other hand, they did not talk to end-users in this problem definition stage; that is an area for improvement.

In the TA survey, it also became clear that they believed that a structured approach to problem analysis and problem-solving is important[13]. One wanted to have the workshop before the first project course (in year 1). Some students echoed this same sentiment at the end of the workshop session. This points to the idea that some students and teaching assistants think basic design knowledge is fundamental for a sound software engineer.

Most students were positive about the workshop[9]. They thought the workshop was helpful. It gave structure to the first weeks of the project, helped them make a better project plan, and helped them do a better problem analysis. It is interesting and a bit puzzling that a relatively low number of students think the workshop contributed to the success of the project outcome. Why some students think it helped to make a better project plan and do a better problem analysis but did not improve the project outcome is unclear and could be something for further investigation. They also thought the workshop helped with brainstorming and low-fidelity prototyping[9]. TAs who were asked about their expectations beforehand did not expect a difference in time spent on that between the workshop and non-workshop groups[14]. Although 'helped with' does not necessarily mean 'more time spent', it is still a result worth noting.

Taken together, a picture emerges that the workshop would be helpful to some of the students. Most importantly, there was no evidence that the workshop groups performed less than their peers. The participants themselves assessed it positively, and the majority of their TAs were optimistic about their results and the impact the workshop had on them. The original intent of the workshop was for students to carry out a structured design sprint on their own. In the workshop, we go through the phases of the framework one by one and base our next steps on the outcome of the previous phase. The last half hour of the workshop was dedicated to planning the design sprint. During the workshops, we already observed that none of the groups were interested in doing that at that moment, but we hoped they would maybe do it the next day. Unfortunately, this turned out to be wishful thinking. In the evaluation, only one group indicated they did a design sprint, and none of the interviewed TAs observed anything about a formal design sprint. However, as explained above, we observed a positive impact of the workshop.

[7] as presented in the label problem iteration 4.4.3.4
[8] as presented in the TA survey results 4.4.1
[9] as presented in the student survey results 4.4.2
[10] as presented in the label defining the problem 4.4.3.5
[11] as presented in the label user tests 4.4.3.7
[12] as presented in the paragraph explaining the results of Q8 4.4.2
[13] as presented in figure 4.5
[14] as presented in the TA survey results 4.4.1

5.6. Limitations

The limitations inherent to the used research methods are listed in every method's data analysis section (3.4). This section discusses the limitations regarding the conclusions that can be drawn from the acquired results.

There are several limitations to this work. This section first discusses the limitations with regard to the client survey. Then, the impact of this on RQ1 – RQ3 is discussed, together with other limitations to these research questions. After that, the research methods used to answer RQ4: the student survey, the TA survey, and the TA interviews. This section closes with RQ4 itself.

Client survey Due to a low response rate, there is an inherent risk that the responses are not representative of the entire population. It is possible that the percentage of clients who plan to participate again is higher among the respondents than among all clients because the survey was sent together with a call for new projects. Clients who were not interested in participating again could have dismissed the email before seeing the survey. Furthermore, they might be less interested in providing feedback to improve the course. This could have led to a bias toward positive experiences. After all, with a positive experience, one is more inclined to participate again.

RQ1 – RQ3 The aforementioned possible positive bias in the client survey could mean that the results are only representative of the priorities of easily satisfied clients or of clients who are able to attract high-performing groups. Furthermore, the low sample rate of eight interviews and twelve survey respondents on a total of over eighty clients means there can also be biases in the results purely because the sample is not representative of the population.

Student survey Due to the low response rate, there is an inherent risk that responses are not representative of the entire population. This could be a positive bias from students who were enthusiastic about the workshop and would gladly take the time to provide feedback. On the other hand, it could also be a negative bias from students who thought it was a waste of time.

TA survey Almost all (87.5%) of the teaching assistants who attended the workshop test run filled in the survey. Attending the test run was voluntary, though, and not all TAs attended. This creates the risk of a bias in the attending population. For instance, the attending TAs could already be more interested in the subject or otherwise have a positive connotation with it. This was partly mitigated by compensating the TAs with their normal TA tariff.

TA interviews The interviews with teaching assistants pose several risks to the validity of this research. First of all, the interviews were conducted with only half of the TAs who supervised a workshop group. The experiences and observations of these TAs may differ from the other half. Furthermore, the TAs are not professional educators. If trained teachers had been available as teaching assistants, this could have yielded different results. Thirdly, as with all interviews, there could be a social desirability bias. Interviewees might have felt a social pressure to give desirable answers. This risk is heightened by the fact that the interviewees knew the interviewer also developed and gave the workshop. The risk of this was attempted to be minimized by talking factually about the workshop and about the student groups as much as possible. Still, the interviewees might have answered differently with a different interviewer.

RQ4 The workshop was voluntary and given for the first time; this brings several limitations. First, the fact that the workshop was voluntary likely skewed the population of workshop groups. The results and discussion above show that the workshop groups seemed more motivated than the general population. Other biases could also exist in the workshop group population due to the workshop's voluntary nature or its small size compared to the entire population. Second, since this was the first time the workshop was given, there was only one cohort to analyse. This means we cannot say anything about the generalizability of the results over cohorts. Furthermore, another risk to external validity is that the research was conducted in a single course at one university. Every capstone course is different, but our course meets most of the recommendations of the ACM/IEEE working group that described the 'ideal' capstone project[15]. An area where it is different is the length of the project. With just one quartile of an academic year, the length is one-quarter the size recommended by the working group. More time for a project intuitively leads to less pressure to find a solution fast, so more time is needed to research users' wishes and determine the real problem. I.e., more time to do design. However, research must be done to determine if this is true.

[15]as described in the section context of the method chapter (3.2)

6
Future work

The research methods employed in this thesis are qualitative, and the results agree with earlier qualitative work on the perspective of industry clients in software engineering project courses. Our course has had over eighty clients per year for several years. It is realistic that this number will stay above sixty for the foreseeable future. This creates the possibility to confirm the qualitative findings quantitatively. The first step in this research could be to repeat the survey we developed and sent to clients of the course. The survey can be included in the grading process to increase the response rate. The course staff is considering this regardless of whether the scientific work continues because they deem the survey a valuable evaluation tool.

Another thing to investigate further is why some clients are satisfied with the project, even though they indicate that it did not result in a useful product or valuable insights. It would be interesting to know what they get from the project and if they still think it worthwhile.

On the workshop side, there are several future research directions. To start, the workshop itself can be improved by adding a simulation of a user interview to stress its importance. The TA interviews clearly showed that there is still a lack of user interviews early in the projects. The updated workshop material indicates its importance to the teaching assistants through the facilitator material. Intuitively, this is not the best solution. A simulation in the workshop where the students can experience and practice a user interview as part of the design process would be preferred.

Collaboration with design (thinking) researchers may lead to other useful updates and additions to the workshop. For example, if the research is continued at Delft University of Technology, the researchers at the Connected Creativity lab of the Industrial Design Engineering (IDE) could be contacted[1]. They get several computer science students in their minor every year and are interested in the design challenges in our field.

Furthermore, the next iteration of the workshop can be monitored. It would be interesting to continue scientifically following the workshop. Letting the student assistants give the workshop is an experiment in itself. If this works, it could serve as a model for other courses.

Moreover, for the upcoming course edition, the project plan was also changed based on some of the ideas in this report. Due to logistical reasons, it will not be possible to give the workshop to all student groups yet. Therefore, this year, there will be a unique opportunity to analyse the results of student groups that did not take the design thinking workshop and compare it with the results from last year. The idea behind the project plan changes is to subconsciously nudge students towards doing design before drafting requirements. If this works, this could be much more significant than the impact of the workshop developed in this thesis.

Additionally, this thesis focused on startups in particular. The results suggest that the workshop could benefit all types of clients. However, this should be further investigated before we can draw firm conclusions.

Besides, our results do not explain why students think that the workshop helped them make a better project plan and do better problem analysis but do not think it improved their project outcome. This implies that students believe better problem analysis does not necessarily lead to an improved project outcome. On the surface, this seems contradictory, which warrants further research.

[1] https://delftdesignlabs.org/connected-creativity-lab/

References

[1] M. Spichkova, “Industry-Oriented Project-Based Learning of Software Engineering,” in *2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS)*, Nov. 2019, pp. 51–60. DOI: 10.1109/ICECCS.2019.00013.

[2] B. Bruegge, S. Krusche, and L. Alperowitz, “Software engineering project courses with industrial clients,” *ACM Transactions on Computing Education*, vol. 15, no. 4, pp. 1–31, Dec. 2015. DOI: 10.1145/2732155.

[3] M. Exter, S. Caskurlu, and T. Fernandez, “Comparing Computing Professionals’ Perceptions of Importance of Skills and Knowledge on the Job and Coverage in Undergraduate Experiences,” *ACM Transactions on Computing Education*, vol. 18, no. 4, 21:1–21:29, Nov. 2018. DOI: 10.1145/3218430. (visited on 06/02/2023).

[4] H. Burden, J.-P. Steghöfer, and O. Hagvall Svensson, “Facilitating Entrepreneurial Experiences through a Software Engineering Project Course,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, May 2019, pp. 28–37. DOI: 10.1109/ICSE-SEET.2019.00012.

[5] H. M. Haddad, “One-Semester CS Capstone: A 40-60 Teaching Approach,” in *2013 10th International Conference on Information Technology: New Generations*, Apr. 2013, pp. 97–102. DOI: 10.1109/ITNG.2013.21.

[6] J. Khakurel and J. Porras, “The Effect of Real-World Capstone Project in an Acquisition of Soft Skills among Software Engineering Students,” in *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*, ISSN: 2377-570X, Nov. 2020, pp. 1–9. DOI: 10.1109/CSEET49119.2020.9206201.

[7] H. J. C. Ellis, G. W. Hislop, S. Jackson, and L. Postner, “Team Project Experiences in Humanitarian Free and Open Source Software (HFOSS),” *ACM Transactions on Computing Education*, vol. 15, no. 4, 18:1–18:23, Dec. 2015. DOI: 10.1145/2684812. (visited on 06/02/2023).

[8] M. Paasivaara, J. Vanhanen, and C. Lassenius, “Collaborating with Industrial Customers in a Capstone Project Course: The Customers’ Perspective,” in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, May 2019, pp. 12–22. DOI: 10.1109/ICSE-SEET.2019.00010.

[9] O. Cico, L. Jaccheri, A. Nguyen-Duc, and H. Zhang, “Exploring the intersection between software industry and software engineering education - a systematic mapping of software engineering trends,” *Journal of Systems and Software*, vol. 172, p. 110 736, Jul. 13, 2020. DOI: 10.1016/j.jss.2020.110736.

[10] I. Bosnić, I. Čavrak, and M. Žagar, “Assessing the Impact of the Distributed Software Development Course on the Careers of Young Software Engineers,” *ACM Transactions on Computing Education*, vol. 19, no. 2, 8:1–8:27, Jan. 2019. DOI: 10.1145/3274529. (visited on 06/02/2023).

[11] M. Host, R. Feldt, and F. Luders, “Support for Different Roles in Software Engineering Master’s Thesis Projects,” *IEEE Transactions on Education*, vol. 53, no. 2, pp. 288–296, May 2010, ISSN: 1557-9638. DOI: 10.1109/TE.2009.2016106.

[12] D. Stahl, K. Sandahl, and L. Buffoni, “An Eco-System Approach to Project-Based Learning in Software Engineering Education,” English, *IEEE Transactions on Education*, vol. 65, no. 4, pp. 514–523, 2022, ISSN: 0018-9359. DOI: 10.1109/TE.2021.3137344.

[13] M. Unterkalmsteiner, P. Abrahamsson, X. Wang, *et al.*, “Software startups - a research agenda,” en, *e-Informatica Software Engineering Journal*, vol. 10, no. 1, pp. 89–123, 2016, ISSN: 1897-7979. DOI: 10.5277/E-INF160105. [Online]. Available: http://mural.maynoothuniversity.ie/11326/ (visited on 11/17/2022).

[14] M. Cantamessa, V. Gatteschi, G. Perboli, and M. Rosano, "Startups' Roads to Failure," en, *Sustainability*, vol. 10, no. 7, p. 2346, Jul. 2018, ISSN: 2071-1050. DOI: 10.3390/su10072346. [Online]. Available: https://www.mdpi.com/2071-1050/10/7/2346 (visited on 03/05/2024).

[15] B. H. Ximenes, I. N. Alves, and C. C. Araújo, "Software Project Management Combining Agile, Lean Startup and Design Thinking," en, in *Design, User Experience, and Usability: Design Discourse*, A. Marcus, Ed., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 356–367, ISBN: 9783319208862. DOI: 10.1007/978-3-319-20886-2_34.

[16] B. Aulet, *Disciplined entrepreneurship, 24 steps to a successful startup*. Hoboken, N.J.: J. Wiley & Sons, 2013, 11 pp., Includes bibliographical references and index. - Description based on print version record.

[17] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, "A decade of agile methodologies: Towards explaining agile software development," en, *Journal of Systems and Software*, Special Issue: Agile Development, vol. 85, no. 6, pp. 1213–1221, Jun. 2012, ISSN: 0164-1212. DOI: 10.1016/j.jss.2012.02.033. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121212000532 (visited on 03/11/2023).

[18] F. Dobrigkeit, D. de Paula, and M. Uflacker, "InnoDev: A Software Development Methodology Integrating Design Thinking, Scrum and Lean Startup," en, in ser. Understanding Innovation, C. Meinel and L. Leifer, Eds., Cham: Springer International Publishing, 2019, pp. 199–227, ISBN: 9783319970820. DOI: 10.1007/978-3-319-97082-0_11. (visited on 03/06/2023).

[19] R. Verganti, "Leveraging on systemic learning to manage the early phases of product innovation projects," en, *R&D Management*, vol. 27, no. 4, pp. 377–392, 1997, ISSN: 1467-9310. DOI: 10.1111/1467-9310.00072. (visited on 03/11/2023).

[20] D. F. O. de Paula and C. C. Araújo, "Pet Empires: Combining Design Thinking, Lean Startup and Agile to Learn from Failure and Develop a Successful Game in an Undergraduate Environment," en, in *HCI International 2016 – Posters' Extended Abstracts*, C. Stephanidis, Ed., ser. Communications in Computer and Information Science, Cham: Springer International Publishing, 2016, pp. 30–34, ISBN: 9783319405483. DOI: 10.1007/978-3-319-40548-3_5.

[21] M. Staples, M. Niazi, R. Jeffery, A. Abrahams, P. Byatt, and R. Murphy, "An exploratory study of why organizations do not adopt CMMI," en, *Journal of Systems and Software*, vol. 80, no. 6, pp. 883–895, Jun. 2007, ISSN: 0164-1212. DOI: 10.1016/j.jss.2006.09.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121206002573 (visited on 03/10/2023).

[22] G. Coleman and R. V. O'Connor, "An investigation into software development process formation in software start-ups," *Journal of Enterprise Information Management*, vol. 21, no. 6, pp. 633–648, Jan. 2008, ISSN: 1741-0398. DOI: 10.1108/17410390810911221. (visited on 01/25/2023).

[23] G. Coleman and R. O'Connor, "Investigating software process in practice: A grounded theory perspective," en, *Journal of Systems and Software*, Software Process and Product Measurement, vol. 81, no. 5, pp. 772–784, May 2008, ISSN: 0164-1212. DOI: 10.1016/j.jss.2007.07.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121207001811 (visited on 01/25/2023).

[24] C. Giardino, N. Paternoster, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: The greenfield startup model," *IEEE Transactions on Software Engineering*, vol. 42, no. 6, pp. 585–604, Jun. 2016. DOI: 10.1109/tse.2015.2509970.

[25] E. Klotins, M. Unterkalmsteiner, P. Chatzipetrou, *et al.*, "A Progression Model of Software Engineering Goals, Challenges, and Practices in Start-Ups," *IEEE Transactions on Software Engineering*, vol. 47, no. 3, pp. 498–521, Mar. 2021, ISSN: 1939-3520. DOI: 10.1109/TSE.2019.2900213.

[26] M. H. N. Nasir and S. Sahibuddin, "Critical success factors for software projects: A comparative study," *Scientific research and essays*, vol. 6, no. 10, pp. 2174–2186, 2011. [Online]. Available: https://academicjournals.org/article/article1380714134_Nasir%20and%20Sahibuddin.pdf.

[27] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," en, *Journal of Systems and Software*, Agile Product Line Engineering, vol. 81, no. 6, pp. 961–971, Jun. 2008, ISSN: 0164-1212. DOI: 10.1016/j.jss.2007.08.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121207002208 (visited on 05/31/2023).

[28] J. Reel, "Critical success factors in software projects," *IEEE Software*, vol. 16, no. 3, pp. 18–23, May 1999, ISSN: 1937-4194. DOI: 10.1109/52.765782.

[29] W. S. Junk, "The dynamic balance between cost, schedule, features, and quality in software development projects," 2000. [Online]. Available: http://lvid.org/samba/alibaba/DynamicBalance.pdf.

[30] E. Klotins, "Software start-ups through an empirical lens: Are start-ups snowflakes?" English, vol. 2305, 2018.

[31] J. Larson, S. S. Jordan, M. Lande, and S. Weiner, "Supporting Self-Directed Learning in a Project-Based Embedded Systems Design Course," *IEEE Transactions on Education*, vol. 63, no. 2, pp. 88–97, May 2020, ISSN: 1557-9638. DOI: 10.1109/TE.2020.2975358.

[32] R. Buchanan, "Wicked problems in design thinking," *Design Issues*, vol. 8, no. 2, p. 5, 1992, ISSN: 0747-9360. DOI: 10.2307/1511637.

[33] D. Dunne and R. Martin, "Design thinking and how it will change management education: An interview and discussion," *Academy of Management Learning & Education*, vol. 5, no. 4, pp. 512–523, Dec. 2006, ISSN: 1944-9585. DOI: 10.5465/amle.2006.23473212.

[34] R. Cooper, S. Junginger, and T. Lockwood, "Design thinking and design management: A research and practice perspective," *Design Management Review*, vol. 20, no. 2, pp. 46–55, Jun. 2009, ISSN: 1948-7169. DOI: 10.1111/j.1948-7169.2009.00007.x.

[35] S. C. Stewart, "Interpreting design thinking," *Design Studies*, vol. 32, no. 6, pp. 515–520, Nov. 2011, ISSN: 0142-694X. DOI: 10.1016/j.destud.2011.08.001.

[36] K. Roth, D. Globocnik, C. Rau, and A.-K. Neyer, "Living up to the expectations: The effect of design thinking on project success," en, *Creativity and Innovation Management*, vol. 29, no. 4, pp. 667–684, 2020, ISSN: 1467-8691. DOI: 10.1111/caim.12408. (visited on 04/14/2023).

[37] H. Plattner, C. Meinel, and U. Weinberg, *Design thinking* (Understanding Innovation). Springer, 2009. DOI: 10.1007/978-3-642-13757-0.pdf.

[38] M. Reimann and O. Schilke, "Product differentiation by aesthetic and creative design: A psychological and neural framework of design thinking," in *Design Thinking*. Springer Berlin Heidelberg, Nov. 2010, pp. 45–57, ISBN: 9783642137570. DOI: 10.1007/978-3-642-13757-0_3.

[39] H. Edison, N. M. Smørsgård, X. Wang, and P. Abrahamsson, "Lean Internal Startups for Software Product Innovation in Large Companies: Enablers and Inhibitors," en, *Journal of Systems and Software*, vol. 135, pp. 69–87, Jan. 2018, ISSN: 0164-1212. DOI: 10.1016/j.jss.2017.09.034. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121217302157 (visited on 03/07/2023).

[40] I. Signoretti, L. Salerno, S. Marczak, and R. Bastos, "Combining User-Centered Design and Lean Startup with Agile Software Development: A Case Study of Two Agile Teams," en, in *Agile Processes in Software Engineering and Extreme Programming*, V. Stray, R. Hoda, M. Paasivaara, and P. Kruchten, Eds., ser. Lecture Notes in Business Information Processing, Cham: Springer International Publishing, 2020, pp. 39–55, ISBN: 9783030493929. DOI: 10.1007/978-3-030-49392-9_3.

[41] D. S. Silva, A. Ghezzi, R. B. d. Aguiar, M. N. Cortimiglia, and C. S. ten Caten, "Lean Startup, Agile Methodologies and Customer Development for business model innovation: A systematic review and research agenda," *International Journal of Entrepreneurial Behavior & Research*, vol. 26, no. 4, pp. 595–628, Jan. 2020, ISSN: 1355-2554. DOI: 10.1108/IJEBR-07-2019-0425. (visited on 03/07/2023).

[42] J. Risku and P. Abrahamsson, “What Can Software Startuppers Learn from the Artistic Design Flow? Experiences, Reflections and Future Avenues,” en, in *Product-Focused Software Process Improvement*, P. Abrahamsson, L. Corral, M. Oivo, and B. Russo, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 584–599, ISBN: 9783319268446. DOI: 10.1007/978-3-319-26844-6_44.

[43] S. Tenhunen, T. Männistö, M. Luukkainen, and P. Ihantola, *A systematic literature review of capstone courses in software engineering*, Jul. 2023. DOI: 10.1016/j.infsof.2023.107191.

[44] V. Berg, J. Birkeland, A. Nguyen-Duc, I. O. Pappas, and L. Jaccheri, “Software startup engineering: A systematic mapping study,” *Journal of Systems and Software*, vol. 144, pp. 255–274, Oct. 2018. DOI: 10.1016/j.jss.2018.06.043.

[45] J.-P. Steghöfer, H. Burden, R. Hebig, *et al.*, “Involving external stakeholders in project courses,” English, *ACM Transactions on Computing Education*, vol. 18, no. 2, 2018, ISSN: 1946-6226. DOI: 10.1145/3152098.

[46] C. M. Mendonça de Sá Araújo, I. Miranda Santos, E. Dias Canedo, and A. P. Favacho de Araújo, “Design thinking versus design sprint: A comparative study,” in *Lecture Notes in Computer Science*. Springer International Publishing, 2019, pp. 291–306, ISBN: 9783030235703. DOI: 10.1007/978-3-030-23570-3_22.

[47] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, Jan. 2006, ISSN: 1478-0887. DOI: 10.1191/1478088706qp063oa. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa (visited on 02/29/2024).

[48] E. Blair, “A reflexive exploration of two qualitative data coding techniques,” *Journal of Methods and Measurement in the Social Sciences*, vol. 6, no. 1, pp. 14–29, Jan. 19, 2015. DOI: 10.2458/v6i1.18772.

[49] P. Ralph and P. Kelly, “The dimensions of software engineering success,” in *Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE ’14, ACM, May 2014. DOI: 10.1145/2568225.2568261.

[50] J. Melegati, “Towards a framework to guide the creation of development practices for software startups,” in *Lecture Notes in Business Information Processing*. Springer International Publishing, 2021, pp. 155–164, ISBN: 9783030885830. DOI: 10.1007/978-3-030-88583-0_15.

A
Startup interview questions

Hi, nice to meet you and thank you for taking the time. How are you? Busy period at X?

I will introduce myself. I completed a bachelor's degree in computer science at TU Delft and am currently completing my master's degree. That is also the reason you are sitting here with me today. For my thesis, I am investigating what opportunities there are for startups in involving software engineering students. The Software Project is of course a great example of a place where students are involved in startups, so that's why I started looking at those historical projects. This coincided nicely a while ago with the request from the professional organization of the software project to evaluate the profession from the business side. So I am now having a number of these conversations to, on the one hand, listen to how the course can be improved and, on the other hand, to hear for my thesis how the collaboration with the students in general has gone.

I also participated in the summer from a startup as a supervisor, so I have some idea of how the profession works, but every project is of course different, so I am very curious about your experience.

Before we begin, would you mind if the conversation was recorded for my research? Only I and at most my supervisor will view the recordings.

A.1. Company overview

1. Can you give us an introduction of yourself?
2. Can you give us an introduction into your company?
3. What is your role in the company?
4. Is your company a software intensive company? (Is the main product a software product?)
5. To what extent do you have software development expertise in-house?
6. What types of software applications are created as part of the product/service?

   - ○ Data-dominant software (high degree of data processing/storage, often found in websites/-mobile apps etc.)
   - ○ Systems software (operating systems, networking and communications, drivers, middleware)
   - ○ Control-dominant software (embedded, hardware control, real-time, process control software)
   - ○ Computation-dominant software (Operations research, artistic creativity, scientific software)
   - ○ I do not know
   - ○ Other

7. To what extent is the product/service tailored to individual customers?

   - ○ The start-up does not offer per-customer customization
   - ○ Occasionally, the product is tailored to suit needs of an individual customer
   - ○ More often than not, the product is tailored to suit needs of an individual customer
   - ○ Each product or service instance is tailored for an individual customer
   - ○ I do not know
   - ○ n/a
   - ○ other

8. Are there any other products developed or services provided at the same time as the primary product/service?
9. What is the primary quality goal of the product/service architecture?

   - ○ Time to market
   - ○ Functionality (accuracy, interoperability with other systems, security, functionality compliance)
   - ○ Reliability (maturity, fault tolerance, recoverability from errors, reliability compliance)
   - ○ Usability (Understandability, learnability, operability, attractiveness, usability compliance)
   - ○ Efficiency (time resource utilization, efficiency compliance)
   - ○ Maintainability (analyzability, changeability, stability, testability, maintainability compliance)
   - ○ Portability (adaptability to different environments, installability, co-existence with other systems, replaceability, portability compliance)
   - ○ I do not know
   - ○ n/a
   - ○ other

10. There was a constant...

	Strongly disagree	Disagree	Agree	Strongly agree
time pressure	○	○	○	○
shortage or resources	○	○	○	○

Table A.1: Likert scale questions about time and resources.

11. Additional relevant info

A.2. Project Overview

12. Broadly, what did the students do in their project?
13. What was your primary goal with the project?
14. To what extent was this goal related to your primary quality goal?
15. To what extent did the students achieve this goal?
16. To what extent did the students achieve secondary goals? (Possibly discovered during the project)
17. Additional relevant info

A.3. Supervision

18. How did you organise the supervision?
19. How many hours did you spend on supervision?
20. To what extent do you have software development expertise yourself?

21. Likert scale questions

	Strongly disagree	Disagree	Agree	Strongly agree
The additional overhead of supervising the students was do-able	○	○	○	○
Supervising the software project was motivating	○	○	○	○
The additional overhead resulting from the project was as expected	○	○	○	○

Table A.2: Likert scale questions about supervision.

22. Was the additional overhead more or less than expected?
23. Why did you decide to do the project with us?
24. What would you have liked to know beforehand regarding guidance and supervision?
25. Additional relevant info

A.4. Result

26. To what extent was the work of the students used internally?
27. To what extent was the work of the students used externally (e.g., released to users)?
28. If it was used, how did you maintain it?
29. Likert scale questions

	Strongly disagree	Disagree	Agree	Strongly agree
The Software Project (SP) as a whole was worth the effort	○	○	○	○
What the students made was useful	○	○	○	○
The result (the resulting artefact) in itself made the SP worth the effort	○	○	○	○

Table A.3: Likert scale questions about supervision.

A.5. Evaluation

30. What would you change in a next edition, both with regard to your own actions and the setup of the Software Project, to improve the end-result reached by students?
31. What opportunities do you see for increasing students' involvement in your type of company or organisation (i.e., startup, SME, multinational, etc.)?
32. Is there anything else you'd like to share in the context of the Software Project?

B Company survey Software Project

This evaluation survey was sent to all companies from the previous two years as part of the call for new projects for the next edition of the course.

Welcome to this evaluation form for the Software Project. Your input will be used to improve the project and for scientific research.

You have received the email, because you are our contact person. If colleagues have also been actively involved in supervising the students, we would like to ask you to include their perspective when answering the questions. If you were not the main point of contact for the students, we would like to ask you to complete the form with that colleague if possible.

The survey mainly consists of multiple-choice and short-answer questions. The questions are divided into 5 small sections. Answering all the questions should not take more than 10 minutes.

If anything is unclear while completing this questionnaire, please send an email to softwareproject-evaluatie-ST@tudelft.nl or call me on [redacted]

Thanks for filling in!

Francis Behnen

B.1. Introduction and organizational features

1. Organization name

   Will not be shared with the course organization.

   Individual answers can only be viewed for scientific research. The teachers directly involved in the course organization will only receive anonymised/aggregated results.

   We would like to get as complete a picture as possible of the Software Project, so this field is not required. If you are willing to share more with us anonymously, you can leave the field blank. In that case, it is unfortunately possible that you will receive a reminder e-mail from us later.

2. Completed by, name(s) + position(s)

   Were you not the point of contact for the students in practice? We would like to hear the perspective of both our contact person and the daily supervisor of the students. If possible, please include both perspectives when answering or fill in together.

   Your name is not required, see previous field for explanation.

3. In what year was your organization founded/started?

   For startups and other young organizations, please also choose the correct month, for organizations with more than a few years of history, '1/1/yyyy' will suffice.

4. How many employees does your organization have?
5. Is software development one of your core tasks/activities as an organization
   - ○ Yes
   - ○ No
   - ○ Other:
6. Did you sell mutiple products/services at the time of the Software Project
   - ○ Yes
   - ○ No
   - ○ Other:
7. Was your product/service still in development?
   - ○ Yes
   - ○ No
   - ○ Other:
8. What was the status of the product/service during the Software Project?
   - ○ A product prototype is developed and has not yet been released to market
   - ○ Product ws released to the market and is actively developed further with customer input
   - ○ Product is rather stable, the focus is on gaining customer base
   - ○ Product is stable, market size, share and grwth rate are established. Focus is set on launching new variations of the product
   - ○ n/a
   - ○ Other:
9. To what extent is the product/service tailored to individual customers?
   - ○ The start-up does not offer per-customer customization
   - ○ Occasionally, the product is tailored to suit needs of an individual customer
   - ○ More often than not, the product is tailored to suit needs of an individual customer
   - ○ Each product or service instance is tailored for an individual customer
   - ○ I do not know
   - ○ n/a
   - ○ Other:
10. Comments

B.2. Project

11. What was your intention with the project beforehand?
    - ○ Develop software to use or deploy
    - ○ Develop a prototype/demo to investigate or develop an idea
    - ○ Gain insight/knowledge
    - ○ Other:
12. What types of software applications are created as part of the product/service?
    - ○ Data-dominant software (high degree of data processing/storage, often found in websites/-mobile apps etc.)
    - ○ Systems software (operating systems, networking and communications, drivers, middleware)
    - ○ Control-dominant software (embedded, hardware control, real-time, process control software)
    - ○ Computation-dominant software (Operations research, artistic creativity, scientific software)
    - ○ I do not know
    - ○ Other:

13. Did the students start from scratch (with a new code base)?
    - ○ Yes
    - ○ No
    - ○ Other:
14. Were the students dependent on internal software that had yet to be developed?
    - ○ Yes
    - ○ No
    - ○ Other:
15. How did the dependencies relate to existing (internal) software and/or hardware?
16. What was your primary quality goal for the project?
    - ○ Time to market
    - ○ Functionality (accuracy, interoperability with other systems, security, functionality compliance)
    - ○ Reliability (maturity, fault tolerance, recoverability from errors, reliability compliance)
    - ○ Usability (Understandability, learnability, operability, attractiveness, usability compliance)
    - ○ Efficiency (time & resource utilization, efficiency compliance)
    - ○ Maintainability (analyzability, changeability, stability, testability, maintainability compliance)
    - ○ Portability (adaptability to different environments, installability, co-existence with other systems, replaceability, portability compliance)
    - ○ I do not know
    - ○ n/a
    - ○ Other:
17. Comments

B.3. Supervision

If you have supervised multiple groups with similar experience, please answer this section as if it were one group. If there were major differences between the groups, we would like to hear about it in the comments field.

18. To what extent has the person that was the main (technical) point of contact for the students received formal education in software development?

    If you did the project on your own you were the main point of contact for the students.
    - ○ No formal computer science education
    - ○ Dutch HBO bachelor's degree in computer science, or equivalent
    - ○ Dutch WO bachelor's degree in computer science, or equivalent
    - ○ Master's in computer science
    - ○ Other:
19. To what extent has the person that was the main point of contact for the students expertise in software development?
    - ○ No work experience as a software developer
    - ○ Only as a hobby or secondary responsibility
    - ○ Part-time work experience as a software developer
    - ○ 0-2 years of full-time work experience as a software developer
    - ○ 2-5 years of full-time work experience as a software developer
    - ○ 5+ years of full-time work experience as a software developer
20. Was an office or workplace available for the students?
    - ○ Yes
    - ○ No
    - ○ Other:

21. How much were the students present at the office?
22. How many hours in total did you spend supervising the students?

    Average over the project groups if you have supervised several groups. You can break it down by colleague if you did it together. If there were major differences between the groups, you can also break it down by group.

    The project has a duration of 10 weeks.
23. To what extent do you agree with the statement?

	Strongly disagree	Disagree	Agree	Strongly agree
The additional overhead for supervising the students was low	○	○	○	○
Supervising the software project was motivating	○	○	○	○
The additional overhead for supervising the students was as expected	○	○	○	○

Table B.1: Likert scale questions about the supervision.

24. Was the additional overhead more or less than expected?
    - ○ Less
    - ○ More
    - ○ As expected
    - ○ Other:
25. What would you have liked to know in advance about guidance and supervision?
26. Comments

B.4. Result

27. To what extent has the work of the students been used or deployed?
    - ○ What the students have developed was put in use immediately
    - ○ What the students have developed has been developed further and is or will be put into use
    - ○ The work of the students is used as an example for our own developers (prototype)
    - ○ The students' research is actively used
    - ○ Thanks to the work of the students, we know that we don't have to go in that direction (for the time being)
    - ○ The students' work has not been used
    - ○ Other:
28. If you chose option 5 or 6 in the previous question, can you explain why you cannot use the students' work?
29. To what extent do you agree with the statement?

	Strongly disagree	Disagree	Agree	Strongly agree
The Software Project (SP) as a whole was worth the effort	○	○	○	○
The new knowledge/in sights we've gained made the SP worth the effort	○	○	○	○
The final product (the resulting artifact) in itself made the SP worth the effort	○	○	○	○

Table B.2: Likert scale questions about the usefulness.

30. Comments

B.5. Evaluation

31. How did you hear about the Software Project (SP)?

    ○ One of the supervisors has participated as a student in the SP or a predecessor (Bachelor Final Project)
    ○ One of the students was already an employee and pointed out the possibility to us
    ○ Someone in our network pointed us to the possibility
    ○ Via the internship office, valorisation centre, innovation & impact center or Delft Enterprises
    ○ We were looking for opportunities to attract students and found the Project Forum

32. For how long have you already participated in the SP?

    ○ Last year was the first time
    ○ 2 years
    ○ 3 years
    ○ 4 years
    ○ 5+ years

33. Did you offer students a summer internship or part-time job afterwards?

    ○ One or more students have received an offer and accepted
    ○ One or more students have received an offer, but none accepted
    ○ We were unable to make an offer to the students due to a shortage of resources
    ○ We did not make an offer to the students because we were not satisfied
    ○ Other:

34. What would you like to change in a next edition about the design of the Software Project or in your own approach?
35. What opportunities do you see for increasing student involvement in your type of company or organization (e.g. startup, SME, multinational, etc.)?
36. Is there anything else you would like to share in relation to the Software Project?
37. May we possibly approach you again for the purpose of the research?

    Please fill in your e-mail address here

C

Software Project expectations TAs

This form was sent to all TAs that attended the workshop dry-run

You would help me out a lot by filling out this survey and providing me with your perspective.

1. This is my ... time TAing the Software Project

   ○ 1st
   ○ 2nd
   ○ 3rd
   ○ 4th
   ○ 5th

2. This is my ... year TAing

   ○ 1st
   ○ 2nd
   ○ 3rd
   ○ 4th
   ○ 5th
   ○ 6th

3. In your opinion, what are the most important factors that contribute to the success of a student team in the Software Project?
4. The student teams that join the Software Project normally...

	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
have the competences/capabilities to do a good problem analysis	○	○	○	○	○
have the competences/capabilities to do good problem solving	○	○	○	○	○

Table C.1: Likert scale questions about student competences according to the TAs.

5. In your expectation, *how* will the student teams that participate in the Software Project approach problem analysis and problem-solving during their project?
6. What do you think are key challenges that students face in problem analysis and problem-solving during the Software Project?

   □ Ambiguity in the assignment context

- ☐ Ambiguity of the assignment
- ☐ Inadequate consideration of user needs and feedback
- ☐ Lack of structure in their approach to problem solving
- ☐ Limited creativity and innovation in their solutions
- ☐ Not understanding the importance of defining (the) problem(s)
- ☐ Unclear client contact person
- ☐ Other:

7. How important do you think it is for student teams that join the Software Project to have a *structured* approach to problem analysis and problem-solving?

Not important at all	Slightly important	Moderately important	Very important	Extremely important
○	○	○	○	○

Table C.2: Likert scale question on the importance of a structured approach.

8. How do you think the Software Project student teams that ***have participated*** in the Design Sprint workshop will approach problem analysis and problem-solving?
9. How do you think teams that did ***NOT*** join the Design Sprint workshop will spend their time in the first week?

   Please copy the below list to the answer field. Let the percentages add up to 100%. Add the numbers as first two characters, you can leave the underscores.

   __%: Brainstorming
   __%: Company introduction
   __%: Decide on languages
   __%: Decide on frameworks
   __%: Desk research (researching problem context)
   __%: Defining (the) problem(s)
   __%: Design Sprint workshop
   __%: Interviews with client
   __%: Interviews with end users
   __%: Kick-off
   __%: Low-fidelity prototyping
   __%: Set up technical infrastructure (git, CI/CD, communications channel, etc.)
   __%: Teambuilding activities
   __%: Testing prototypes
   __%: Think about/discuss requirements

   (other):
   __%

   Please double check that the percentages add up to 100%
10. How do you think teams that ***DID*** *join* the Design Sprint workshop will spend their time in the first week?

    Please copy the below list to the answer field. Let the percentages add up to 100%. Add the numbers as first two characters, you can leave the underscores.

    __%: Brainstorming
    __%: Company introduction
    __%: Decide on languages
    __%: Decide on frameworks
    __%: Desk research (researching problem context)
    __%: Defining (the) problem(s)
    __%: Design Sprint workshop
    __%: Interviews with client

__%: Interviews with end users
__%: Kick-off
__%: Low-fidelity prototyping
__%: Set up technical infrastructure (git, CI/CD, communications channel, etc.)
__%: Teambuilding activities
__%: Testing prototypes
__%: Think about/discuss requirements

(other):
__%

Please double check that the percentages add up to 100%

11. Is there anything else you would like to share with regards to the Software Project and the Design Sprint workshop?

D
Student evaluation questions

This form was sent out to all students in the student groups that signed up for the workshop.

Welcome to the evaluation form for the design sprint workshop. This form is intended for students that did the workshop only. We hope the workshop was useful to you and that you had fun doing it. For further improvement of the workshop and the scientific work related to it, we would like to get your opinion. Please answer the form truthfully! The form is anonymous, so if you thought the workshop was not useful don't hesitate to indicate this.

1. Did you attend the Design Spint workshop (by Francis Behnen)

   ○ Yes
   ○ No
   ○ I don't know

2. What is your group number?
3. What went right in your Software Project?
4. What went wrong in your Software Project?
5. The Design Sprint workshop in week 1

	Strongly disagree	Disagree	Agree	Strongly agree
Was useful	○	○	○	○
Gave structure to the first two weeks	○	○	○	○
Helped to make a better project plan	○	○	○	○
contributed to the success of our project outcome	○	○	○	○
Helped to get a higher grade for the course	○	○	○	○
Helped to do a better problem analysis	○	○	○	○
Helped to create better requirements	○	○	○	○

Table D.1: General likert scale questions about the usefulness of the workshop.

6. Did you do a Design Sprint

   ○ Yes
   ○ No
   ○ Partly (used some concepts in our approach)
   ○ Other:

7. How likely are you to

	Very unlikely	Somewhat unlikely	Somewhat likely	Very likely
recommend the workshop to other students	○	○	○	○
do a design sprint in future projects	○	○	○	○
use techniques from the workshop again	○	○	○	○

Table D.2: Likert scale questions about referral and future use.

8. The Design Sprint/Problem Analysis workshop helped with...

	Strongly disagree	Disagree	Agree	Strongly agree
Brainstorming	○	○	○	○
Deciding on languages	○	○	○	○
Deciding on frameworks	○	○	○	○
Desk research (researching the problem context)	○	○	○	○
Defining (the) problem(s)	○	○	○	○
Interviews with the client	○	○	○	○
Interviews with the end users	○	○	○	○
Low-fidelity prototyping	○	○	○	○
Setting up technical infrastructure (git, CI/CD, communications channel, etc.)	○	○	○	○
Teambuilding activities	○	○	○	○
Testing prototypes	○	○	○	○
Think about/discuss requirements	○	○	○	○

Table D.3: Likert scale questions about different activities in relation to the workshop.

E

Teaching assistant interview questions

This appendix contains the interview protocol for the teaching assistant interviews. Before the interview started, the interviewees were asked to answer some Likert scale questions about the differences they observed between workshop and non-workshop groups. The reasons behind this are described in the methods chapter.

Welcome once again, as I said, this interview will be about the impact of the design sprint workshop you observed last year in your groups. This interview is conducted to understand what could be improved and should be different next time to help the organizers of the next edition to improve.

Likert scale questions completed before the interview

See next page.

1. Teams that followed the Design Sprint/Problem Analysis workshop spent more time in the first 2 weeks

	Strongly disagree	Disagree	Agree	Strongly agree	Varied per group	Don't know
On the re-quiements	○	○	○	○	○	○
Defining the problem	○	○	○	○	○	○
Setting up infrastruc-ture	○	○	○	○	○	○
Prototyping	○	○	○	○	○	○
Testing prototypes	○	○	○	○	○	○
Interviewing users	○	○	○	○	○	○
Brainstorming	○	○	○	○	○	○
Deciding languages & frameworks	○	○	○	○	○	○

Table E.1: Likert scale questions about groups that were at the workshop.

2. Teams that *did not* follow the Design Sprint/Problem Analysis workshop spent more time in the first 2 weeks

	Strongly disagree	Disagree	Agree	Strongly agree	Varied per group	Don't know
On the re-quiements	○	○	○	○	○	○
Defining the problem	○	○	○	○	○	○
Setting up infrastructure	○	○	○	○	○	○
Prototyping	○	○	○	○	○	○
Testing prototypes	○	○	○	○	○	○
Interviewing users	○	○	○	○	○	○
Brainstorming	○	○	○	○	○	○
Deciding languages & frameworks	○	○	○	○	○	○

Table E.2: Likert scale questions about groups that were *not* at the workshop.

E.1. Organizational

1. How many groups did you supervise in last year's SP?
2. How many groups attended the design sprint workshop?
3. Were you at the workshop dry-run?

E.2. Workshop evaluation

1. How satisfied were the clients of your groups that did the workshop with the end-product they received?
2. How ambitious were the group(s) that did the workshop?
3. Did the workshop group(s) do a design sprint?

    - *If yes, continue to 2.1, else 2.2*

E.2.1. Did a design sprint

1. To what extent was the company involved in the design sprint?
2. Ask about striking answers in Likert scale questions
3. What other differences in focus did you notice in the first two weeks between workshop and non-workshop groups?
4. How did this focus difference continue in the rest of the project?
5. What did teams that did do a design sprint do differently compared to groups that did not do a design sprint?
6. What effect did doing the design sprint have?
7. To what extent was the design sprint useful in their process?
8. How did this focus difference, in your opinion, translate into the final product?
    - *User centered, user validation over backend functionality*
9. If you think about the final product was the fact that they did the workshop reflected?
10. Would the final product have looked different if they didn't do the workshop?
11. What other differences did you notice between workshop and non-workshop groups?
    - *Motivation (reported in literature)*
12. To which extent did the design sprint influence the satisfaction of the client with the delivered product, in your opinion?

E.2.2. Didn't do a design sprint

1. Ask about striking answers in Likert scale questions
2. What other differences in focus did you notice in the first two weeks between workshop and non-workshop groups?
3. How did this focus difference continue in the rest of the project?
4. How did this focus difference, in your opinion, translate into the final product?
    - *User centered, user validation over backend functionality*
5. If you think about the final product was the fact that they did the workshop reflected?
6. Would the final product have looked different if they didn't do the workshop?
7. What other differences did you notice between workshop and non-workshop groups?
    - *Motivation (reported in literature)*

E.3. Workshop iteration

1. What would you improve with regards to the workshop for next year's edition?
2. What would be the conditions under which it can be a self-learning module?

    - Motivation, inspiration

3. Would your non-workshop groups have benefitted from the workshop?
4. To which extend should we make a design sprint a mandatory element of the project next year?
5. Under what conditions would you as a TA be willing to give the physical workshop? Imagine the theory would be a self-learning module on Brightspace for example, but you could schedule two hours with all your groups together to do the workshop part of the workshop.

E.4. Personal information

1. Can you tell me about your study background?

    - *Study phase (bachelor/master) last year, cohort year*

2. Can you tell me about your TA background?

    - *TA experience in SP, other TA experience*

F
Brightspace self-learning module

The following pages of this appendix are a digital print of the learning module on the online learning environment. The original content can be checked with a Delft Univesity of Technology Brightspace account at `https://brightspace.tudelft.nl/d2l/le/content/595308/Home`. The closed sub-module ”TA Overview” is added as a separate appendix after this appendix.

Tackling the problem analysis with Design Thinking

In this module we will introduce a problem-solving process called **design thinking**. This process can help you develop innovative solutions to complex challenges (such as the problem your client has). By understanding users' needs, generating creative ideas, and refining solutions through testing, you can create products and services that truly meet users' needs.

What Is Design Thinking? An Overview

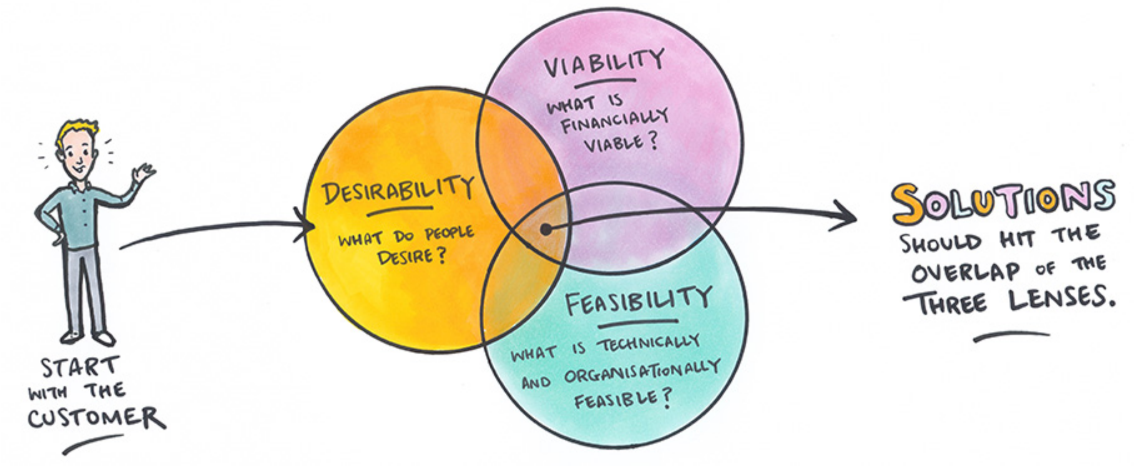
Video

Part 1: Where to start a project

As software engineers the first question we often ask ourselves in a new project is: "What can we do?". Maybe the project involves a shiny new technology; what can we do (with it)? Or maybe you just learned how to use a new tool; what can I do (with it)?

It is fun to dream about the possibilities this shiny new technology or your newly acquired skill open up. However, it may not be the most helpful starting point. In the end you want to create software that people will actually use. Therefore we would pose that a more helpful question to kick off your project would be: "What should we do?". With the quick follow-up: "What does the user want?".

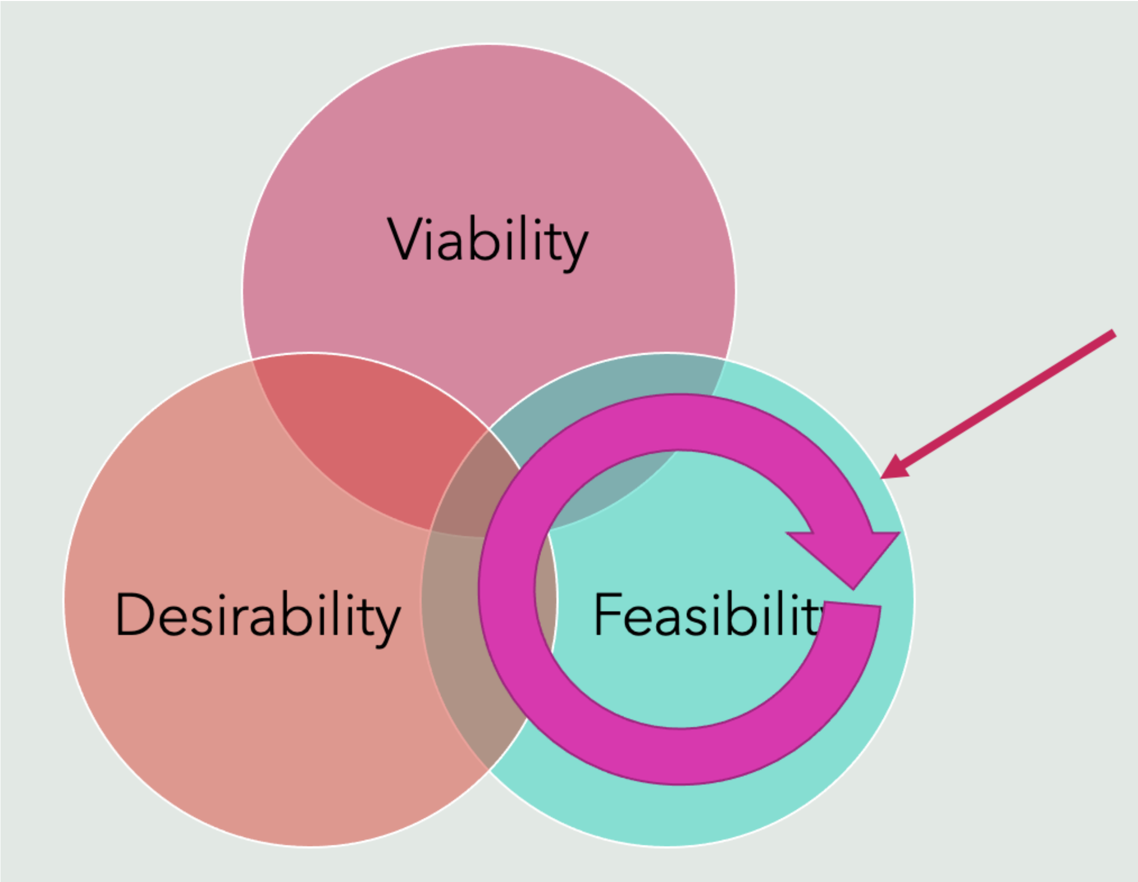
What can be done is still a relevant question, but it should be explored *after* knowing what the user wants. A project is only successful if it is something users want, within technical capabilities, and financially possible. The following graph sums up this idea:



Watch this video for a longer explanation about the three lenses:

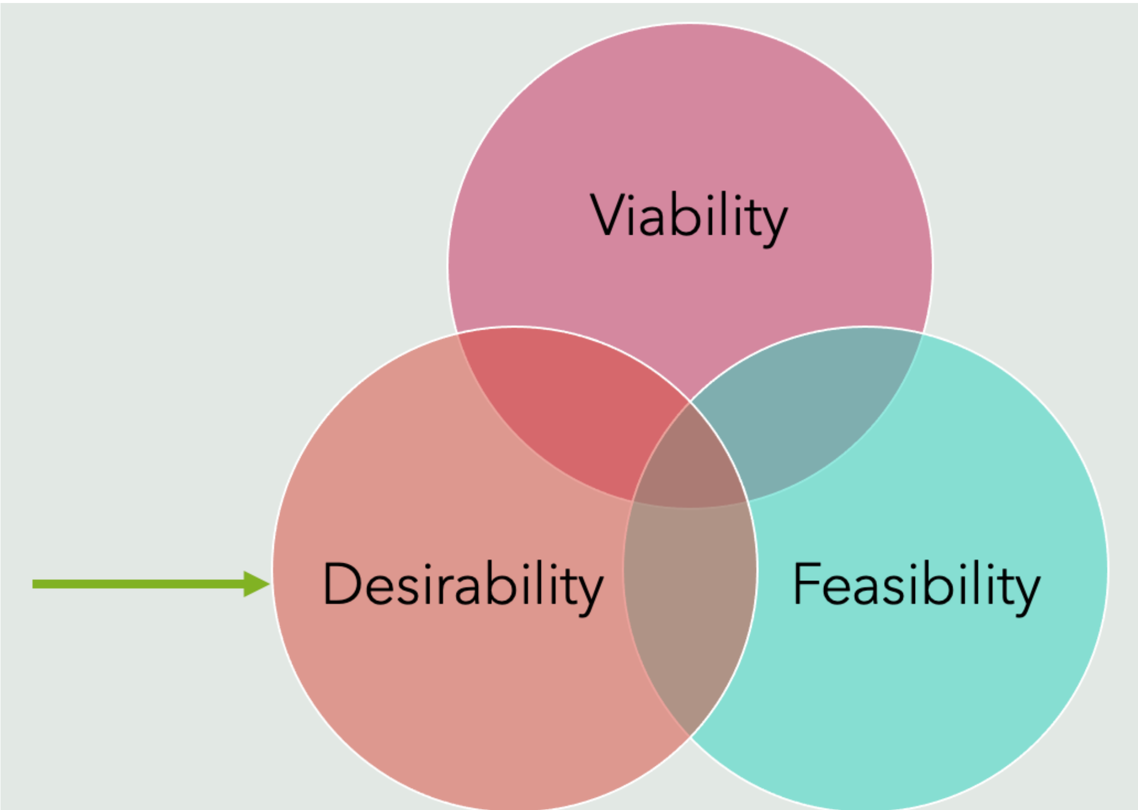
Pay specific attention to what he says at 2:00. **“There’s no way you can build a business on something people don’t want.”**

As said in paragraph one, as software engineers we tend to start to dream about what is feasible. We are also good at figuring out what is feasible. This is because Scrum is a method to figure out what is feasible and usually this is the only method employed in a project. In terms of the three lenses, SCRUM is a method to iterate on the feasibility.



Scrum is what we know, so that is were we start.

In this module and the workshop by your TA you will learn a structured method to start with the desirability lens and iterate on desirability.



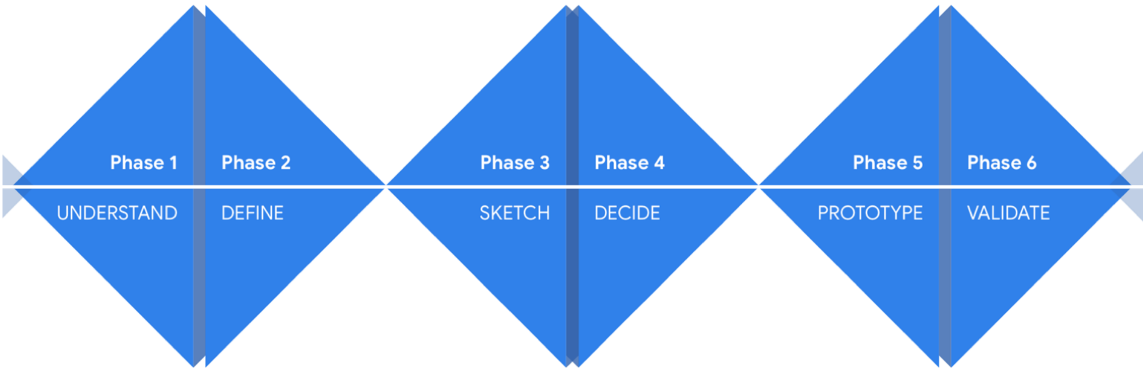
Drag and drop files here to create and update topics

Part 2a: Design sprint

Engineers tend to think they need a demo product with the basic functionalities before they can test it with potential customers. With software, just like in other engineering disciplines, this is a costly business. Months of engineering elapse before you have something basic. The design sprint method turns this idea on its head. According to the design sprint method, you can always know if you're on the right track *within a week*. User feedback first, engineering second; desirability first, feasibility second.

This is a great property in the software project, too. One week to find out what your users want leaves ample time to create your software. In addition, it provides structure to the exploratory phase of your project.

Design sprint intro



A design sprint consists of 6 distinct phases, each serving its own sub-goal and together answering the question of whether you're on the right track.

Watch this video to learn how the design sprint is used in a well-established design company.

What we want you to take away from this video is the general idea of doing a design sprint and why companies are doing them. You will notice that they talk about general activities per day of the week, instead of the six phases in the picture above. Their specific process is not important, but for those interested, this is how their process maps to the phases:

- Monday morning: Phases 1 & 2
- Monday afternoon: Phase 3
- Tuesday morning: Phase 4
- Tuesday afternoon: Start of phase 5
- Wednesday: Phase 5
- Thursday: Phase 6

If you are interested in more context about the design sprint, this playlist has more relevant videos.

The Design Sprint is a time-constrained, structured problem-solving process popularized by Google Ventures (GV). It was initially developed by Jake Knapp and has been widely adopted across various industries for its effectiveness in solving complex problems, validating ideas, and rapidly developing new products or services.

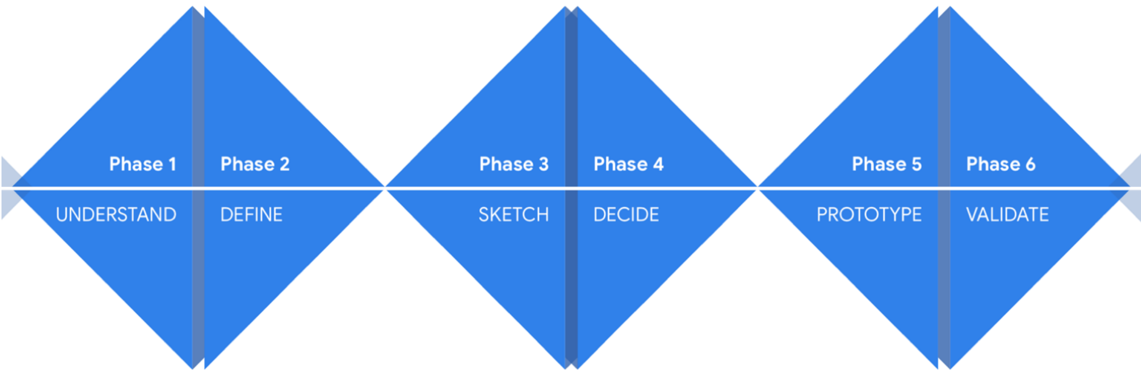
The Design Sprint method was originally conducted over five days (but it can be adapted to different timeframes) and involves six key phases:

1. **Understand**: In this phase, the team gathers information about the problem at hand, the target audience, and the market context. They discuss assumptions, goals, and constraints to ensure everyone has a shared understanding of the challenge.
2. **Define**: The team collectively identifies the most critical problem to be solved during the sprint, typically by focusing on user needs and pain points. This phase is crucial for establishing a clear and actionable problem statement.
3. **Sketch**: Each team member individually generates ideas and solutions by

sketching their thoughts on paper. This phase encourages creativity, diverse perspectives, and equal participation from all members.

4. **Decide**: The team comes together to share, critique, and discuss their sketches. They then decide on the most promising solution(s) to move forward with, often by using a structured voting process.
5. **Prototype**: The team rapidly creates a tangible prototype of the chosen solution(s), which can take various forms depending on the project (e.g., a wireframe, a physical model, or a storyboard). The goal is to create something that can be tested with users to gather feedback.
6. **Validate**: In the final phase, the team tests the prototype with real users, stakeholders, or experts to gather feedback and insights. This phase helps the team understand if their proposed solution is desirable, viable, and feasible before investing more time and resources into development.

The Design Sprint method emphasizes rapid iteration, collaboration, and user-centricity. It helps teams make informed decisions, reduce risk, and accelerate product development. It's particularly useful for organizations looking to innovate, validate new ideas, or tackle complex problems with limited time and resources.



Drag and drop files here to create and update topics

Part 2b: Practice

At the start of the project, you can sign up for a design sprint workshop to practice this method. Please read this sub-module before attending the workshop.

Every phase of the design sprint (understand, define, sketch, etc.) has specific exercises. A very helpful exercise in phase 1 (understand) is the "How Might We (HMW)" exercise. You can apply this directly in your first conversation with your client!

https://designsprintkit.withgoogle.com/methodology/phase1-understand/hmw-sharing-and-affinity-mapping

During the workshop, you will also practice with this. The workshop will further (loosely) be based on the following schedule. Keep in mind that it's a creative session, so don't be surprised if your TA deviates from it!

Understand & Define (Phases 1 & 2)

Goal: have a one-sentence goal for the prototype that will be tested at the end of the sprint. The answer to the sentence should make it clear if the current path makes sense, or if it's better to try something else.

- How might we's (HMW) https://designsprintkit.withgoogle.com/methodology/phase1-understand/hmw-sharing-and-affinity-mapping
    - Write down all the opportunities you see (on rectangular sticky notes)
- (if needed, group opportunities "that belong together" https://designsprintkit.withgoogle.com/methodology/phase1-understand/method-1)
- Dot vote https://designsprintkit.withgoogle.com/methodology/phase1-understand/hmw-voting
    - Hang all the HMWs on the wall
    - Every team member gets two or three dots to vote on the challenges they'd like to solve
- Choose a sensible number of challenges (likely 2 or 3 in the workshop setting) to try to solve with the prototype (given the constraints, time limit, etc.)
- Take the challenges chosen in the dot vote and formulate a one-sentence goal for the prototype that encapsulates all those challenges

Here you can find other exercises to come to your goal for the sprint: https://designsprintkit.withgoogle.com/methodology/phase2-define

Now, you should have a clear goal for the sprint. You have ***defined*** the challenge. In other words, phases 1 & 2 are done.

Sketch & Decide (Phases 3 & 4)

Goal: Have a clear idea of the prototype that will be built.

- Brainwriting, or 'braindrawing' in this case. Paragraphs 2 to 4 of this article:
    - Every group member takes one piece of paper and a pen(cil)
    - Divide a piece of paper in 4 or 5 sections.
    - Set a timer for one minute.
    - Draw an idea for a solution in one of the sections.
    - When the timer is up, everyone passes their paper over to the person next to them.
    - Repeat as many times as you have group members, if you are with less than 4 make sure you fill all 4 sections.
- Draw a final idea for the prototype each https://designsprintkit.withgoogle.com/methodology/phase3-sketch/solution-sketch
    - Take some time to look at all the ideas after the 5 minutes are over
    - Take a fresh piece of paper and draw your final idea for the prototype
- Present final drawings to each other https://designsprintkit.withgoogle.com/methodology/phase4-decide/present-solution-sketches
- (dot vote on final drawings if needed https://designsprintkit.withgoogle.com/methodology/phase4-decide/dot-vote)
- Synthesize one final design for the prototype from the final drawings. This can be one of the final drawings if everyone votes on that one drawing, or a synthesis of multiple ideas

Now you know what your prototype is going to look like. You have ***decided*** which prototype to build. In other words, phases 3 & 4 are done.

Prototype & Validate (Phases 5 & 6)

Goal: know what to do next.

- Take 10 minutes to build the prototype (tips for later https://designsprintkit.withgoogle.com/methodology/phase5-prototype/assign-tasks)
- Test the prototype
- Reflection on result
    - New full Design Sprint iteration?
    - Or circle back to one of the other phases? E.g., go back to phase 4 and take a different idea to prototype
    - Or was the prototype an epic win and can you continue down on this path?
    - What goals were met?

    - How to proceed from here? https://designsprintkit.withgoogle.com/methodology/phase6-validate/sprint-conclusion-recap-and-next-steps

Now you have an answer to the challenge you defined in phase 2. Was this the right challenge to pursue? Was the prototype the right answer to the challenge? The answers to these questions may help you determine your next step.

When you run your own design sprint phase 6 should tell you how to proceed. After validating the prototype, it should be clear whether the idea is worth pursuing. If the prototype did not fulfil the user's wishes, another prototype idea could be tried next by going back to phase 4 (decide). Or maybe team members have new ideas, so another brainstorming session (sketch, phase 3) is possible. Another outcome could be that the problem statement established at the end of phase 2 does not seem correct anymore, because the design sprint has uncovered new insights previously unknown to the team. In that case, the team will have to come to a consensus again on the problem statement and start in phase 1.

Google identifies the following outcomes:

- **An efficient failure:** The prototypes didn't hit the mark, but you learned something (or many things) and saved your team four to six months of work building the wrong product. You might want to run a follow-up Sprint.
- **A flawed success:** Some ideas met your user's needs, but not all of them. You learned something and can now iterate and test again.
- **An epic win:** The concept met your user's needs; they were able to complete tasks easily and engaged with all the features you mapped out. You are ready to implement!
- **Closing the Sprint.** The team has worked really hard, generated a broad range of ideas, and learned from their users. The Sprint Master should acknowledge all of this hard work and celebrate the learnings the team has achieved. A closing circle is a nice way to bring everyone together. Here you can ask people to share insights, what they will take away from the experience, and give people a sense of accomplishment. And perhaps discuss what problem you want to tackle next!

(https://designsprintkit.withgoogle.com/methodology/phase6-validate/sprint-conclusion-recap-and-next-steps)

I bombarded you with a load of reference material in this section. If you didn't read all of them, at least read paragraphs 2 to 4 of this article https://trainingmag.com/brainwalking-in-search-of-better-brainstorms/.

Your own design sprint

Some of the exercises from the workshop like the HMW, the brainwriting and the dot vote will be helpful for your own project. Others may be less easily applicable for your specific project. The design sprint kit website provides a plethora of different exercises. Use it as inspiration on what you could do if you feel that the exercise from the workshop is not helpful. You could also browse through the YouTube channel and playlist linked earlier. Don't be afraid to come up with your own exercise. Every project is unique and needs its own set of exercises to solve. If you can get buy-in from your fellow group members that you should do a certain exercise, be it your own, or one you found online, it's likely a good idea.

Drag and drop files here to create and update topics

Feedback

TA overview

G

Brightspace facilitator module

The following pages of this appendix are a digital print of the workshop facilitator module on the online learning environment. This module is meant as a help for the teaching assistants that will give the workshop in the new set-up. The original content can be checked with a Delft Univesity of Technology Brightspace account at `https://brightspace.tudelft.nl/d2l/le/content/595308/Home`.

TA overview

Any condition must be met

Enrolled in current org unit as the role: **TA with grade access**

Enrolled in current org unit as the role: **Teaching Assistant**

Enrolled in current org unit as the role: **Instructor**

Enrolled in current org unit as the role: **Course Manager**

Workshop introduction

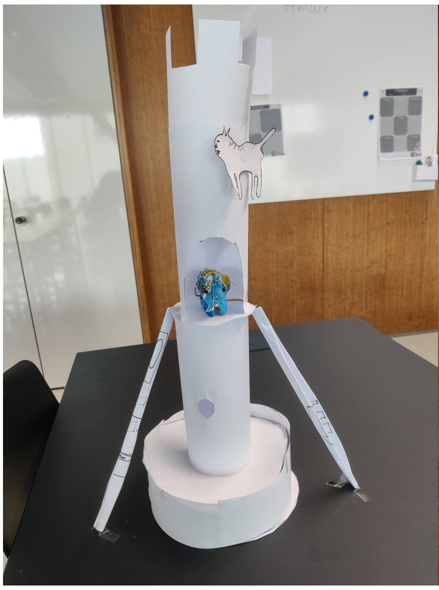
The design sprint method provides students with tools to structure the first weeks of their project. It allows them to properly explore the problem space and the solution space before drafting requirements for a specific solution.

The workshop allows students to experience the method and practice with some of the tools and techniques. In the workshop, the students will use the design sprint method to design a paper "tower" that fulfils certain goals or design constraints and then build a prototype. In the first two phases of the workshop, they come up with the tower's functionality and design constraints. By doing this, they practice exploring the problem space. In phases 3 and 4, they can practice creative brainstorming techniques for coming up with innovative solutions.

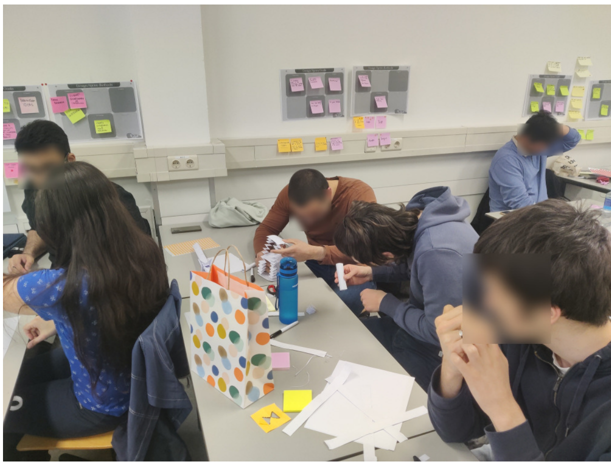
A number of things are good to keep in mind as workshop facilitator:

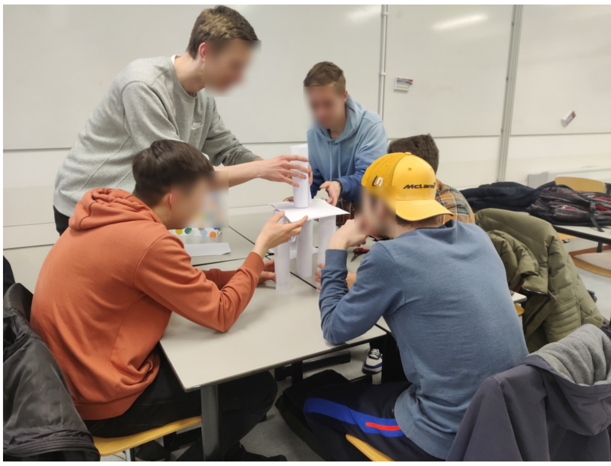
- The goal of the workshop is to experience a design sprint
- What the students design and build does not matter
- CS students tend to search for the "optimal" solution, which is why I gave time indications for some of the exercises. Completing the full process is more important than finding the best solution

This workshop was given for the first time in SP 22/23 Q4.




HMWs
Design Sprint Methods
Planner






If you *have not* attended the workshop yourself, make sure you have worked through the student material before continuing this facilitation document.

If you *have* attended the workshop before, the student material can still be helpful as a refresher. The following things changed with respect to edition 22/23 Q4:

- No theory in the workshop itself anymore, the theory that was in the workshop is moved to the online learning module
- Instead of building a tower twice, at the start and at the end of the workshop, the students build a tower only at the end of the workshop.

Workshop description

Game Name: The Paper Tower Challenge

Game objective: Build a useful paper tower

Learning objective: Encourage human-centred thinking by introducing the Design Sprint

Duration: 1.5 - 2 hours

Materials:

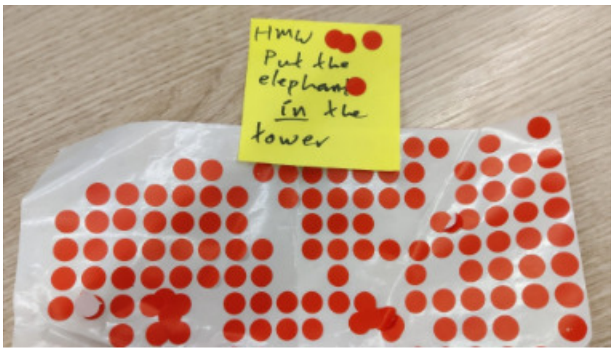
- 10 Sheets of paper per team
- 1 or 2 Scissors per team

- 1 roll of tape per team

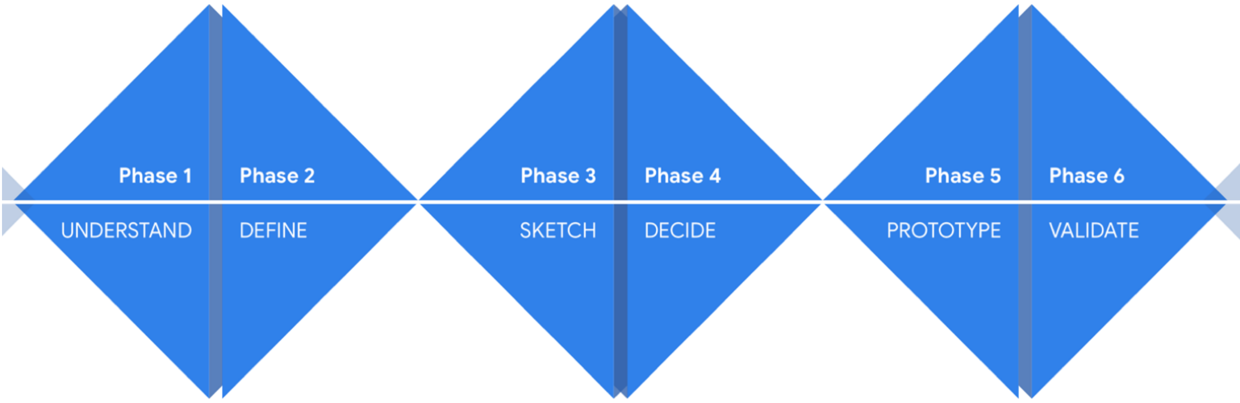
- Square post-its

- Rectangular post-its (for the HMW exercise)
- 2 or more sharpies per team

- Voting dots



The workshop's idea is to have the students experience design thinking with a toy engineering example. We made it “the paper tower challenge”, because paper is easily available, while simultaneously providing different possibilities for manipulation (e.g. cutting, rolling, folding).



The workshop follows the 6 phases of the design sprint. For me the tricky thing with some less enthusiastic groups was to get them started coming up with ideas for a paper tower. The best thing would be if they come up with an idea completely by themselves, but not all groups are able to do this. To remove the competition element we wanted every group to have their own goal. The workshop is not about winning, it is about the method. It is tricky to give the students enough so they can start with the HMW exercise, but not too much, whereby, in essence, you have come up with all the ideas. You are the client for the "project", so you can definitely give pointers, but they should also explore the assignment themselves to learn that creative divergent thinking. For me it would broadly work in the following way:

1. Ask all groups "what can a paper tower do?"
   - With a good group you can some response here already, if not you'll have to get it out of them with additional questions
   - "where can a paper tower be?" -> "what can it do there?" this brings you back to "what can a paper tower do?"
   - "what rooms do you have in a house?", this was a trick question to get them engaged. The answers to the question are the same as the previous question, but no one can pretend they don't know what types of rooms a house has. So if they mention a room in a house you can still ask the follow up question "what can a paper tower do in that room?".
2. From here we tried to get them to come up with additional ideas. Although all ideas are diverging now, we would still do this classically. My hope was that by doing it like that groups would also help each other with ideas, but in practice this never happened. Still probably it's better to do this classically to keep everyone engaged, until they have enough ideas to get started. Two examples:
   - One group said a paper tower could "be pretty" and "in the living room", so I asked them "what else could it do in the living room, while being pretty?"
   - Another group had the idea of a tower holding photos. So I asked them what else it could do while holding photos and they came up with a rotating photo holder (to my amazement it worked too at the end of the workshop :o)

3. For really hard groups that refused to come up with something themselves I would have some items with me that I could divide over the groups. E.g.: keys, keychain, pens, small toy.
4. From here you can transition to the HMW exercise.
    - “How might we create a paper tower?”
    - “How might we make a paper tower that holds a pen?”
    - “How might we create a pretty tower?”
    - “How might we create a paper tower for in the toilet?”
    - “How might we create a rotating tower?”
    - All ideas and opportunities should be listed here to generate enough HMWs to vote on in phase 2

PS: one of the workshops a student asked the existential question “what is a tower?”. This is not important, the point is to build something out of paper. It did however provide an opportunity for additional constraints. So I returned the question “yea, what is a tower?”. Any definition is fine, it’s just an additional design constraint.

You don’t have to follow this protocol. The point is that the students have a starting point to start the understand phase.

I haven’t tried this, but another fun idea I thought of later could be to role-play the chaotic client talking non-stop without clear aim, like in this video:

For instance, you can tell a story that you always loose everything around the office:

- House keys
- Car keys
- There’s never a pen to be found when you need one
- You want to have pictures on your desk, but not always the same ones
- Your mouse and keyboard cables are always in the way (point to them in the classroom)
- Think of other things to fix in an office and bring some attributes yourself

And then go on a tangent about the nasty coffee in the office or the toilet paper always being run out, something to make it funny (see video).

Phases

Rewatch and reread if needed:

- https://youtu.be/sRGk5oKXgCk?si=WIc1QEuY7gSkt-2o
- Paragraph 2 to 4 https://trainingmag.com/brainwalking-in-search-of-better-brainstorms/

It’s more important to complete the full process than to do everything perfectly. Our students tend to want to do everything “right”. In design there is no right or wrong, only different options. To force the students to come up with ideas, rather than consider every option to the last detail I timed most exercises. The timings I put in this schedule are a suggestion, if you see something meaningful going on after the timer has ended you don’t have to force the students to stop. But keep in mind, completing the process is more important than doing it perfectly. So don't boss them around with a stopwatch in hand, but do try to save them from going an hour over time. In the next sections I copied the schedule from brightspace with info for the facilitator in **bold** and the hyperlinks removed for readability.

Understand & Define

Goal: have a one sentence goal for the prototype that will be tested at the end of the sprint. The answer to the sentence should make it clear if the current path makes sense, or if it’s better to try something else.

- How might we’s (HMW)
    - Write down all the opportunities you see
    - **At least 10 HMWs so that there can be discussion and trade-offs about which to pursue**
- (if needed, group opportunities “that belong together”)
- Dot vote (**3m**)
    - Hang all the HMWs on the wall
    - Every team member gets two or three dots to vote on the challenges they’d like to solve
- Choose a sensible number of challenges (2, 3 or likely) to try to solve with the prototype (given the constraints, time limit etc)
- Take the challenges chosen in the dot vote and formulate a one sentence goal for the prototype that encapsulates all those challenges (**3m**)

Now you should have a clear goal for the sprint. You have ***defined*** the challenge.

Sketch & Decide

Goal: have a clear idea for the prototype that will be built.

- Brainwriting, or 'braindrawing' in this case.
    - **To prevent the students from discussing their drawings extensively while switching the papers I had a timer on 5 minutes and would call out every minute. This forces them to switch quickly and stay in the "new ideas" mode, instead of the "what do you mean with this line?" mode. You can also tell them to do the exercise in silence.**
    - Every group member takes one piece of paper and a pen(cil)
    - Divide a piece of paper in 4 or 5 sections.
    - Set a timer for one minute.
    - Draw an idea for a solution in one of the sections.
    - When the timer is up, everyone passes their paper over to the person next to them.
    - Repeat as many times as you have group members, if you are with less than 4 make sure you fill all 4 sections.
- Draw a final idea for the prototype each (**2m)**
    - Take some time to look at all the ideas after the 5 minutes are over
    - Take a fresh piece of paper and draw your final idea for the prototype
- Present final drawings to each other (**30s per group member**)
- (dot vote on final drawings if needed (**1m**)
- Synthesize one final design for the prototype from the final drawings. This can be one of the final drawings if everyone votes on that one drawing, or a synthesis of multiple ideas (**3m**)

Now you know what your prototype is going to look like. You have ***decided*** which prototype to built.

Prototype & Validate

Goal: know what to do next.

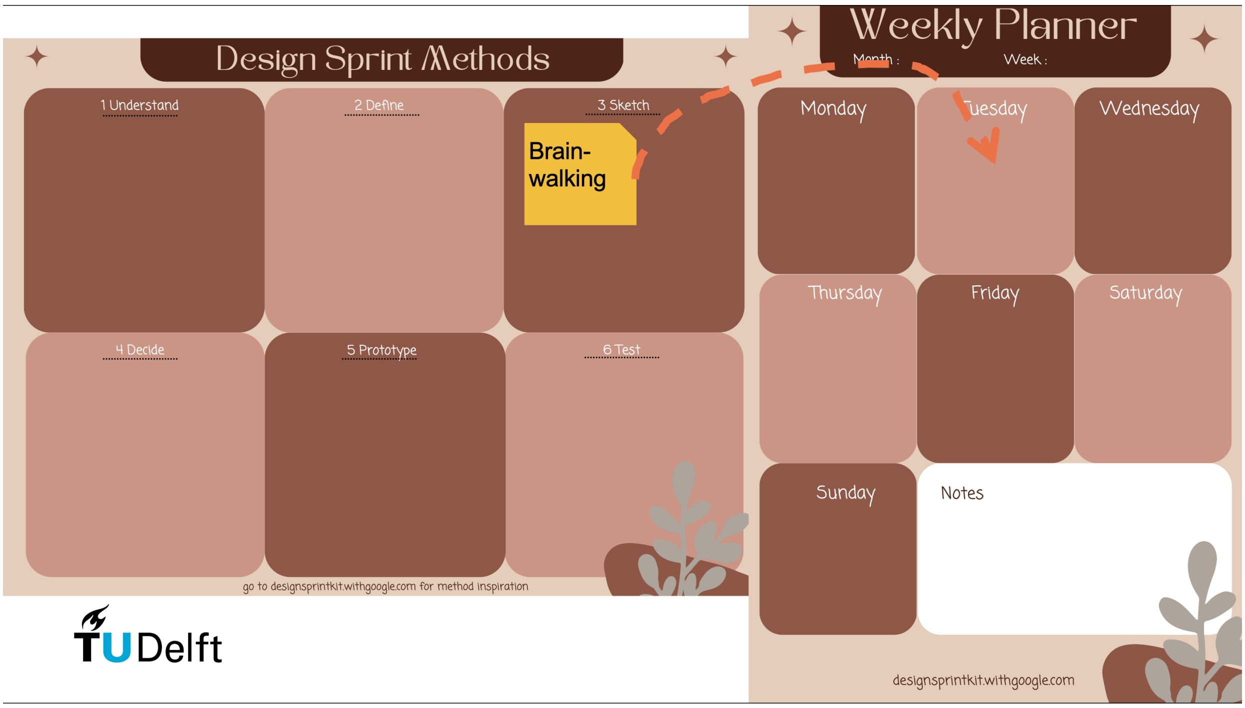
- Take 10 minutes to build the prototype (**10m**)
- Test the prototype (**For most towers this shouldn't take more than a minute, but usually the participants are still high in their energy level. Give them a minute to blow off steam and maybe look at the results of the other groups**)
- Reflection on result
    - New full Design Sprint iteration?
    - or circle back to one of the other phases? E.g. go back to phase 4 and take a different idea to prototype
    - or was the prototype an epic win and can you continue down on this path?

    - What goals were met?
    - How to proceed from here?
- **Reflection on process**
    - **What did you think of the process?**
    - **What are you going to apply in the software project?**
    - **What are you going to apply in professional projects?**
    - **Are you going to do a full design sprint in the software project?**

Now you have an answer to the challenge you defined in phase 2. Was this the right challenge to pursue? Was the prototype the right answer to the challenge? The answers to these questions may help you determine your next step.

Planning

If groups are motivated to do a design sprint you could close the workshop by planning the design sprint. There are posters to make a planning in the supplementary material. There's a poster per phase and a poster per day of the week. I would have the students collect all methods they thought were interesting on sticky notes and stick them to the appropriate phase on the methods poster. Method inspiration can be found on the designsprintkit website https://designsprintkit.withgoogle.com/methodology/overview. Afterwards, they would plan their design sprint by physically moving some of the sticky notes from the methods poster to the planner.



Closing thought on user interviews

It is strongly recommended that students plan user interviews early in their design sprint or user demos at the end of their design sprint. I'm convinced that most can be learned by talking to real users. Ideally, show the users the mockup that is the result of the design sprint, as professed in the AJ&Smart videos.

Unfortunately, I could not find a way to simulate this in the workshop, and this showed in the evaluation. None of the pilot-year groups sat down with users in the first two weeks, and students didn't think the workshop helped with interviewing users. Interviewing users in the second half of the project is good for evaluating the software's user experience, changing a colour here and there or changing some user flow. What it doesn't do however, is checking if the idea the team is working on is a good idea at all. For this the team should talk to potential users much much earlier, before investing time in a solution nobody wants. In this regard, it could help that you, as TA, are closer to the students than I was.

TA Workshop v1.0

PowerPoint Presentation

These are slides you can use during the workshop.

Handouts for students

Zip Compressed File

You can download and print these materials before the workshop. Make sure to print one of each document per group (except for the script).

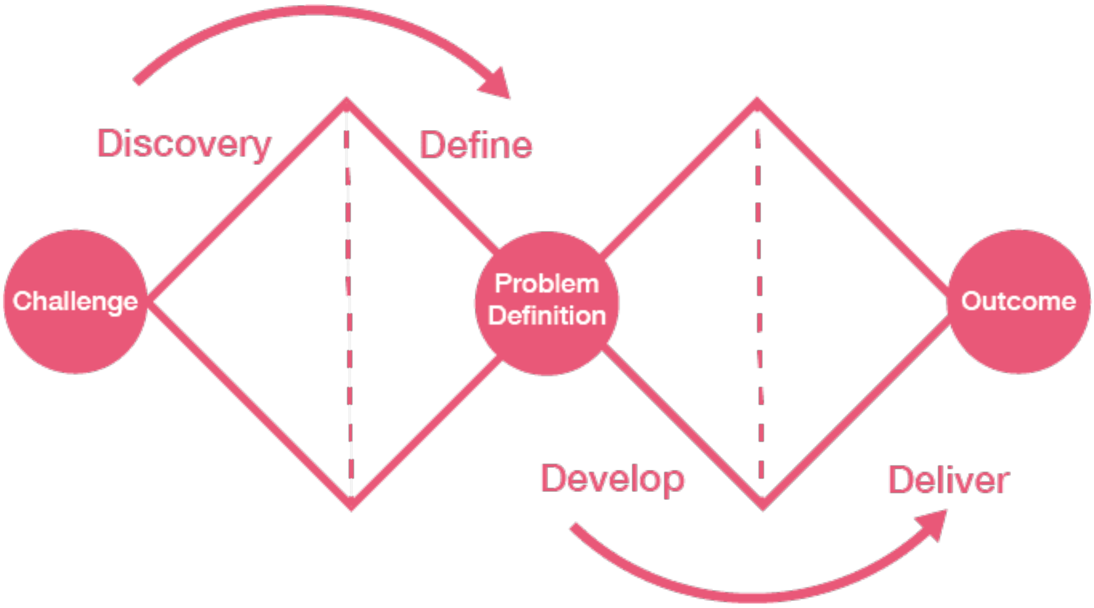
Divergent & convergent thinking

Divergent (<) and convergent (>) thinking in the design sprint (<><><>)

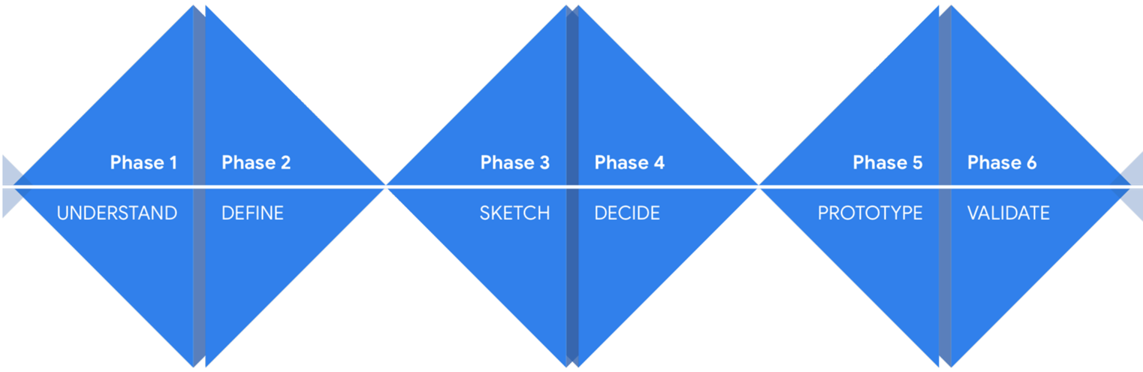
The model is intentionally shaped like a triple diamond (<><><>) to emphasise the difference between divergent (<) and convergent (>) phases. Divergent thinking always comes before convergent thinking, hence the expanding < shape first, followed by a contracting shape >: https://youtu.be/cmBf1fBRXms?si=cuS9GjJUSr4aUDQO&t=82.

This video explains the concept with an older more general model:

The first diamond of the double diamond model is pretty much the same as the first diamond in the design sprint. However, the second and third diamond of the design sprint are lumped together in the double diamond model. The concept is easiest understood with phase 3 (<) and phase 4 (>) of the model. In the sketching (<) phase the group comes up with lots of different ideas broadening (<) the design space in search of a new, creative, never thought of solution. In other words, the design space diverges. In the deciding (>) phase the design space is narrowed to one idea to pursue, it is converging (>). Phase 1, 3 and 5 are divergent phases (<). Phases 2, 4 and 6 are convergent phases (>). For this reason the model looks like three diamonds <><><>. Diverging and converging phases follow each other.



(https://mgearon.com/ux/double-diamond-model/)



(https://designsprintkit.withgoogle.com/methodology/overview)

With this idea in mind the idea behind phase 1 (<) is easier to grasp. It is not about understanding the solution the client wants you to build. Rather, it is about generating as much opportunities (problems to solve) you see as possible.
Only *after* that proces, in phase 2 (>), you converge on one opportunity to pursue, with approvement by the client of course.

Drag and drop files here to create and update topics