

The Use of Agile Methods in ICT Training

Jouko Loijas

Master's Thesis

Information Systems Management

November 2016



<p>Author or authors Jouko Loijas</p>	<p>Group or year of entry 2014</p>
<p>Title of report The Use of Agile Methods in ICT Training</p>	<p>Number of pages and appendices 53</p>
<p>Teacher(s) or supervisor(s) Hietala Heikki Paal Heidi</p>	
<p>This thesis presents how the agile methods are used in ICT training in Amiedu. The word agile is characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans. The benefits of agility that should be attainable are improved lead time, customer happiness, quality, and transparency.</p> <p>Scrum is a lightweight and simple to understand agile framework. It has been in use since the early 1990s and has been widely used in different kinds of projects. The Scrum framework defines Scrums teams and associated roles, events, artifacts, and rules. The eduScrum is a framework that is based on Scrum and it is meant to use in education. Within the framework, the students can tackle complex problems while achieving learning goals and personal growth. Our model of agile education owes a lot to Scrum and eduScrum, but it also has its differences. Our model is a bit stricter about what and when the students are doing, but a bit looser on the events and artifacts than eduScrum is.</p> <p>Comparing the results of different educational projects showed that our model of agile education is competitive in expenses, but has trouble to achieve a good number of graduates. Survey conducted to the students and analyzed with DCA method, shows that we have succeeded in creating a free atmosphere with active social interaction in our courses, but we are still struggling with the efficiency of the project. The biggest factor to overall evaluation seems to be the change that the students achieve from their education. The students in more traditional courses feel that they get more out of the courses than our agile students. Reasons behind the smaller level of change seem to be in the level of engagement and reflection. Both of which are core values in our system.</p> <p>Overall it seems that our model has the right values and we are aiming to do the right things, but we need to find a way to improve the engagement level of the students, which leads to better social interaction, better reflection, and greater change.</p>	
<p>Keywords agile, education, Scrum, eduScrum, ICT training, Amiedu</p>	

Table of contents

1	Introduction	1
2	Agile methods	3
2.1	What agility means	3
2.2	Reasons to work agile	5
2.3	Scrum	6
2.4	EduScrum.....	13
2.5	Teacher as a product owner.....	18
2.6	Our model of agile education	20
2.7	Pedagogical reason for agile education.....	22
3	Research	25
3.1	Methods	25
3.2	Financial results	27
3.3	Graduation	28
3.4	The quality of the studies	30
3.5	Reasons behind happiness	38
4	Conclusion	49
4.1	Analysis of the research results	49
4.2	Ideas for future research	50
	References	52

1 Introduction

We started a new degree program to teach game graphic designers in Amiedu in April 2015. The official title of the degree program is Further Qualification in Audiovisual Communication in Game Graphic Design. The aim of the program is to train professional game graphic designers that are ready to enter into the growing gaming industry of Finland within one year. As a starting level of the students, we defined that they will need to have some experience in ICT and they would have some kind of background in traditional arts or in technical drawing. They would also need to be familiar with gaming and highly motivated in entering the business as an employee or an entrepreneur.

We wanted to make the education flexible, effective and motivating. I have been working with agile methods in other projects to reach the same kind of goals and once I heard about agile pedagogy that Helsinki Business College uses to train programmers, I decided to start researching agile methods that could be used in ICT training. One of my biggest inspirations in the research was the eduScrum methodology that is developed and used in Dutch school system.

In our model, the students are divided into small groups that are hopefully going to stay through the whole education. Instructors will give a deadline and a backlog of tasks to complete in every sprint. The sprints will usually last from two to three weeks and after each sprint, the instructors will go through the tasks and give feedback for the students. The communication between the students is the most important thing for the success of the program and we have given them a minimum of three weekly meetings, where they will go through what they have done, what they will do next and what kind of problems they have met in their tasks. Other than this, the groups can decide themselves how they will perform the given tasks. The instructors will take part in the student's meetings every once in a while, and we follow their learning diaries and group discussions online. These tools will help to make learning as transparent as possible even when we are not sitting in the same room.

Using the experience from the Netherlands and HBC and Scrum methodology, we created a model that we felt that works well in our education and we are currently doing our first implementation with this method. The aim of my research is to find out how motivating, effective and flexible the students find this model to be and how can we improve it. To complete my research, I am planning to make a qualitative analysis of how our students will progress in their studies. I will conduct a survey and conduct a dynamic concept analysis. I am also comparing the results and the costs of the program to other degree programs we are running.

2 Agile methods

This chapter describes what agility means and what kind of agile methods are being used in software development and in education.

2.1 What agility means

The agile Manifesto: *“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:*

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

(Beck et al. 2001)

Words ‘agile’ and ‘agility’ are used widely in software development and more and more in other industries as well, but what is actually behind those buzzwords? There are many definitions for the words. Oxford dictionary defines the word as following:

1. Able to move quickly and easily.
2. Able to think and to understand quickly.
3. Relating to or denoting a method of project management, used especially for software development, that is characterized by the division of tasks into short phases of work and frequent reassessment and adaptation of plans.

(Oxford Dictionaries)

While the third one of the definitions clearly is what I am talking about in this thesis, the first two – more traditional definitions – give also a good image of what the agile should be.

According to Kelly Waters in All About Agile -website the ten key principles of agile are:

1. *Active user involvement is imperative*
2. *The team must be empowered to make decisions*
3. *Requirements evolve but the timescale is fixed*
4. *Capture requirements at a high level; lightweight & visual*
5. *Develop small, incremental releases and iterate*
6. *Focus on frequent delivery of products*
7. *Complete each feature before moving on to the next*
8. *Apply the 80/20 rule*
9. *Testing is integrated throughout the project lifecycle – test early and often*
10. *A collaborative & cooperative approach between all stakeholders is essential*

(Waters 2007)



Figure 1. Waterfall Methodology (Screenmedia)

Agile development is usually compared to traditional waterfall process. As seen in figure 1 a project that follows the waterfall method is strictly set in phases. Each of the phases has a gateway that needs to be passed before getting into the next phase. At first, the whole project is planned and defined. Then all the details are designed before the actual build phase. There should not be any changes to the design in the build phase. After the product is built it will be tested, reviewed and finally launched. This methodology is designed to be safe and clear and fixed phases give a sense of confidence that project will go as it should go. Unfortunately, there almost always are

inevitable changes and unforeseen challenges. It is usually difficult to renegotiate the agreed scope. This inflexibility makes it harder to finish the project successfully and this is the reason why many developers have turned into the more flexible agile development. As seen in figure 2 agile development is often seen as cycles that contain the same phases as the waterfall. Instead of trying to plan, design and build the whole project at once, in agile development it is important to break the project into small parts that can be completed on their own and go through them one by one. (<http://www.screenmedia.co.uk/blog/2014/08/what-is-agile-development-a-brief-introduction/>)

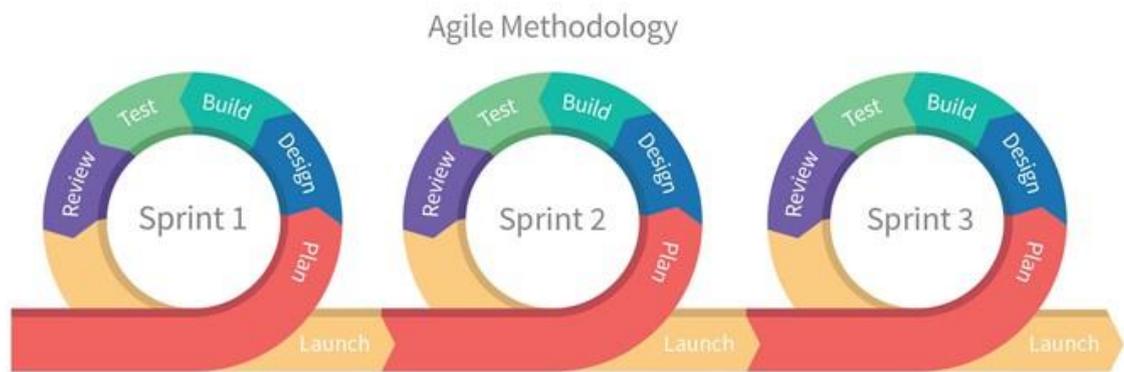


Figure 2. Agile Methodology (Screenmedia)

2.2 Reasons to work agile

What are the reasons behind the use of agile? A common argument for agile development is the cost saving. Another argument is that it makes changes easier during the project. These are not usually directly valid, but might become reality non-directly if the whole organization works well. The real benefits of agility that should be attainable (but are not automatic) are improved lead time, customer happiness, quality, and transparency. (Auer et al. 2013, 26.)

To improve the lead time, the whole organization needs to be agile. It is not enough that only a development team is agile since they might have to wait for some other part of the organization. In many project organizations, there is a problem that the people are working on many different projects and even if the developers would be in only one project they might need to get permissions or direction from the executives that

do not have time to handle all the projects they are involved in. It is vital to keep the ongoing projects in a number that the whole organization can handle. Another way to improve the lead time is to simplify the processes and make sure that only the most important factors of the projects are under heavy control and direction and let the developers decide on themselves on the minor things. (Auer et al. 2013, 26 – 27.)

The customer is usually happy if he gets the right products at the right time. The basic idea of agile development is to do the most important elements first. The most important elements are usually those that add the most value to the customer. Agile development helps to get early feedback from the real users and thus it is important that the customer is represented by those that are affected by the project. A common problem in project organization is that the customer, on the whole, is forgotten while the project teams are all working on their own solutions. (Auer et al. 2013, 27.)

Quality is a combination of many things. In software development, it can mean usability and technical functionality. Agility helps in achieving quality by keeping the development cycles small enough and by testing everything that is produced. The self-improving of the teams working patterns will also help to achieve quality. (Auer et al. 2013, 27 – 28.)

The transparency of the development becomes possible with agile development because it is done in small cycles that always prepares something functional. These functional parts of the program are being seen by the development team and the product owners which helps them to understand where the whole project is going. This gives a much better transparency than the traditional way of only following the costs and percentage of completion. (Auer et al. 2013, 28).

2.3 Scrum

According to the Scrum Guide, the definition of Scrum is following: “*Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.*” (Schwaber & Sutherland 2016, 3.)

Scrum is lightweight and simple to understand but difficult to master. It has been in use since the early 1990s and has been widely used in different kinds of projects. Scrum is not a technique to build products, but a framework to employ various processes and techniques in. The Scrum framework defines Scrum teams and associated roles, events, artifacts, and rules. (Figure 3.) Everything within Scrum has a purpose. The rules of the Scrum combine the events, roles, and artifacts and specifies the interaction between them. (Schwaber & Sutherland 2016, 3.)

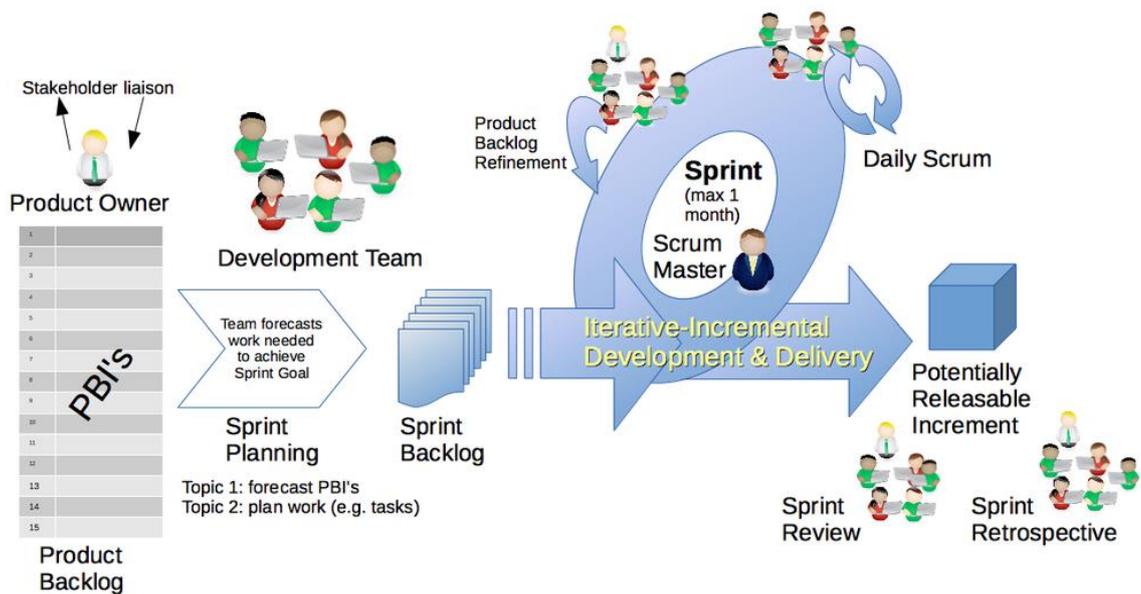


Figure 3. Scrum Framework (Wikipedia)

Scrum is founded on empiricism which asserts that knowledge comes from the experience and decision making is based on what is known. The three pillars of Scrum are transparency, inspection, and adaptation. The important parts of the process must be visible to those responsible for the outcome. To be transparent the most important parts must be defined by a common standard so that everyone has a common understanding of them. Inspection is about knowing where the team is going. It is essential to inspect the artifacts and progress to detect undesired variances, though inspection frequency should not be too high to interfere with the actual work. If an inspector finds aspects that are not within acceptable limits and that the product would not be acceptable, the process or the material must be adjusted as soon as possible. This is called adaptation. The inspection and adaptation are done in formal events of

sprint planning, daily scrum, sprint review and sprint retrospective. The five principal Scrum values of commitment, courage, focus, openness and respect are essential to building up the three pillars of Scrum. (Schwaber & Sutherland 2016, 4.)

The Scrum teams are built of a product owner, the development team and a Scrum master. The teams are self-organizing and cross-functional meaning that they choose their own methods of working rather than are tightly lead from outside and that they have enough competence to accomplish the work without outside help. The team model is designed to be flexible, creative and productive. (Schwaber & Sutherland 2016, 5.)

The product owner is responsible for the value of the product and the work of the development team. The main function of the product owner is to manage the product backlog. According to the Scrum Guide the management includes:

- *Clearly expressing Product Backlog items;*
- *Ordering the items in the Product Backlog to best achieve goals and missions;*
- *Optimizing the value of the work the Development Team performs;*
- *Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,*
- *Ensuring the Development Team understands items in the Product Backlog to the level needed.*

It doesn't matter if the product owner or the development team does these tasks, but the product owner is always responsible that they are done. The product owner should always be a one person - not a committee. The product owner's decisions are visible in the product backlog and no one is allowed to give other tasks for development team – the organization in its whole needs to respect the decisions of the product owner. (Schwaber & Sutherland 2016, 5.)

The development team is built up of professionals. They work in sprints and aim to create a releasable increment of the product at each sprint. The teams organize and manage their own work optimizing the overall efficiency and effectiveness. According to the Scrum Guide the development team has the following characteristics:

- *They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;*
- *Development Teams are cross-functional, with all of the skills as a team necessary to create a product Increment;*
- *Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person; there are no exceptions to this rule;*
- *Scrum recognizes no sub-teams in the Development Team, regardless of particular domains that need to be addressed like testing or business analysis; there are no exceptions to this rule; and,*
- *Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.*

The optimal development team size is from three to nine persons. Less than this easily encounters skill constraints and a larger team might generate too much complexity for an empirical process to manage. The product owner and the Scrum master are not included in the development team count unless they are also working on the sprint items. (Schwaber & Sutherland 2016, 6.)

The Scrum master's key role is to understand Scrum. He is in charge that Scrum is enacted by making sure that the Scrum team adheres to Scrum theory, practices, and rules. Scrum master also helps those outside of Scrum team to interact with the Scrum team to maximize the value created by the Scrum team. According to the Scrum Guide the services of the Scrum master are the following:

Scrum Master Service to the Product Owner

- *The Scrum Master serves the Product Owner in several ways, including:*
- *Finding techniques for effective Product Backlog management;*
- *Helping the Scrum Team understand the need for clear and concise Product Backlog items;*
- *Understanding product planning in an empirical environment;*
- *Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value;*
- *Understanding and practicing agility; and,*
- *Facilitating Scrum events as requested or needed.*

Scrum Master Service to the Development Team

- *The Scrum Master serves the Development Team in several ways, including:*
- *Coaching the Development Team in self-organization and cross-functionality;*
- *Helping the Development Team to create high-value products;*
- *Removing impediments to the Development Team's progress;*
- *Facilitating Scrum events as requested or needed; and,*
- *Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.*

Scrum Master Service to the Organization

- *The Scrum Master serves the organization in several ways, including:*
- *Leading and coaching the organization in its Scrum adoption;*
- *Planning Scrum implementations within the organization;*
- *Helping employees and stakeholders understand and enact Scrum and empirical product development;*
- *Causing change that increases the productivity of the Scrum Team; and,*
- *Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.*

(Schwaber & Sutherland 2016, 6 - 7.)

The events in Scrum framework are used to create regularity and to minimize the need for meetings not defined in Scrum. All of the Scrum events have a maximum duration and each of the events contains inspection and adaptation and they are critical to accomplishing transparency. The most important event in Scrum is the Sprint. It lasts for one month or less. During every sprint, the goal is a product increment that is potentially releasable and defined as “done”. Sprint contains the sprint planning, daily scrums, the development work, the sprint review and the sprint retrospective. Once a sprint is finished, a new one will immediately begin. During the sprint no changes are made that would endanger the sprint goal and quality goals do not decrease. However, it is possible to clarify or re-negotiate the sprint scope between the product owner and

the development team if they gain more accurate information on conditions. The sprint may be canceled by the product owner during the sprint if the sprint goal becomes obsolete. (Schwaber & Sutherland 2016, 7 - 8.)

The sprint always starts with a sprint planning event that is joined by the entire Scrum team. For a month long sprint, the sprint planning has the maximum of eight hours and less for a shorter sprint. Sprint plan answers to the following questions:

- *What can be delivered in the Increment resulting from the upcoming Sprint?*
- *How will the work needed to deliver the Increment be achieved?*

The first topic results in what items from the product backlog are taken into the sprint backlog. Once that is decided the development team will decide how they will work to achieve these goals. (Schwaber & Sutherland 2016, 9 - 10.)

The Daily Scrum is a maximum of 15-minute event that is used to synchronize the activities of the team and create a plan for the next 24 hours. It is held at the same time and at the same location every day. During the daily scrum everyone in the development team should answer the following questions:

- *What did I do yesterday that helped the Development Team meet the Sprint Goal?*
- *What will I do today to help the Development Team meet the Sprint Goal?*
- *Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal?*

Only the development team is allowed to join the daily Scrums, but the Scrum master ensures that the meetings are held. These meetings are important to improve communication and decision making. They will also help to remove the need for other meetings. These are essential for inspecting and adapting the work of the development team. (Schwaber & Sutherland 2016, 11.)

In the end of every sprint is sprint review. During the sprint review the Scrum team and the stakeholders discuss what was done in the sprint. They also inspect and adapt the project backlog and think about what to do in the future. It takes a maximum of four hours for a month-long sprint. According to the Scrum Guide the sprint review includes the following elements:

- *Attendees include the Scrum Team and key stakeholders invited by the Product Owner;*
- *The Product Owner explains what Product Backlog items have been “Done” and what has not been “Done”;*
- *The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved;*
- *The Development Team demonstrates the work that it has “Done” and answers questions about the Increment;*
- *The Product Owner discusses the Product Backlog as it stands. He or she projects likely completion dates based on progress to date (if needed);*
- *The entire group collaborates on what to do next so that the Sprint Review provides valuable input to subsequent Sprint Planning;*
- *Review of how the marketplace or potential use of the product might have changed what is the most valuable thing to do next; and,*
- *Review of the timeline, budget, potential capabilities, and marketplace for the next anticipated release of the product.*

The sprint review should produce a revised product backlog that defines a probable product backlog for the next sprint. (Schwaber & Sutherland 2016, 11 - 12.)

Finally, after the sprint review, there is a sprint retrospective, that is meant for the whole Scrum team to inspect itself and create a plan for improvements for the next sprint. Retrospective lasts a maximum of three hours for a month-long sprint. The Scrum Guide gives the following tasks for the sprint retrospective:

- *Inspect how the last Sprint went with regards to people, relationships, process, and tools;*
- *Identify and order the major items that went well and potential improvements; and,*
- *Create a plan for implementing improvements to the way the Scrum Team does its work.*

(Schwaber & Sutherland 2016, 12 - 13.)

The product backlog is a list of everything that might be needed in the product. Everything that is done for the product is first put into product backlog and it is the only place where items are taken to the sprint backlog for the development team to work on. Product owner takes care of the backlog and puts the items in it and

prioritizes them and together with the development team chooses which items are taken to sprint backlog in every sprint. (Figure 4.) The product backlog is under inspection and adaptation throughout the project and a lot of items might never be included in the final product. The items in backlog are also defined more carefully when they are getting more prioritized. The higher ordered items are usually much clearer and more detailed than the lower ordered ones. When a product backlog item or an increment is described as “done”, everyone must understand what “done” means. One important part of managing the backlogs is that all items need to have a definition of “done”. (Schwaber & Sutherland 2016, 14 - 16.)

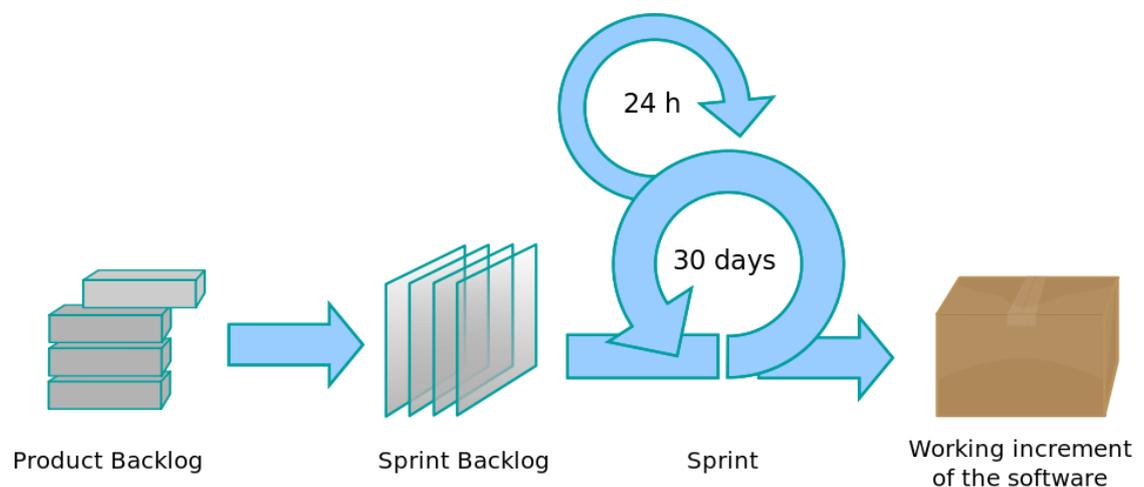


Figure 4. The Scrum Process (Wikipedia)

2.4 EduScrum

The eduScrum is a framework that is based on Scrum and it is meant to use in education. Within the framework, the students can tackle complex problems while achieving learning goals and personal growth. As Scrum, the eduScrum is lightweight and easy to understand, but difficult to master. It has been developed for the classroom environment to work with children. The idea is that the children are not being held responsible, but are themselves feeling responsible for completing the work. Teachers won't be telling to the children what to do and when to do it, but they are only telling what the expected results are. The model heavily leans on Scrum and the

eduScrum Guide is written in pretty much the same form as the Scrum Guide is. (Beaumont, Delhij & Gerrits 2015. 4 – 6.)

The eduScrum has the same basic pillars as the Scrum does - transparency, inspection and adaption. In eduScrum team, the teacher works as a product owner and students form a four-person student team of which one is working as an eduScrum master. The product owner determines what needs to be learned and monitors the improving the quality of educational results while evaluating and judging the educational results. (Beaumont et al. 2015. 8.)

As the development team in Scrum, the student team is self-organizing and autonomous and should be multidisciplinary with all the required skills to be able to achieve the learning goals and develop. This is important so that all the students in the group have their strengths and are able to help the group. From a student group, it is not always possible to find people with all the skills, but the group can be built so that the students have different areas of interests and they want to focus their learning in different skill-sets. The responsibility of completing the goals lies on the team as the whole. The student team uses the acceptance criteria and the definition of done to track their own progress and quality level. (Beaumont et al. 2015. 10.)

One of the students within the team is assigned as the eduScrum master, that works as a serving and coaching leader of the team. Their job is to help the team to perform optimally, but they do not direct the team. The role of the eduScrum master is more constrained than the role of the Scrum master because the teacher as a product owner takes some responsibilities that would fall on Scrum master in a Scrum team. The eduScrum master serves the product owner by creating transparency on progress and facilitates the eduScrum events when needed. In addition to this, he serves the student team by ensuring correct execution of the eduScrum and facilitates cross-team collaboration. (Beaumont et al. 2015. 11.)

The eduScrum has events like Scrum and the one that includes everything else is the sprint. The sprints are usually within a semester or a period and they start with sprint

planning meeting that includes team formation. Stand ups are held in every class and at the end of the sprint there sprint review and retrospective including a personal reflection. During the sprint, the student teams are not changed nor the scope is changed, but there might be clarifications and quality may be renegotiated between the product owner and the student team. Unlike in Scrum, the sprints may not be canceled in eduScrum. (Beaumont et al. 2015. 12.)

Team formation is essential in eduScrum and according to the eduScrum Guide to achieve good team compositions the following things should be notified:

- *qualities of team members are complementary;*
- *balanced ratio of sexes;*
- *different compositions than those during previous assignments;*
- *composition based on friendship is undesirable.*

During the team formation, the eduScrum masters are appointed, who will then choose a team of people with the right skill set. (Beaumont et al. 2015. 13.)

Once the teams are formed it is time to set the learning goals. They should be flexible enough to give the students to plan what and how to deliver during the sprint. Teacher as a product owner tells what is expected of the team at the end of the sprint, but the students themselves decide what to do to get into those goals. Once the goals are set starts the work planning phase. The product owner sets the boundaries where to work, such as how many lessons there are in the sprint and what are the evaluation models. Then the student team starts to organize the tasks and decides who will do what. It is also good to make partial goals and checkpoints for the sprint so that the work is transparent and everyone knows what everyone is doing. At the end of the sprint planning meeting, the student team should be able to explain to the product owner how they will reach the sprint objectives. (Beaumont et al. 2015. 13 - 14.)

The stand-ups are done like in Scrum, but instead of every day they are done at the beginning of every lesson. The questions to answer are a little different than in Scrum:

- *What have I done to help the team achieve the Sprint goal since the previous class?*
- *What will I do this class to help the team achieve the Sprint goal?*

- *What are impediments that block me or the team to achieve the Sprint goal?*

The stand-ups help the team to reach the learning goal. During the stand-ups, the team evaluates the progress and explain to the product owner how they will work together as a self-organizing team. The eduScrum master ensures that the team does the stand-ups, but the team as whole executes them. (Beaumont et al. 2015. 15.)

Sprint reviews are held at the end of the sprint. The student teams display what they have learned in comparison to the learning goals. And finally, the sprint retrospective is an opportunity for the student team inspects themselves. During the retrospective, the students should make a plan to improve themselves as a team and as individuals. In retrospective, the students should evaluate the work methods of the team, evaluate their own and their team members' skills and the points of improvement. Finally, the students should discuss these points. (Beaumont et al. 2015. 16.)

The eduScrum artifacts are the product backlog, the flip, the definition of done and the definition of fun. The product backlog is the list of the learning goals that is being kept up to date by the teacher as a product owner. In eduScrum, the core goals and the learning goals are often well known before the project starts since they are usually derived from the curriculum of the subjects being taught. The items in the backlog do get clearer and more finely defined during the time though and the items high on the list are usually more detailed than the ones low on the list. This is usually due to the fact that the teacher is able to clarify the learning material and the definitions of done are being made. (Beaumont et al. 2015. 17.)

The flip derives its name from the flipchart and is used to maintain the set of the task in the current sprint. It is presented chronologically and the tasks and assignments move their status between 'to do', 'busy' and 'done'. (Figure 5.) It helps the student team to know what to do and it also keeps up the transparency of the progress, since the flip must be visible for all student teams during each meeting. As new work is required the student team adds it to the flip and if some elements are deemed unnecessary, they are removed. (Beaumont et al. 2015. 17 - 18.)



Figure 5. The Flip (eduScrum Ashram College)

As in Scrum, it is very important that everybody has the same view of what 'done' means. When a learning goal is defined as done the whole team and the product owner should understand what it means. The learning goal is the sum of all items to be completed during the sprint and at the end of the sprint it must be done and meet the predefined acceptance criteria. When the student team is defining the done, they should think how it can be checked if they are really done, what criteria must be met and also when is it not done. One artifact that is in eduScrum that you don't find in Scrum is the definition of fun. To maximize learning it is essential that the students

have fun while they are learning. This means that they should think of what and how they do to ensure enjoyable work. (Beaumont et al. 2015. 18 - 19.)

2.5 Teacher as a product owner

The product owner is responsible for managing backlog and ensuring the value of the work that the team performs. The responsibility to lead the development effort to create a product that provides desired benefits belongs to the product manager. To manage this the product manager needs to create a vision of the product, groom the product backlog, plan the release and involve the stakeholders. (Pichler 2010, 2.)

As a responsible teacher of an educational project, the responsibilities are to create a vision for the learning outcome of the students. In degree programs, there are certain guidelines that are given by the Ministry of Education, but those guidelines are usually quite loose and leave a lot for the teachers to decide. The responsible teacher needs to create a schedule for the educational project and decide which items the students should learn and in which order. The responsible teacher also needs to communicate with the students, other teachers, consultants and visiting lecturers and usually the teacher has some kind of economic responsibility of the project as well. When these responsibilities are compared to the product owner's responsibilities, it is found that it's not a big leap from a responsible teacher to a product manager. And that is the main role of the responsible teacher in our model of agile education.

Pichler describes some of the most important characteristics of a good product owner. Product owner needs to be a visionary, a doer, a leader and a team player. Product owner also needs to be a good communicator and negotiator. And finally, a product owner needs to be empowered, committed, available and qualified. (Pichler 2010, 4 – 7.)

The product owner needs to be able to envision the final product and communicate that vision, but he also needs to be a doer that ensures that the vision gets completed. To accomplish this the product manager needs to collaborate closely with the team. Product owner facilitates and encourages the team and can cope with change, debate,

conflict, and experimentation. He is the responsible of the product's success and provides guidance and directions, but he has no formal authority over the development team and he must be able to collaborate – not dictate. To accomplish this the communication is the key function of a product owner. He needs to be able to communicate with all the stakeholders on the level that he is understood and efficient. The way to do this might differ a lot whether the counterpart is an engineer or a sales rep or a student or consultant teacher. Sometimes the communication needs to be straightforward and there are times that “no” is the only answer while sometimes negotiation brings out the best results. (Pichler 2010, 4 – 6.)

The product owner needs authority and management sponsorship to lead development and align to stakeholders. He needs to be someone who can be entrusted with a budget and he needs to be able to create a work environment with creativity and innovation. A successful product owner is confident, enthusiastic, energetic and trustworthy. The job of the product owner should be full-time and it is very important that product owner is available for the team when needed. He also needs to have an understanding of the customer and the market. (Pichler 2010, 6 – 7.)

Pichler also writes about common mistakes in the role of the product owner that will effectively ruin the good qualities he might have. These are the underpowered product owner, that needs to ask for permission for the top to perform. This will lead to delays and loss of confidence from the team. Being overworked is as bad as being underpowered and leads to similar problems. Some organizations split the role of the product owner into two. Another one is outward-facing and has the vision and understanding of the market while the other is inward-facing and works with the sprints and the team. This arrangement blurs responsibility and authority and wastes a lot of time. A distant product owner works away for the team. The distance may derive from geographical locations or simply from work practices. A distant product owner may lead to mistrust, miscommunication, misalignment and slow progress. Face-to-face conversation is the key avoid this. (Pichler 2010, 16 – 19.)

Teacher as a product owner needs to work close with the students and have a lot of face-to-face conversations with them. Sometimes it's good to meet the students one at a time, sometimes by development groups and sometimes talks to the whole class, but it is most vital that teacher is available for the students. Having too many students to guide will ruin the work as easily as being too distant from the students. The underpowered teacher might also be a problem if the work of the students slows down because the teacher needs to ask permission a lot from often very busy superiors.

2.6 Our model of agile education

About two years ago, we decided to start teaching game graphic designers in Amiedu. We wanted to achieve a good quality of training with fewer contact lessons than in our more traditional studies. The target group was the adult people that wanted to work in the game industry but had not much of past experience and we wanted to train them to a level that they would be able to work in a game development group. What made this goal quite ambitious was that we only had one year to work with them, since that is the time that we can use in these degree programs. Another thing was, that we wanted them to be able to study mostly away from the classroom. Once I had heard of the eduScrum I presented the idea of agile education for our team and we started to plan a model that work in this scenario.

Our model of agile education owes a lot to Scrum and eduScrum, but it also has its differences. Our model is a bit stricter about what and when the students are doing, but a bit looser on the events and artifacts than eduScrum is. Our permanent teachers work as guides that combine the roles of the product owner and the scrum master. We find it important that there is more than a teacher to take this part so that there is always some that the students can reach in reasonable time. We have also seen it improve the quality of guidance that there are at least two teachers that know the students and can discuss them. Teachers make sure that the students have the learning goals and a higher level schedule to work with.

We start a new program with an intensive week of classroom-oriented studying that has some games and fun and lots of groups working and talking included. The main

goal of this week is that everybody gets to know each other and we as guiding teachers get as much information out of our students as possible. The students will also conduct a self-evaluation of their skills and character. In the end of the week, the guiding teachers will form groups of four to seven people, depending on the size of the class. These groups will then start working on their own game project that will last about half a year.

The project is split into sprints and every sprint has their goals. The sprint goals are given by the guiding teacher, but the students decide how they work to achieve those goals. The first task for every group is to decide how they will work – what kind of roles, communication, distribution of tasks etc. they have. They need to write them up and everybody in the group needs to agree on working with those principles. Once the first sprint starts the students will mainly work away from school and they may also decide how often they see each other, but we strongly recommend that they would meet in person at least once a week and use online meetings at least three times a week.

The sprints last about two to four weeks and they contain at least one contact day and one group meeting with the guiding teacher. The contact days are mostly held by a professional from game industry that teaches about the topics are the learning goals of the sprint. To prepare for the contact day, the students always have some task to learn about the theory of the topic. These tasks might be to read an article or watch a video tutorial and find some answer to some questions and they are given in the previous sprint – thus the guiding teacher should already have planned the contact day, one sprint ahead. After the contact day, the students are given a task to implement the learned into their game project. For example, in the second sprint, we gave a task for students to watch tutorials of an animation program and read through articles about classic animation. Then at the end of the second sprint, there was a contact day, where the classroom teacher teaches about animation with some practical hands-on methods. In the third sprint, the students create animations that they are using in their game project and start learning the theory of the next topic – level design.

Every sprint has a deadline which basically means that the sprint ends and the new one starts. Unlike in Scrum we don't have a sprint review and retrospective immediately, but we leave a few days for the guiding teacher to go through the work of the students and after that, we have a group meeting with each of the student groups. These meetings last about an hour and during that meeting we give some feedback of how they met their learning goals, but more importantly we are there to listen how the group is functioning and how the students are working. The main job of the guiding teacher in these meetings is to facilitate the best possible learning outcome of the group. We usually have questions like who did what, how did their communication work, how did they meet their own deadlines, how is the group being led etc.

One important part of the studying and making the project transparent is to have a development blog. Each of the groups assigns one person to be the main responsible of the development blog. The blog responsible makes sure that the development of the project is being written into the blog in every sprint. The blog should contain the most important things that the students are learning and everything that is implemented into their game project.

In the end of the half year project, we have a full day gathering where all the groups present their work – hopefully, a small playable game or at least a demo version of it. The guiding teachers and the students give final feedback of the outcome. In the presentation, all the groups will also tell how their project went and how did they meet the learning goals and how did the group work together. After this, we gather up all the best practices the groups came by and all the worst traps to avoid and use them to create guidelines for the next project. Once this is done, the guiding teachers will create new groups and they will start another six months' project.

2.7 Pedagogical reason for agile education

Agile learning is a structured form of group learning, that finds its form from the work-life. The group learning is widely used in education and it has some proven benefits. First of all, the group gives a possibility for socio-cognitive conflicts. This means that individuals in a group have different ways of thinking and seeing the things

they are working with. Comparing the own way of thinking to the others may lead to the higher motivation of learning. Also if the group has a common goal, they often motivate each other. Even if the work would not be interesting the motivation is easier to keep up, when the others need the student's output. (Repo-Kaarento 2007, 24 – 25.)

Another benefit of working together is that often the students can achieve a higher level of skills when they are helped by the group, than what they could do alone. And once the student has achieved this level in the group, it is easier to do it again alone. When one member of the group is having difficulties to learn, the rest of the group may help him to achieve the goals. (Repo-Kaarento 2007, 25.)

For a group to be really efficient for everyone in it, it needs to have common goals and the students should be positively dependent on each other. This means that they should need each other to get to the goals and the competition within the group should be minimal. If everyone in the group has their own responsibilities and everyone is committed to the goals of the group, it is possible to reach the aforementioned benefits from the group learning. To enhance this, it is good to have a heterogenic group where each student has their own strengths and weaknesses. (Repo-Kaarento 2007, 37 - 38.) In agile frameworks, the multi-disciplinary is often mentioned in group forming to maximize the efficiency of the group.

Teacher's role in group learning is to facilitate the working and learning. The teacher should create an environment where learning is possible and plan the goals of learning for the groups and tasks how to achieve them. The teacher should be able to observe the work of the groups and be available for the groups when needed. (Repo-Kaarento 2007, 44.) Teacher as a product manager works as a guiding and supporting entity whose main job is to facilitate the learning.

In group learning, there are certain phases that need to happen for the group to function. First is the forming of the group when the students usually need rules and guidance from the teacher to understand what to do and how to work. Then comes the storming phase when the students have known each other for a while and the first

conflicts arise. After that comes the norming when the group agrees on their own rules and working methods. When this is done the group starts performing and finally the group adjourns when the work is done. (Repo-Kaarento 2007, 72 - 74.) The frameworks of agile working and learning are created to help the group to smoothly transition between the phases and to maximize the performance.

In most of the jobs, the team working is also quite common. The students are not only learning the substance when studying in the groups, but they are also learning vital group working skills that they need in the work-life. The agile methods are very common in software development, but they are getting more and more common in other industries as well, so learning the agile way of working while studying is also a great benefit for the students.

The structure of working is also important to enable good results from the group studies. One model of a good group working is first to tune into the group by having a sit-chat, then to have a conversation to set up the goals for the group. Once this is done it is time to start working and after the work there should be evaluation and reflection of the work and use this to adjust the working methods and set new goals and share the work. (Repo-Kaarento 2007, 78 - 79.) This model is quite close to the way of working in all of the agile frameworks presented in the previous chapter.

3 Research

The research questions are:

1. How expensive is our model of agile education compared to more traditional ways?
2. How many of the students graduate compared to more traditional ways?
3. How motivating, effective and flexible do the students find our model of agile education compared to more traditional ways?
4. What reasons affect to the overall happiness of the students?
5. How can we improve our model of agile education?

3.1 Methods

For my first two question, I looked into the reports of the education projects in Amiedu ICT and simply compared the results of different educational projects.

To find out the answers to my last three research questions, I conducted a survey for the students of Amiedu ICT. To answer the third question, I compare the results of the survey between the game students and other ICT students of Amiedu.

For the fourth and fifth question, I use a method called Dynamic Concept Analysis (DCA). It has been developed by Seppo Kontiainen for studies of human behavior. The DCA method tries to combine nomothetic (relating to the study or discovery of general scientific laws) and idiographic (relating to the study or discovery of particular scientific facts and processes, as distinct from general laws) approaches so that the same source of information could be used for individual cases and generalizations. DCA has been successfully used in research of educational studies and it serves well the goals of this thesis (Kontiainen 2002, 26).

The following list defines the basic concepts of DCA method as Kontiainen has described them:

1. *Reality refers to an entity in which people live, and in which different things occur, and from where concepts for study are derived.*

2. *The concept refers to any variable or quality used in describing a particular state or process.*
3. *Dynamic Concept Analysis (DCA) is a strategy for building an information matrix, and for using information about concept relations in building conceptual models.*
4. *An information matrix here is a matrix of concept relations on which the relations between all the concepts included in a study are plotted, and from where information is taken for nomothetic or idiographic analysis.*
5. *A conceptual model is a model of concept relations built from the information available in an information matrix.*
6. *The process of change is indicated by changes in the conceptual models as related to time.*
7. *Planning of change means the analysis of a current conceptual model, and simulations derived from it, in order to produce new relations between the concepts used, or to add new elements or new concepts to the conceptual analyses and models in order to identify alternatives or directions for change.*
8. *Dynamic can be understood here in two different ways: (a) as referring to the dynamic nature of concept relations in a conceptual model, or (b) as referring to changes in conceptual models as a function of time.*

(Konttinen 2002, 27 – 28)

The problems in studying human behavior, that DCA aims to tackle is the subjectivity of how people feel about reality. Konttinen says that the collected information is often too dependent on the experience and personal interpretations to form a holistic view. DCA tries to achieve a good enough holistic view while taking into consideration the parts of the whole. (Konttinen 2002, 28 – 29)

In DCA method the idea is to build an information matrix to bring together information and data considered central in particular phenomenon. According to Konttinen the matrix will work as a source of information for conceptual analysis. (Konttinen 2002, 33)

According to Konttinen the following stages are common in building an information matrix:

1. *Select relevant concepts for the information matrix.*

2. *Identify the nature of the concept; use specifications if possible.*
3. *Analyze in each row (or cell) of the matrix how the two concepts in question are related to each other by using information already available in one or more studies, or by constructing hypotheses about the nature of relationships.*
4. *Build the information matrix by plotting the information of concept relations onto it.*

(Konttinen 2002, 42)

3.2 Financial results

How expensive is our model of agile education compared to more traditional ways? To find out the answer to this question, I looked into the project outcome reports of the education projects in Amiedu ICT. I compared five different projects that have been completed recently. Two of the projects use our model of agile education. The two of them are more traditional classroom oriented training and one is a course where students go to the school for one evening per week and study also in their own work. I will call these projects Agile 1 and 2 and Traditional 1 and 2 and Evening. The Agile 1 is our first ever implementation of our agile model while the Agile 2 is the second implementation.

The reports include the specified revenues and expenses of the projects. Since these projects are degree programs mostly funded by the government, the revenues are straightly related to the number of students attending. There are minor differences on the amount of money received per students and each of the courses has some special revenues and expenses that are not seen in this report. So these numbers are a bit simplified, but they give a general understanding of the financial results of these projects. Most of the expenses are generated by personnel expenses and of the use of external consultants. Differences in other expenses are insignificant. (Amiedu 2016)

The most interesting key figure of these projects is the operating profit. Out of these five projects the order of operating profit from best to worst is:

1. Agile 2
2. Evening
3. Traditional 1

4. Agile 1
5. Traditional 2

The difference between the best two is very small and there is quite a gap to the third and fourth that are again quite close to each other and the last one is way below the others. The degree program of Further Qualification in Audiovisual Communication in Game Graphic Design was the first time ever executed in Amiedu in Agile 1. Agile two was for the same degree program, but the second implementation and the other three have been going on for years. Because of this the Agile 1 had a lot of expenses that are no longer needed in the following projects. (Amiedu 2016)

3.3 Graduation

How many of the students graduate compared to more traditional ways? Each of the education projects I am looking into, last for a year. This time is for preparatory studies that prepare the students for passing the degree. After the students are prepared to pass the degree, they will need to do a vocational skills demonstration, that is evaluated by a committee that includes a representative of working life, a representative of an employer and a representative of the learning institution. The student may demonstrate the whole degree at once or do parts of the degree one by one. Once the demonstrations are approved, the student will pass the degree and get their certificates.

As seen in figure 6, the amount of completed preparatory studies is slightly lower in agile education projects than it is in traditional projects. In completed degrees, Traditional 1 is way ahead of the other projects. This is partly because of the difference in giving the demonstrations. In Traditional 1 the demonstrations are given by lab exams, that simulate the real-life situation, while in the other projects the demonstrations must be given in real work projects.

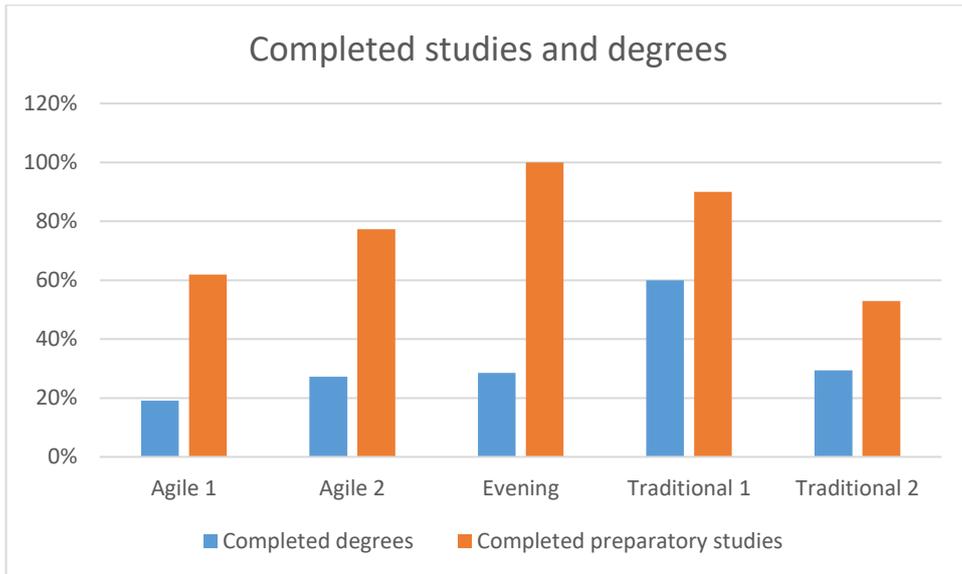


Figure 6. Completed studies and degrees

In figure 7 is shown the expected amount of degrees. The reason for this table is, that there are still some students giving their demonstrations in both Agile 1 and 2 and in Traditional 2. If the expectations are met and the students still doing their demonstrations pass their degrees, the percentage of passed degrees in agile and more traditional projects are quite close to each.

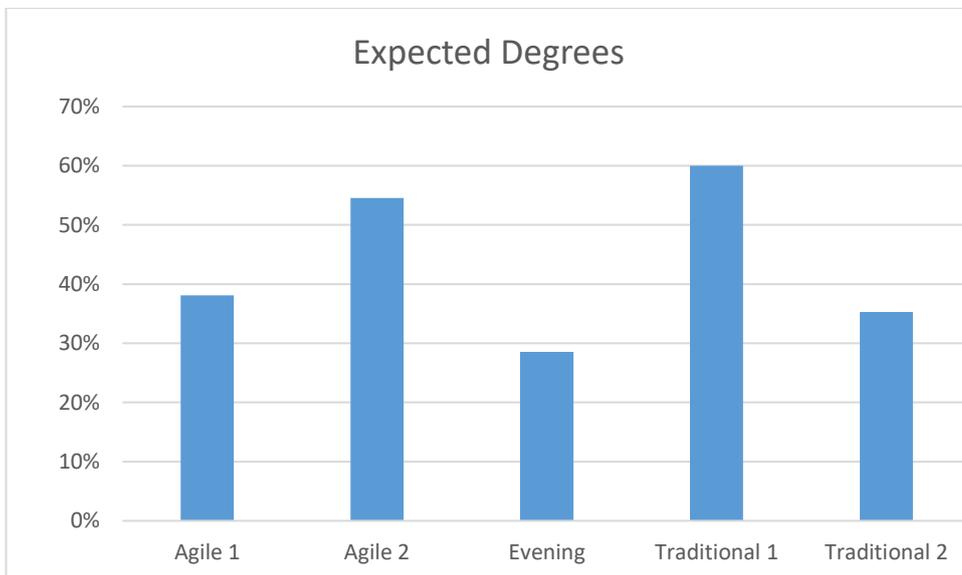


Figure 7. Expected degrees

3.4 The quality of the studies

How motivating, effective and flexible do the students find our model of agile education compared to more traditional ways? To find an answer to this, I conducted a survey for all of my game developer groups that are going on at the moment. This includes three groups of game graphic designers and one group of game programmers. I also asked replies from four groups of our other ICT students – two that are studying in evening studies once a week and two groups that are attending to classroom studies three days a week. The survey started by asking what degree they are studying, in what age category they belong to and how long have they been studying with us. These questions were for background information for me to categorize the answers.

There were 12 questions concerning how the students feel about their own studies. The questions were about students' role, engagement, motivation, importance of contents, social interaction, study atmosphere, approach, reflection, amount of memorized content, quality of learning, change, and overall evaluation. In the following figures, I will compare the answers of the game graphic designers and the other ICT students. I decided to leave the game programmers out of this comparison due to a low number of replies. Some of the figures give a straight feedback about the method of learning and some of them give more information when they cross-referenced with the other figures.

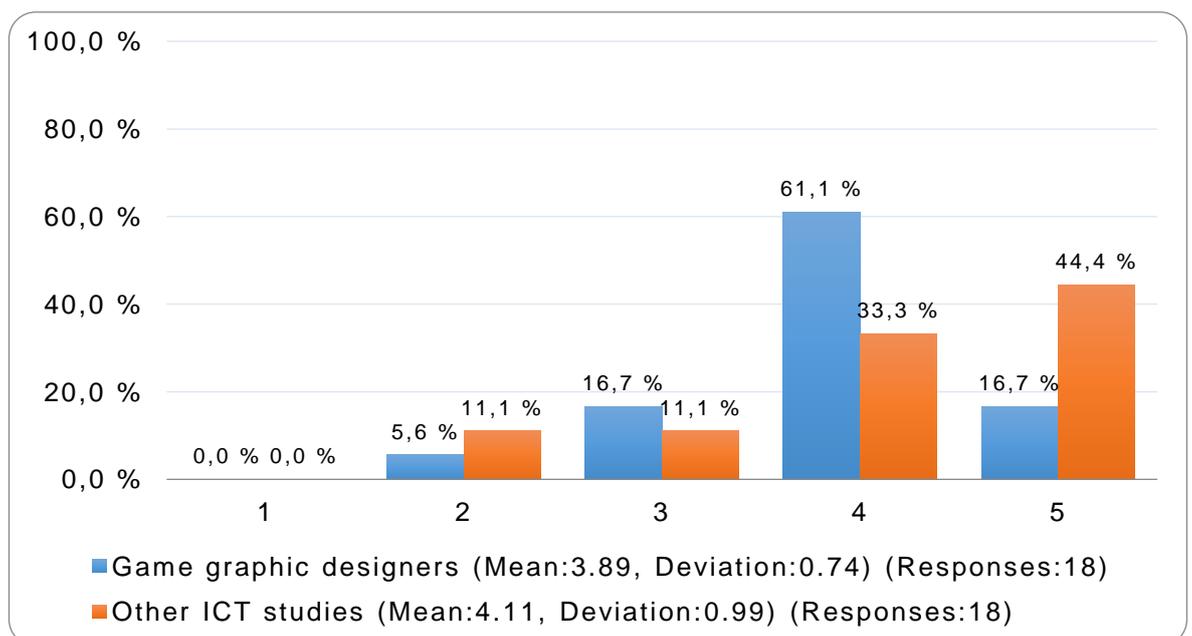


Figure 8. Overall evaluation of the studies (1 = Negative, 5 = Positive)

In figure 8 is shown how the game students and the other ICT students feel about the studies. While both of the groups are mostly positive, the overall score is much higher for the other ICT students. The biggest difference is that most of the game students from agile projects score four out of five, while the traditional ICT students' most common score is full five. When all the fours and fives are combined to see how many feel positive about the studies both of the groups score the same 78%. On the other end, only 6% of the game students give a negative score against 11% of the other ICT students.

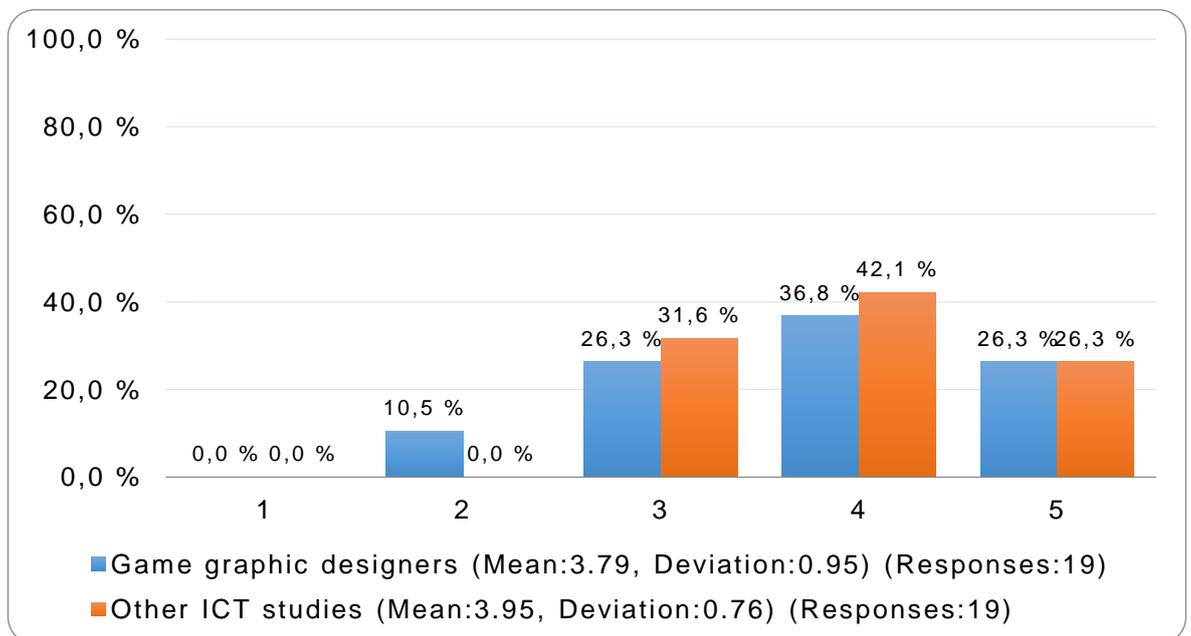


Figure 9. Change (1 = Small, 5 = Great)

The figure 9 shows how the students feel that the studies have impacted them. They were asked if the studies created a lot of new and usable knowledge and enhanced the potential to take new kind of jobs and if the studies helped to achieve greater control and engaged in independent learning and self-development. And also is the studies enhanced the problem-solving skills and self-confidence and helped to understand new situations. Once again the traditional projects get higher scores with the average of 3.95 while the game studies average 3.79. The interesting number in this figure is that 10

percent of the students in agile projects feel that the change has been quite small. Other than that the scores are quite close to each other.

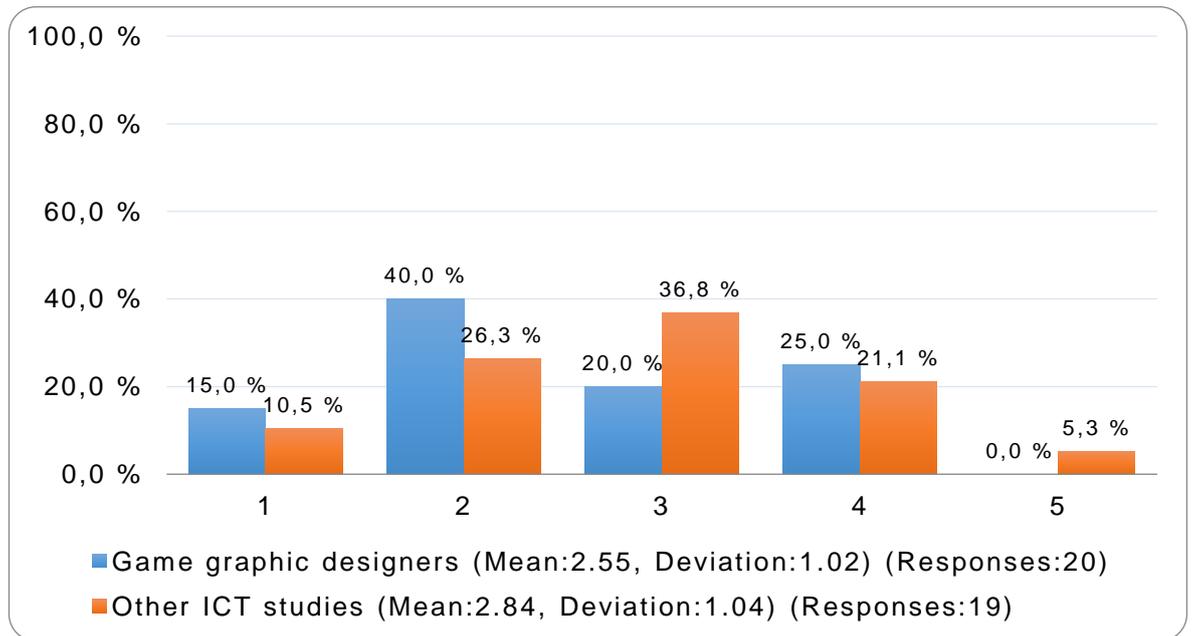


Figure 10. Student's role (1 = independent, 5 = controlled)

Figure 10 presents how the students feel about their own role in the studies. They were asked how they felt about their possibility to control their own studies and make decisions for example of contents and implementation. This figure shows that the students in agile projects feel that they have more control in their own studies. 55% of the agile studies feel that they have a lot of independence in their studies while only 37% of the other ICT students feel the same. A quarter of the both groups feels that the studies are controlled.

Figure 11 presents how the students feel about their own exertion and activity in the studies and figure 12 presents how the students feel of their own motivation.

While most of the students feel that they are strongly involved, it clearly shows that the other ICT students feel more committed to their studies than the game students. Still, the motivation seems to be pretty much the same in both groups.

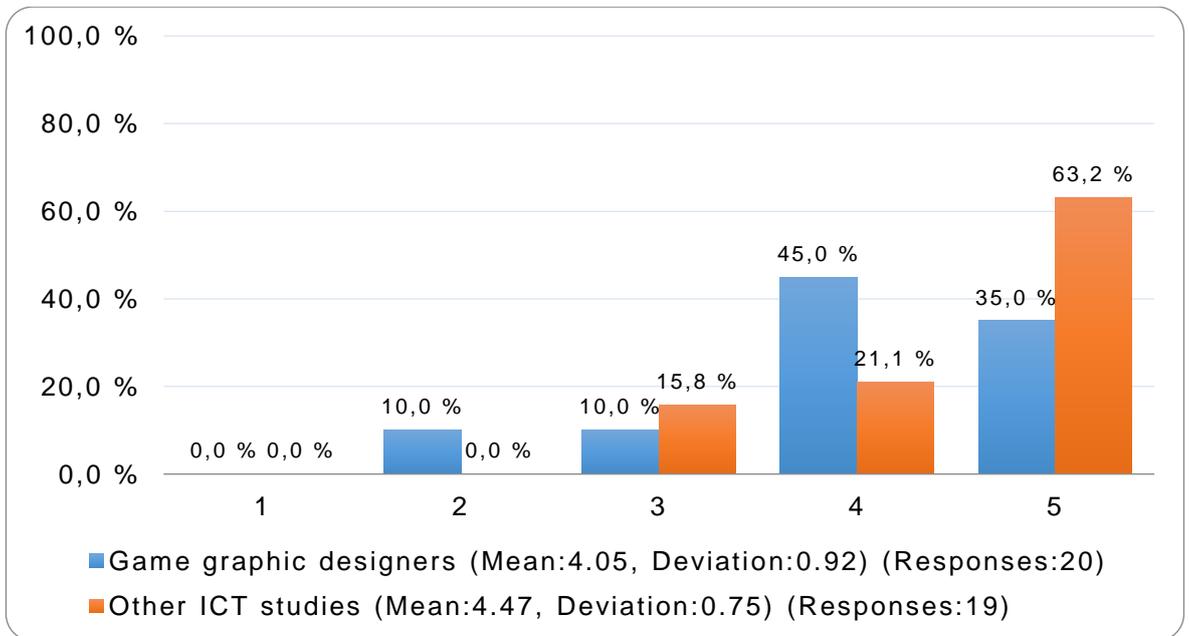


Figure 11. Engagement (1 = Weak, 5 = Strong)

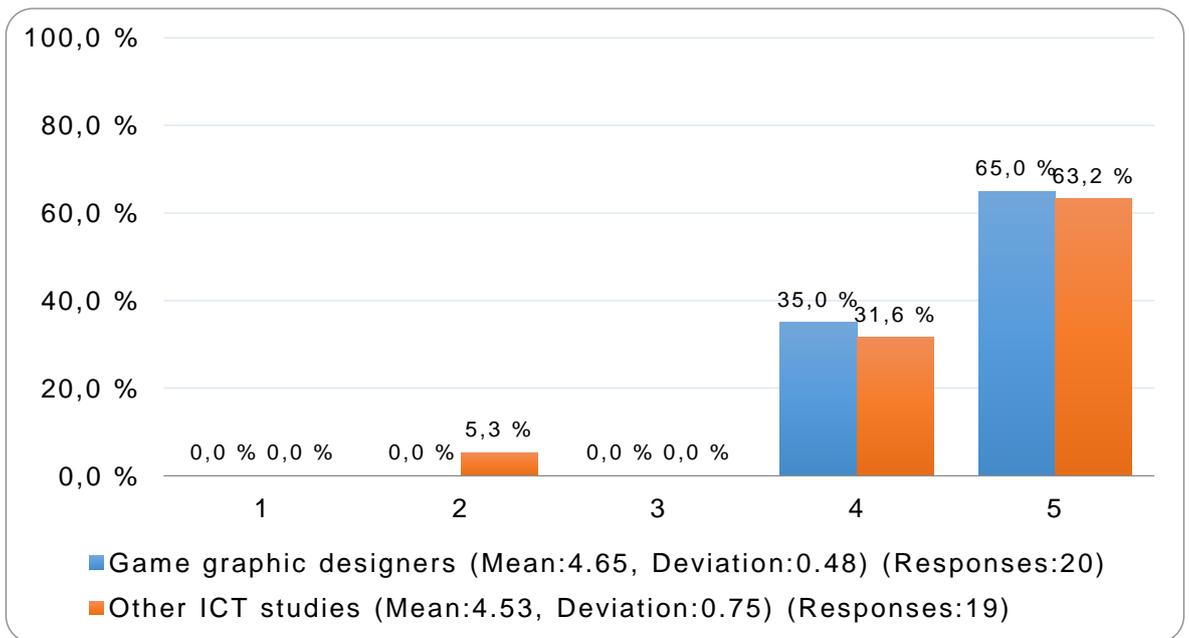


Figure 12. Motivation (1 = Weak, 5 = Strong)

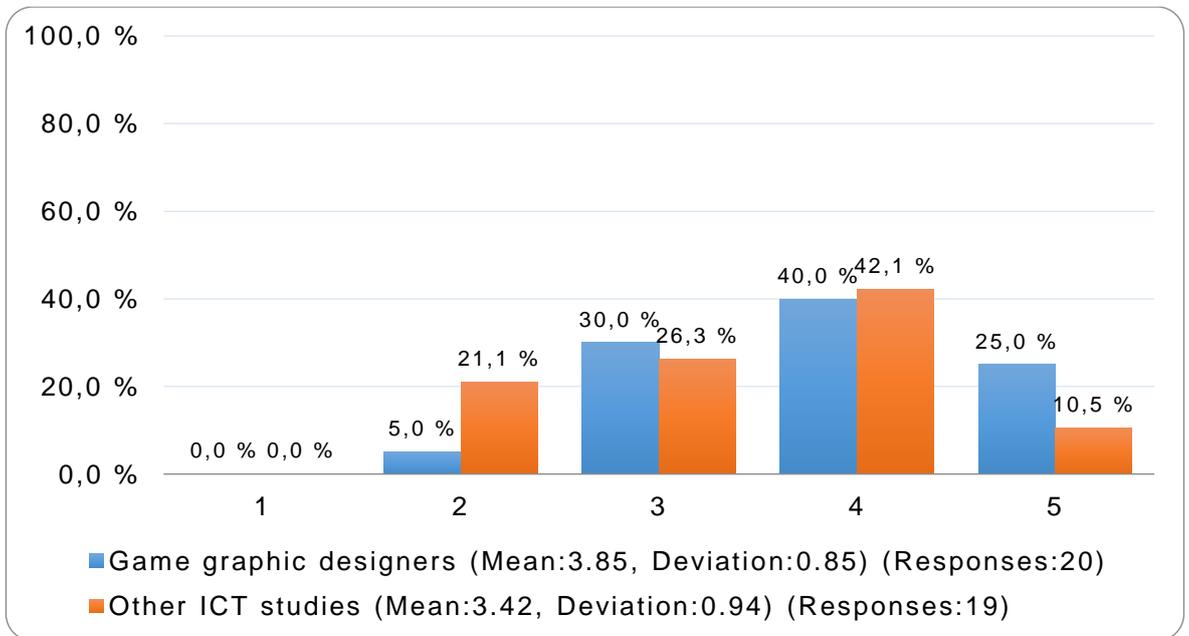


Figure 13. Social interaction (1 = Passive, 5 = Active)

Figure 13 presents the amount of social interaction the students are having during their studies. It shows that the game students are more actively involved with each other than the students in traditional projects.

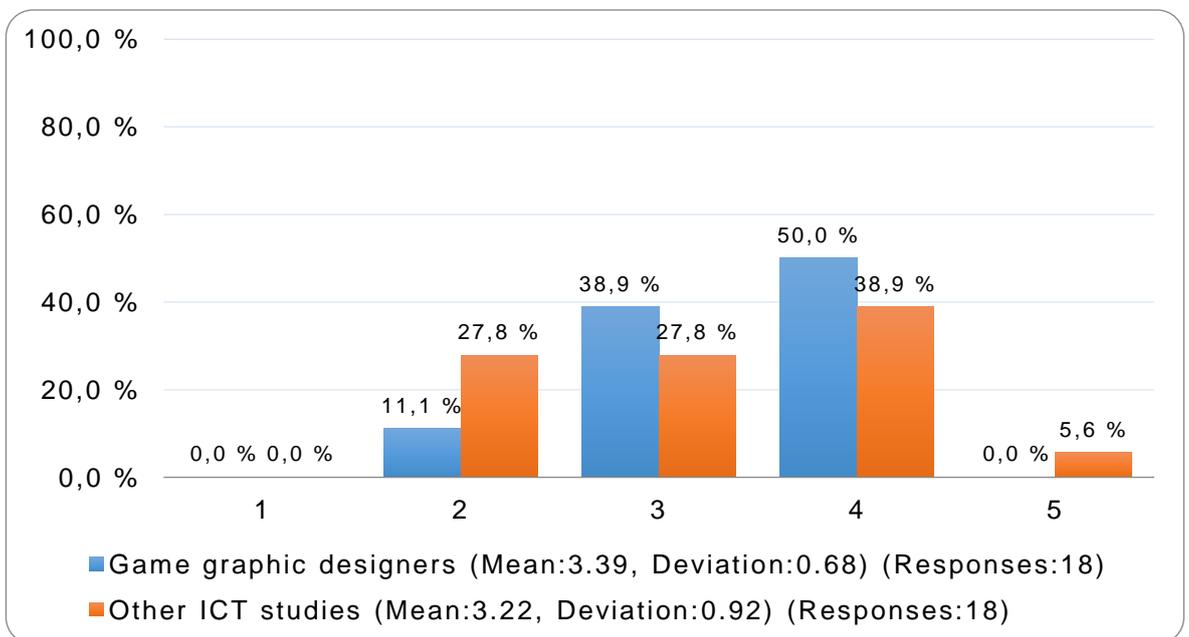


Figure 14. Study atmosphere (1 = Formal, 5 = Free)

Figure 14 presents how the students feel about study atmosphere. Formal atmosphere means a teaching session, where goals, methods, and implementation are mostly

regulated before the session and are mostly teacher oriented. The free atmosphere is more flexible and accounts the needs and wishes of the students. Students have more control in methods and may work without given instructions. The study atmosphere in agile projects averages a bit freer than the traditional projects. 50% of the game graphic designers feel that the atmosphere freer than formal while 46% of the other ICT students feel the same.

Figure 15 presents how much the studies helped the students in evaluating critically the way of their thinking and acting. The scores have a wide variation here in with the emphasis on the middle with both the agile and the traditional projects. The biggest difference is that the traditional studies have almost 95% of the answers in the middle of the scale leaving 5 % to the high end while the agile project has 79% in the middle, 5% in low and 16% in the high end.

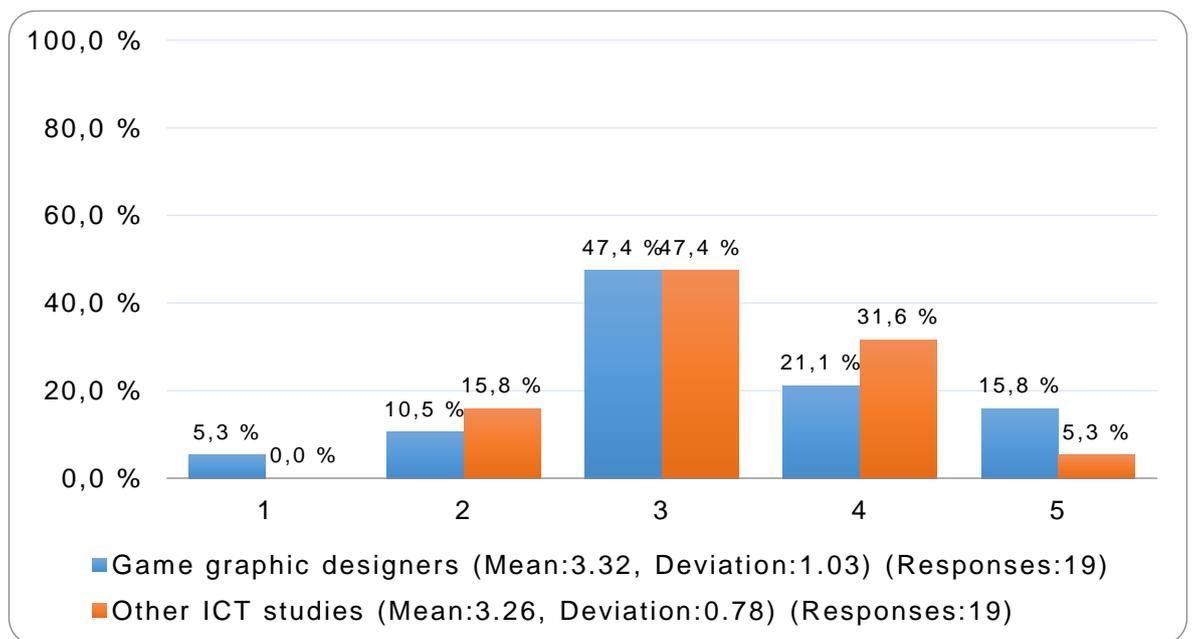


Figure 15. Reflection (1 = Little, 5 = Much)

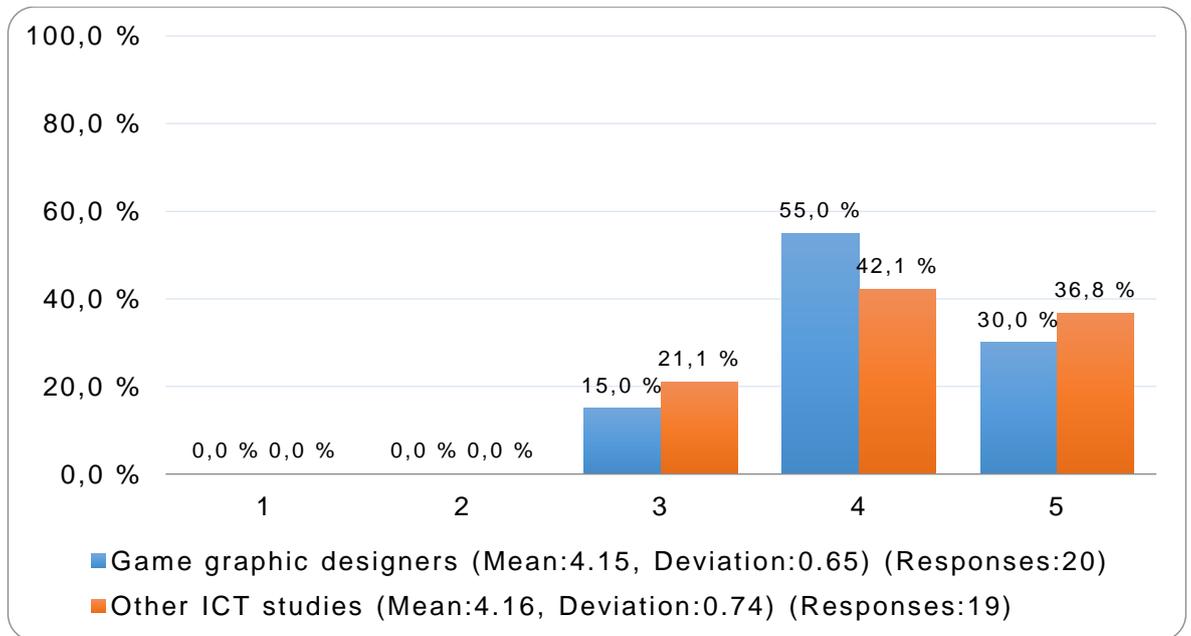


Figure 16. Importance of contents (1 = Relevant, 5 = Irrelevant)

Figure 16 presents how important the students feel that the studied content is for them. Figure 17 presents how the students feel about the approach of the studies. In a theoretical approach, the primary orientation is in the theoretical background or in general knowledge base, while in practical approach the primary orientation is in the practical side without a clear theoretical view. Figure 18 presents the amount of memorized contents in the studies and finally figure 19 presents the quality of learning. Creative learning helps to see things from a new perspective and creates new ideas, while typical learning strengthens the old ways of thinking without creating much of new ideas. These figures do not measure the method of teaching, but they give valuable information of some of the reasons behind the numbers in the other figures. The next chapter will look into the correlations between these subjects.

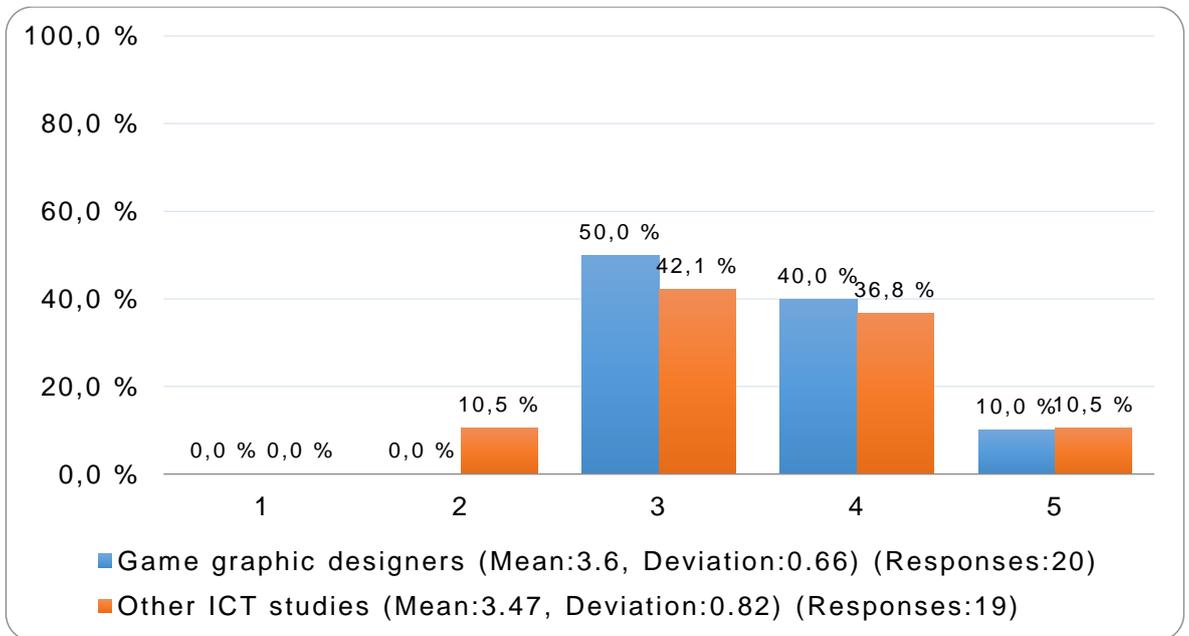


Figure 17. Approach (Theoretical, 5 = Practical)

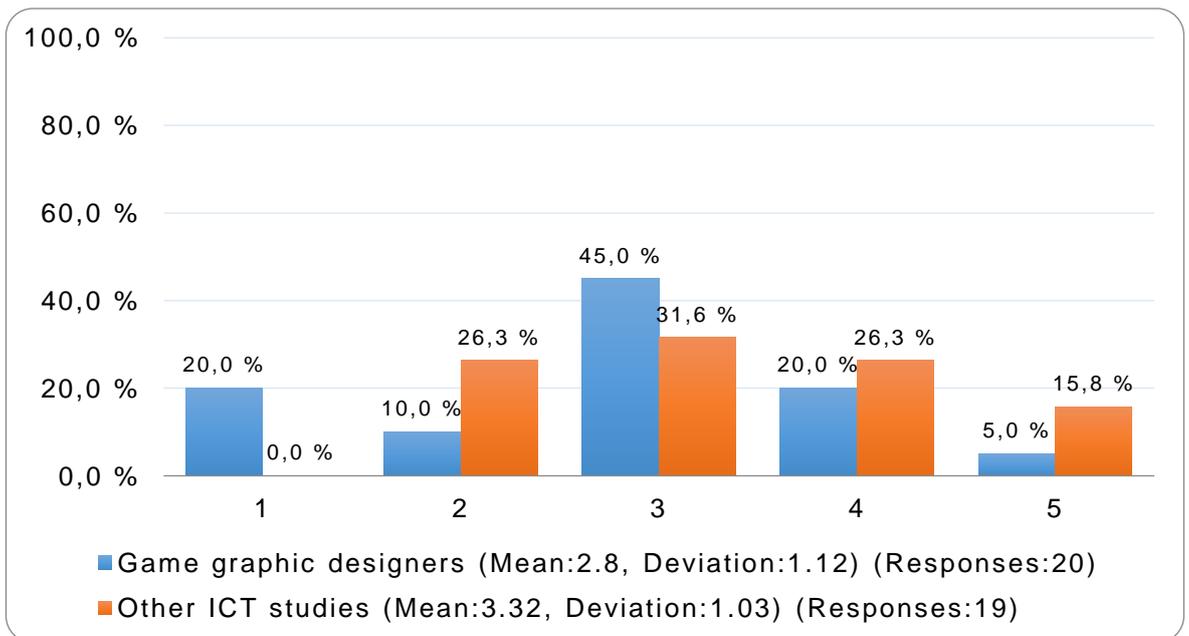


Figure 18. Memorized contents. (1 = Little, 5 = Much)

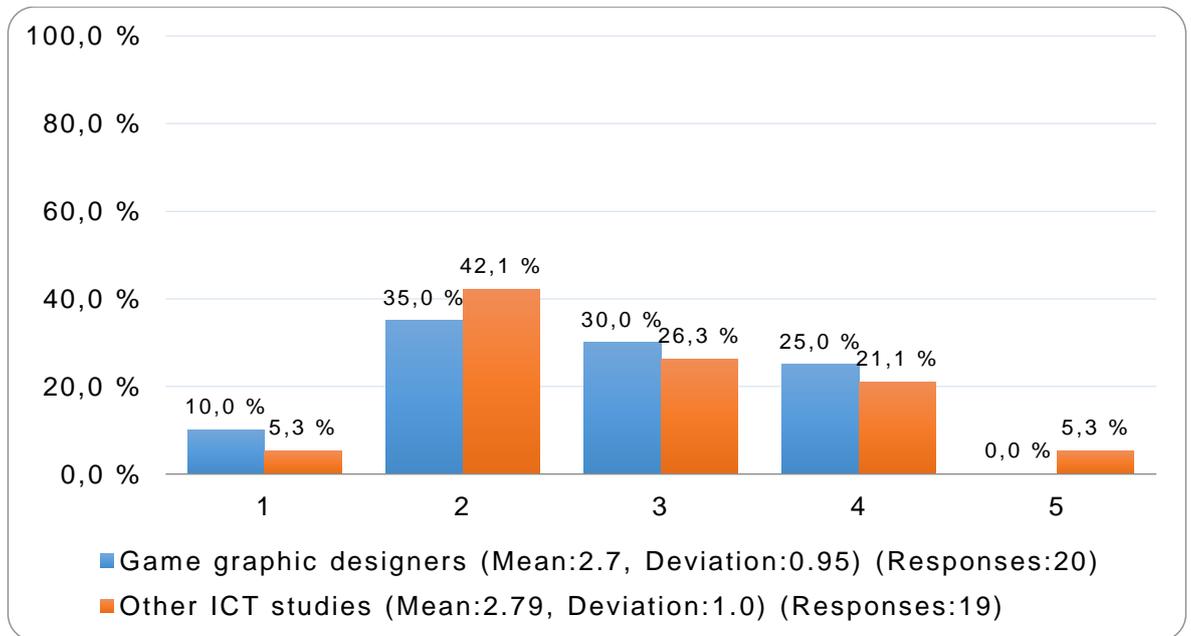


Figure 19. Quality of learning (1 = Creative, 5 = Typical)

3.5 Reasons behind happiness

To find out what reasons affect to the overall happiness of the students, I categorized the answers in three groups; negative, neutral and positive overall evaluation. These numbers include all the students in my survey. Since I had a very little really negative answer I made the categories so that answers from 1 to 3 in overall evaluation is in the negative group, 4 is the neutral group and 5 is the positive group. This was made for the sake of the research so that I can find out which answers have a correlation to each other.

Figure 20 presents the DCA information matrix. I have collected all the causalities into the matrix that I was able to decipher from the survey. Some of the causalities are only one way, which explains many of the blank squares, but also the survey was not large enough to make reliable conclusions to many of the causalities. The matrix is read from the top as following: the column topics are subjects that may have an impact on the subjects on the rows. This is not big enough material make any definite conclusions, but it gives a direction. For example, a great change may lead to a positive or neutral overall evaluation, while weak change may lead to a negative or neutral overall evaluation. In the following figures, I have looked into the strongest causalities found in the research.

CONCEPTS	1	2	3	4	5	6	7	8	9	10	11	12
	a n b	a n b	a n b	a n b	a n b	a n b	a n b	a n b	a n b	a n b	a n b	a n b
1.Student's												
1a Independent	a											
1n Neutral	n											
1b Controlled	b											
2.Engagement												
2a Weak	a b	a	a b		a	a	n					
2n Neutral	a n b	n	n b		n	b	n b					
2b Strong	a n	b	b		b	b	b					
3.Motivation												
3a Weak		a	a	a	a							
3n Neutral		n b	n	n	n	n	n b			n b		
3b Strong		b	b	b	b	b	b			a		
4.Importance												
4a Irrelevant				a								
4n Neutral				n								
4b Relevant				b								
5.Social												
5a Passive			a		a	a n						
5n Neutral			n		n	n b						
5b Active			b		b	n b						
6.Study												
6a Formal						a						
6n Neutral						n						
6b Free						b						
7.Approach												
7a Theoretical							a					
7n Neutral							n					
7b Practical							b					
8.Reflection												
8a Little		a						a				
8n Neutral		n b	n b					n				
8b Much		n	b					b				
9.Memorized												
9a Little									a			
9n Neutral									n			
9b Much									b			
10.Quality												
10a Creative										a		
10n Neutral										n		
10b Typical										b		
11.Change												
11a Small		a n		a n b	a n b	a n	n	a n			a	
11n Neutral		b	b					n			n	
11b Great		b	b	b	n b	n b	n b	b			b	
12.Overall												
12a Negative	n b	a	a	a	a n	n	n				a	a
12n Neutral	n	n b	a n b	n b	n b	b	n b				a	b n
12b Positive	a b	b	b	n b	a n b	b	n b				b	b

Figure 20. DCA information matrix

Figure 21 presents how the student's role affects to the overall evaluation. There seems to be a correlation that the students that feel more independent are also happier in the studies, but then there also is a 9 % group of very happy students that feel, they are in a very controlled environment.

Figures 22 and 23 show that there is a clear correlation between the engagement and motivation of the students to the overall evaluation. I also made a cross-reference between high motivation and engagement. Since most of the students have very high motivation I made two groups – one with students of very high motivation and other with all the rest named as ‘normal motivation’. The correlation between motivation and engagement is very clear although there are some students that feel they are very highly motivated, but still not very engaged. This is presented in figure 2

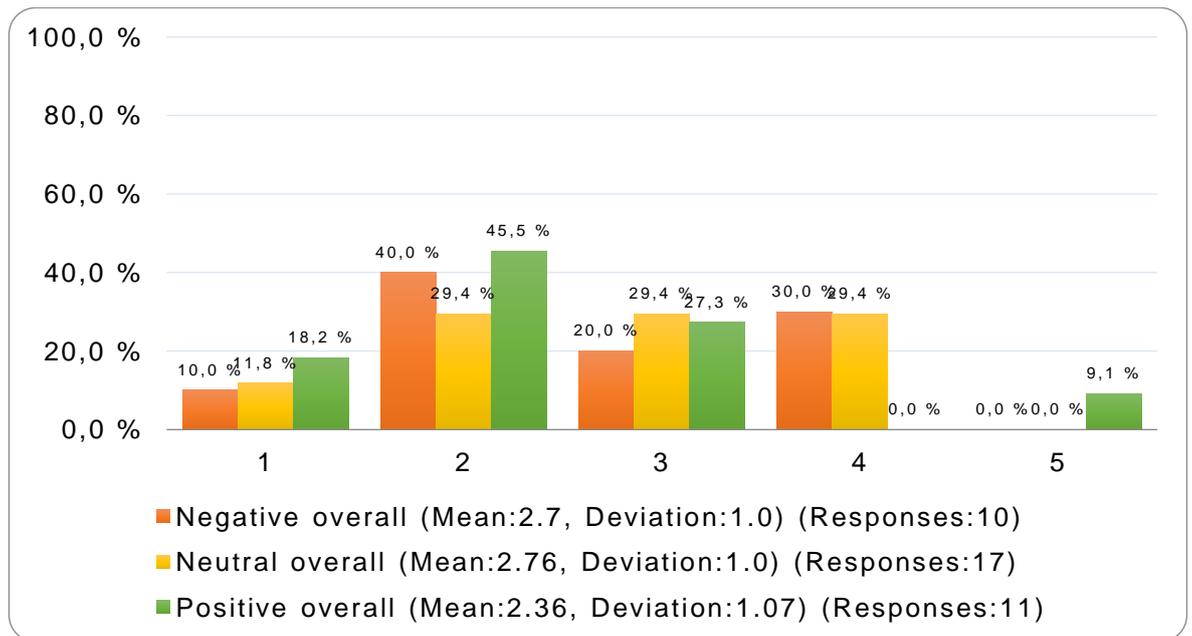


Figure 21. Student’s role (1 = Independent, 5 = Controlled)

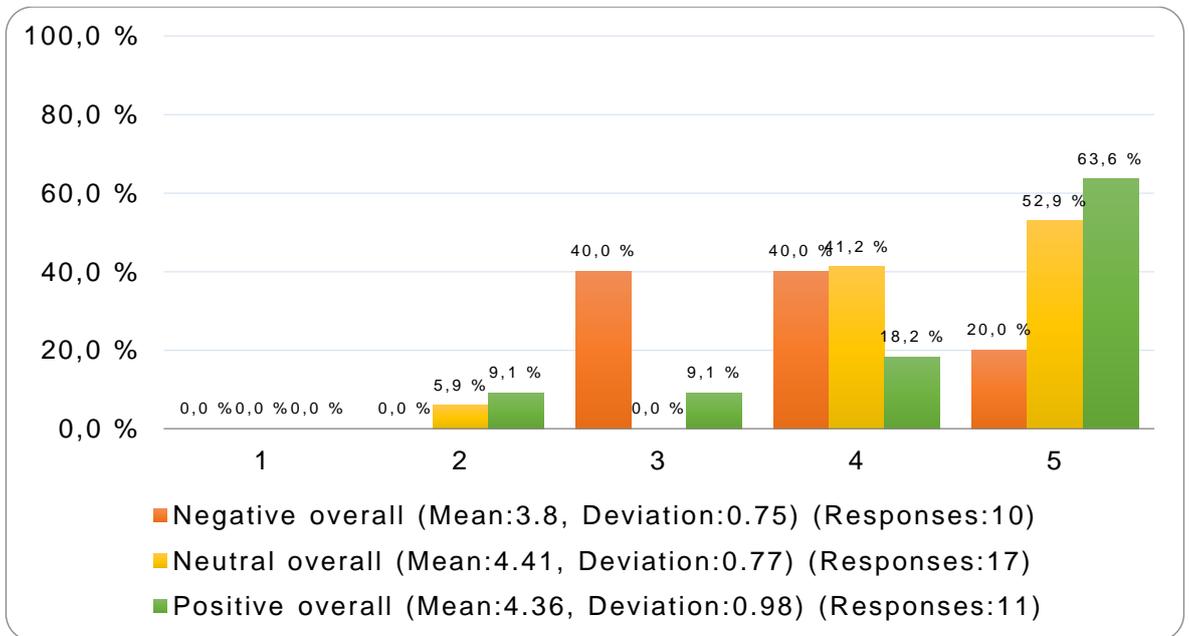


Figure 22. Engagement (1 = Weak, 5 = Strong)

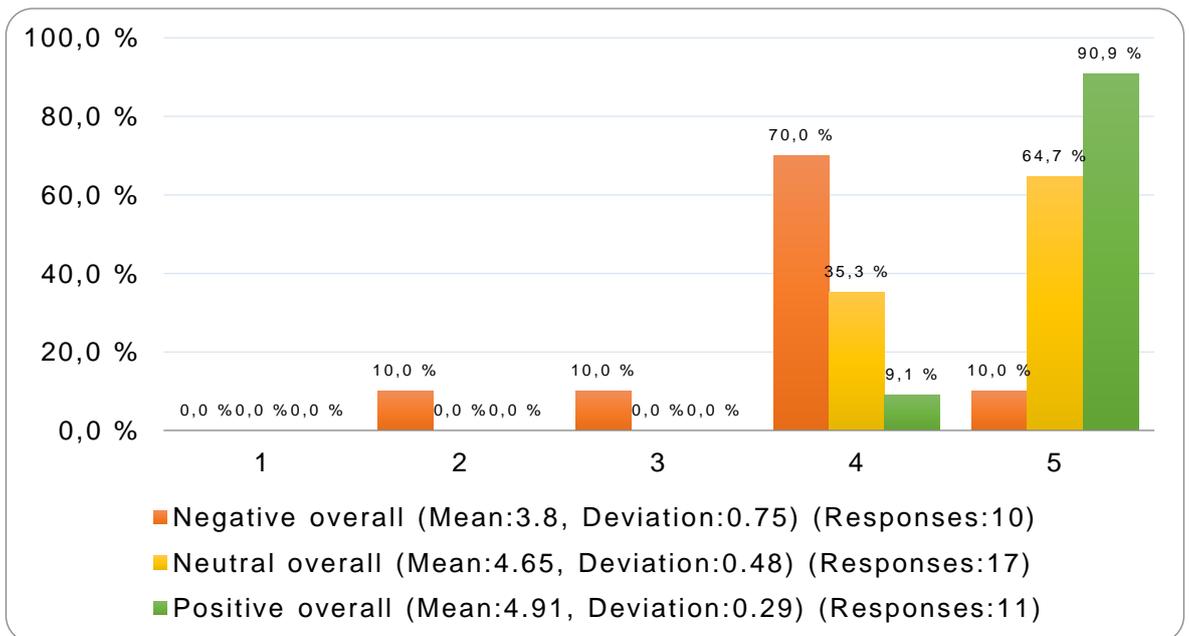


Figure 23. Motivation (1 = Weak, 5 = Strong)

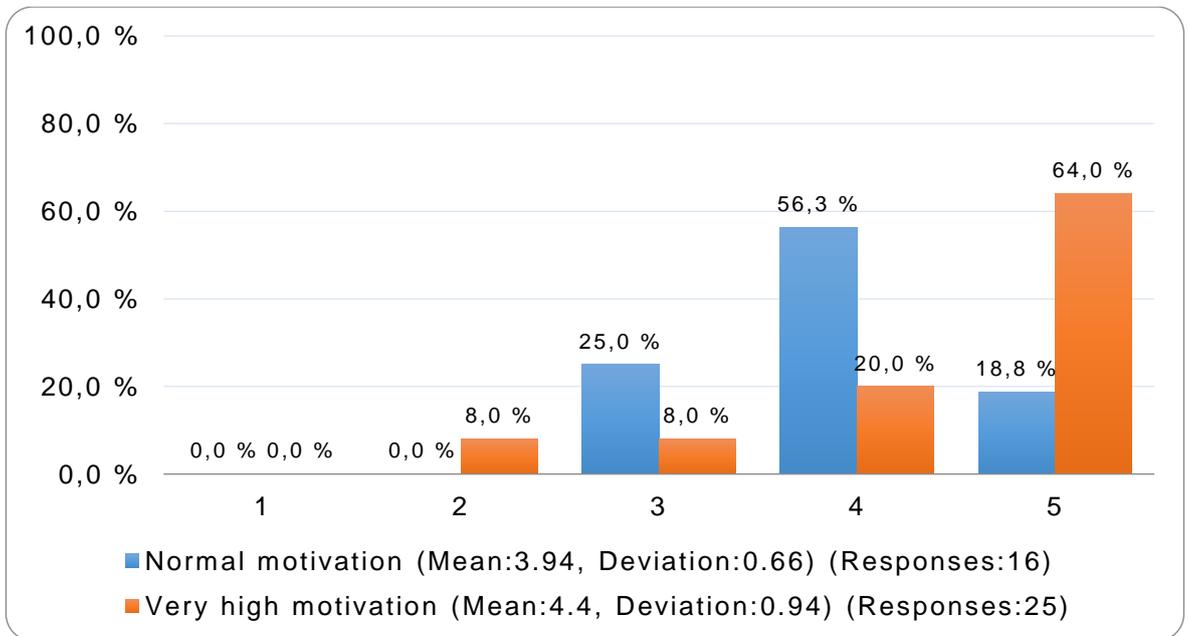


Figure 24. Engagement of very highly and normally motivated students (1 = Weak, 5 = Strong)

The importance of contents is another subject that correlates with positive overall evaluation. Figure 25 presents that 90% of the students in the positive group have scored 4 or 5 in the importance of contents while most of those that feel that contents have only a little or medium relevance to them have given a negative overall evaluation.

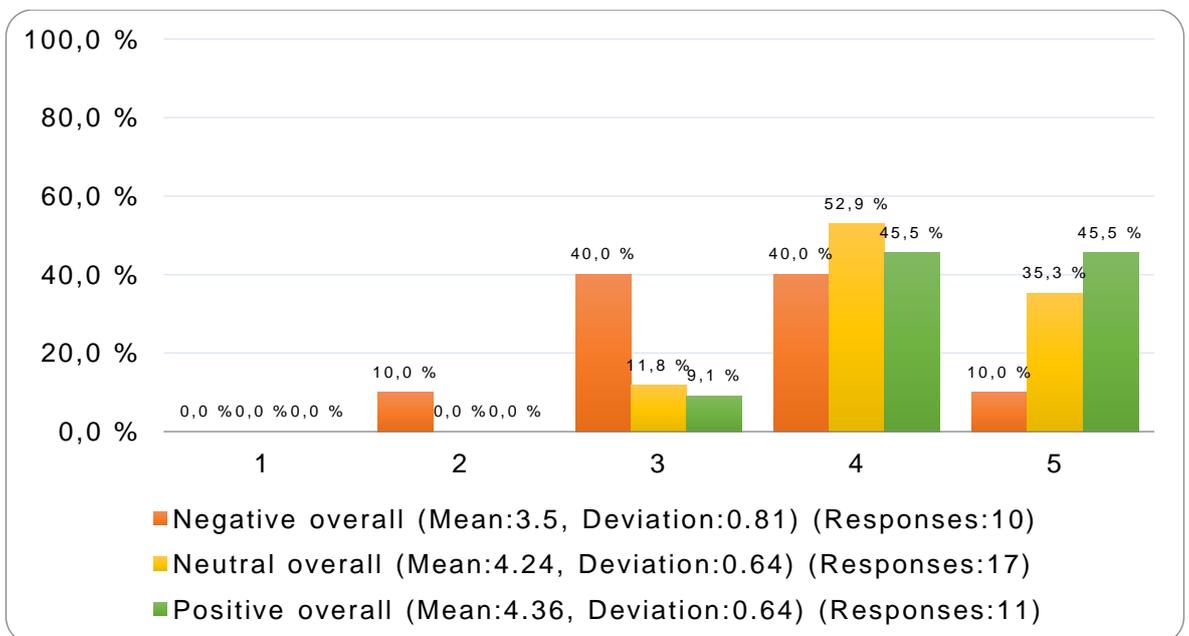


Figure 25. Importance of contents compared to overall evaluation (1 = Irrelevant, 5 = Relevant)

Figure 26 presents the impact of social interaction on the overall evaluation. There seems to be some correlation as well so that more socially active students have given better overall evaluation scores, but the correlation is not perfect. Quite a big part of the positive group does not feel very socially active. As seen in figure 27, half of the other ICT students that have given very high overall evaluation do not feel particularly socially active, while 80% of the graphic designers that have given positive overall evaluation do.

Figure 28 presents how study atmosphere impacts to overall evaluation. Once again there seems to be a very clear correlation. 60% of the positive group and 56% of the neutral group think they are studying in a free atmosphere while only 20% of the negative group feels the same. On the other end there not much difference between the groups feeling they are in the formal environment. Most of the negatives are in the middle.

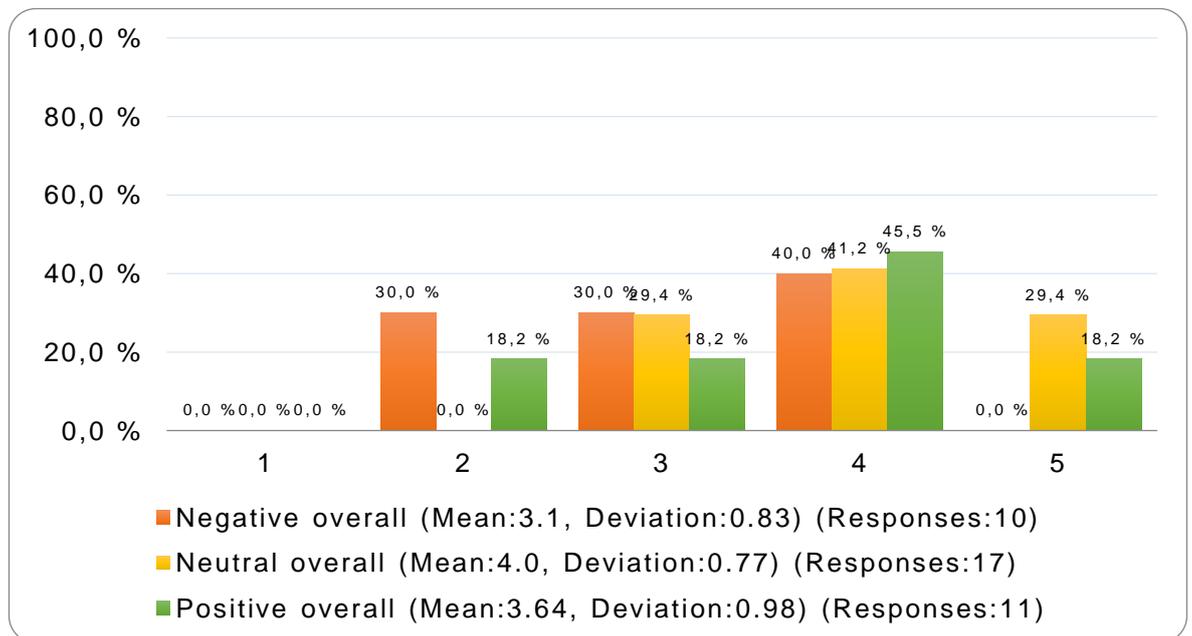


Figure 26. Social interaction compared to overall evaluation (1 = Passive, 5 = Active)

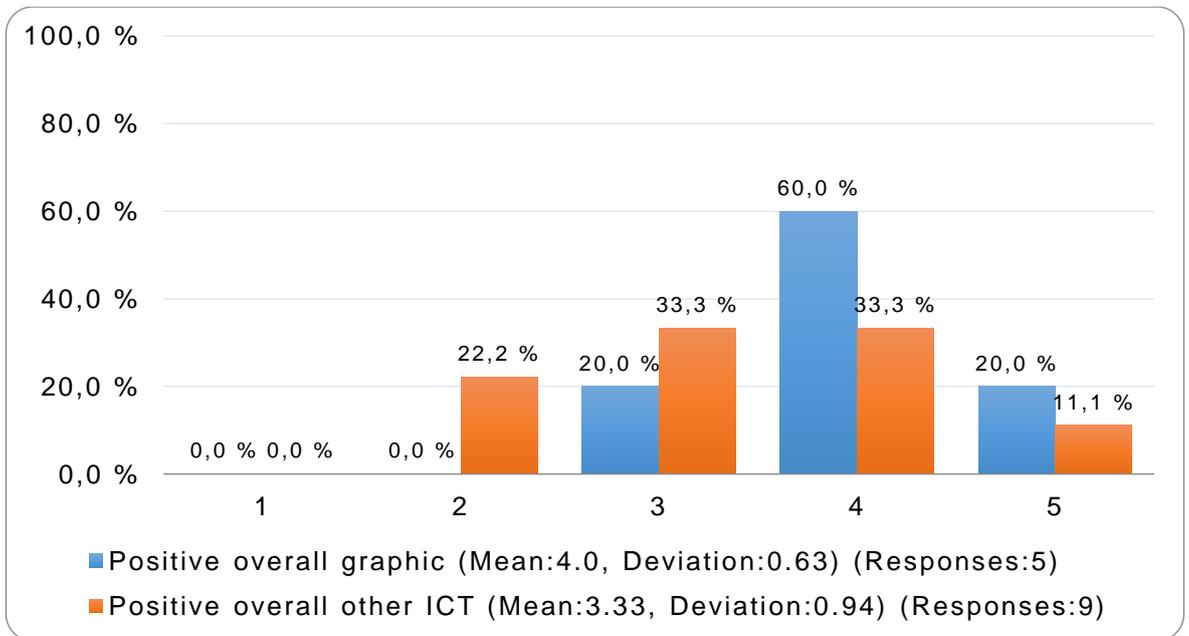


Figure 27. Social interaction of positive graphic designers and positive other ICT students (1 = Passive, 5 = Active)

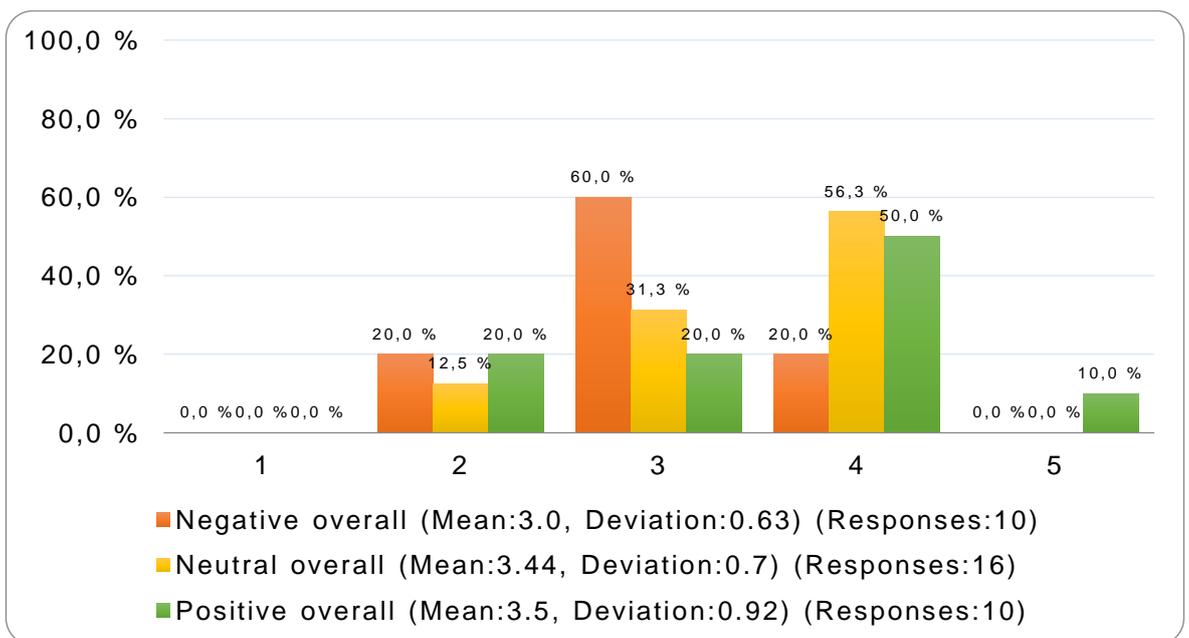


Figure 28. Study atmosphere compare to overall evaluation (1 = Formal, 2 = Free)

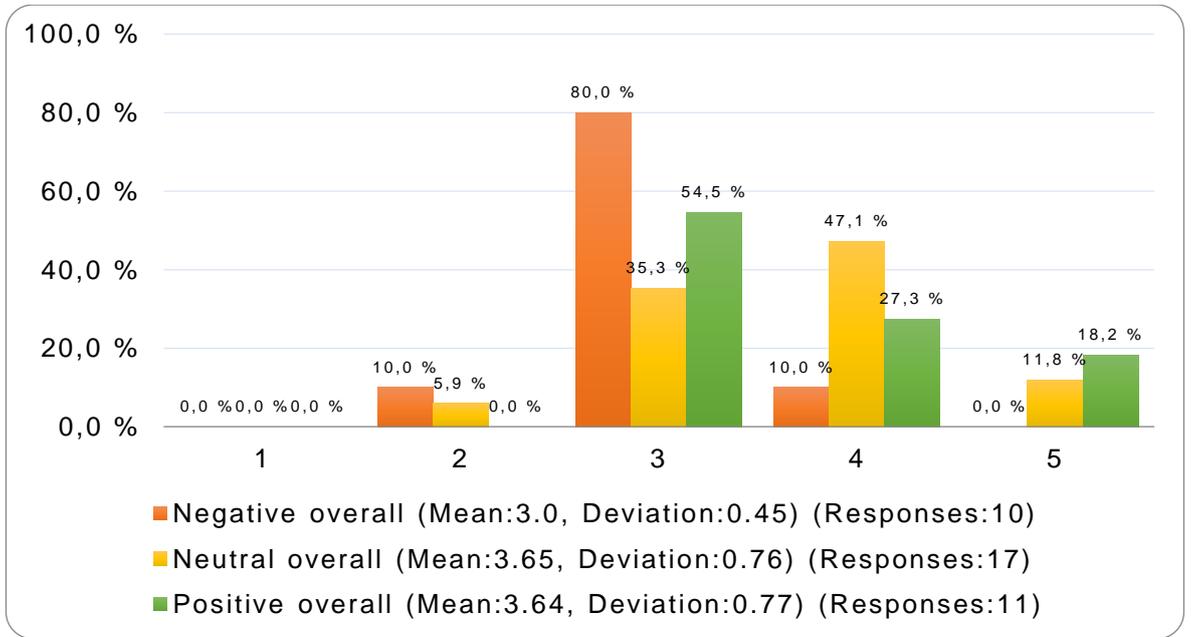


Figure 29. Approach compared to overall evaluation (1 = Theoretical, 2 = Practical)

Figure 29 presents that practical approach to the studies also has a correlation to positive overall evaluation. On the other, the amount of memorized content does not seem to have an impact on the overall evaluation as seen in figure 30.

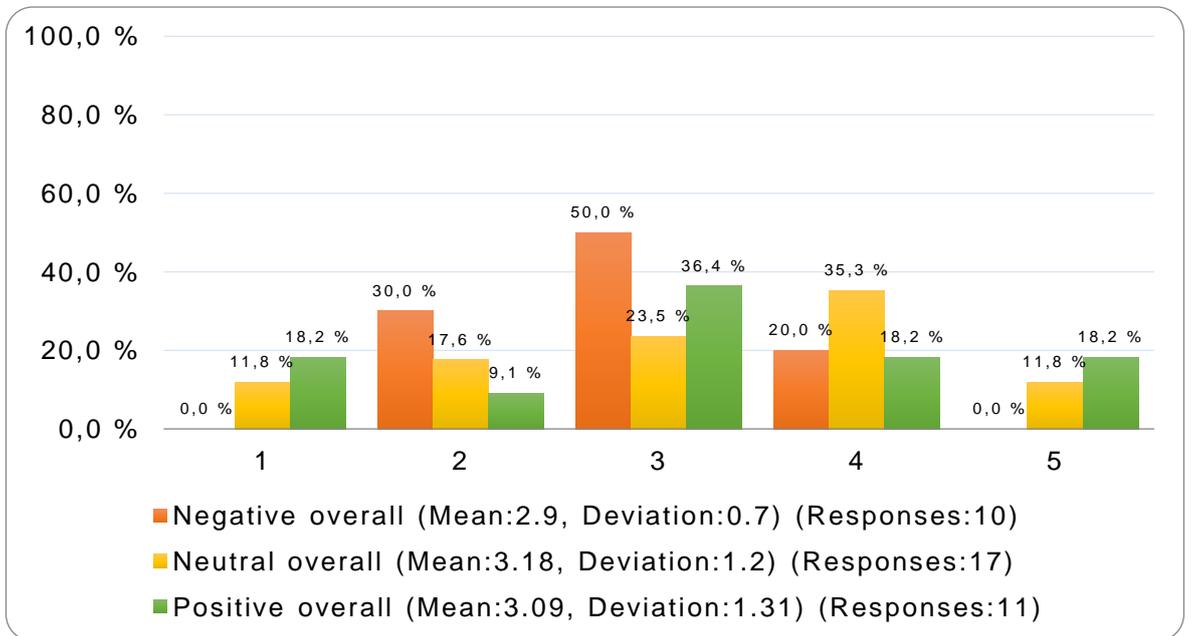


Figure 30. The amount of memorized content compared to overall evaluation (1 = Little, 5 = Much)

Figure 31 presents the correlation between change and overall evaluation, which seems to be very high. The change seems to be the biggest benefactor to the overall evaluation. Figures 32 and 33 present how motivation and engagement impact to the change. Not so surprisingly there seems to be a correlation between both, but engagement has a much stronger correlation than motivation. Yet, many students that do not feel strong engagement to the studies, seem to feel achieving a great change. In addition to motivation and engagement, another subject that seems to have a strong correlation with change is the reflection. As seen in figure 34, every student that feel like achieving great change have scored at least three in reflection, while half of the students that feel like achieving small change have scored one or two in reflection. Figure 35 presents how engagement impacts on reflection. It seems clear that weak engagement leads to weak reflection as well.

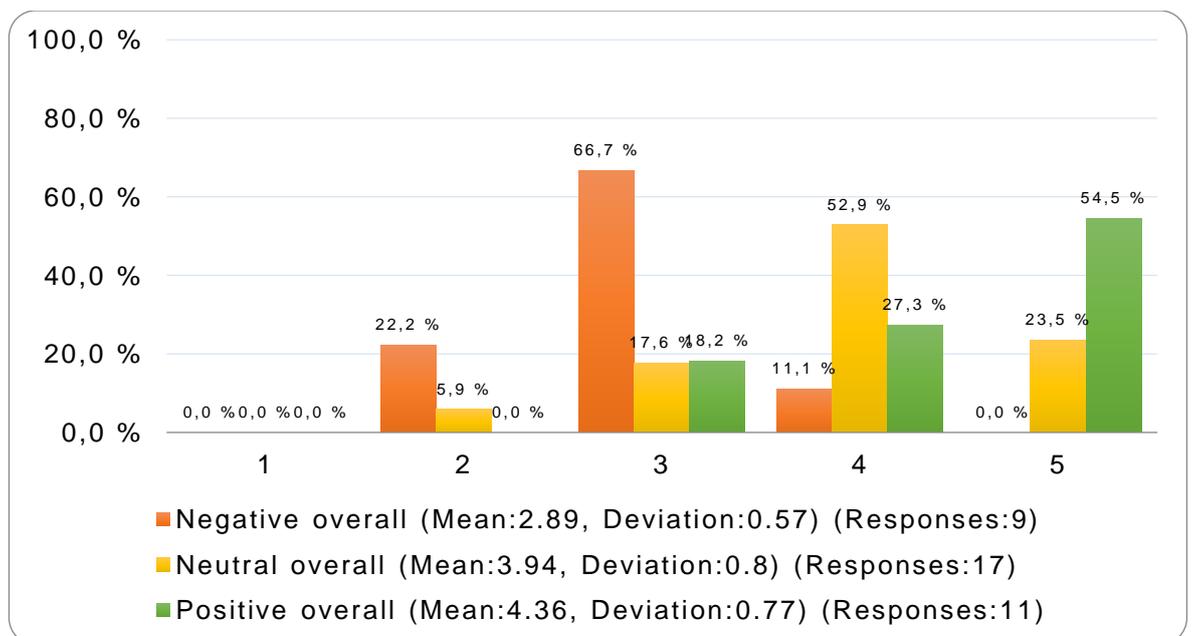


Figure 31. Change compared to overall evaluation. (1 = Small, 5 = Great)

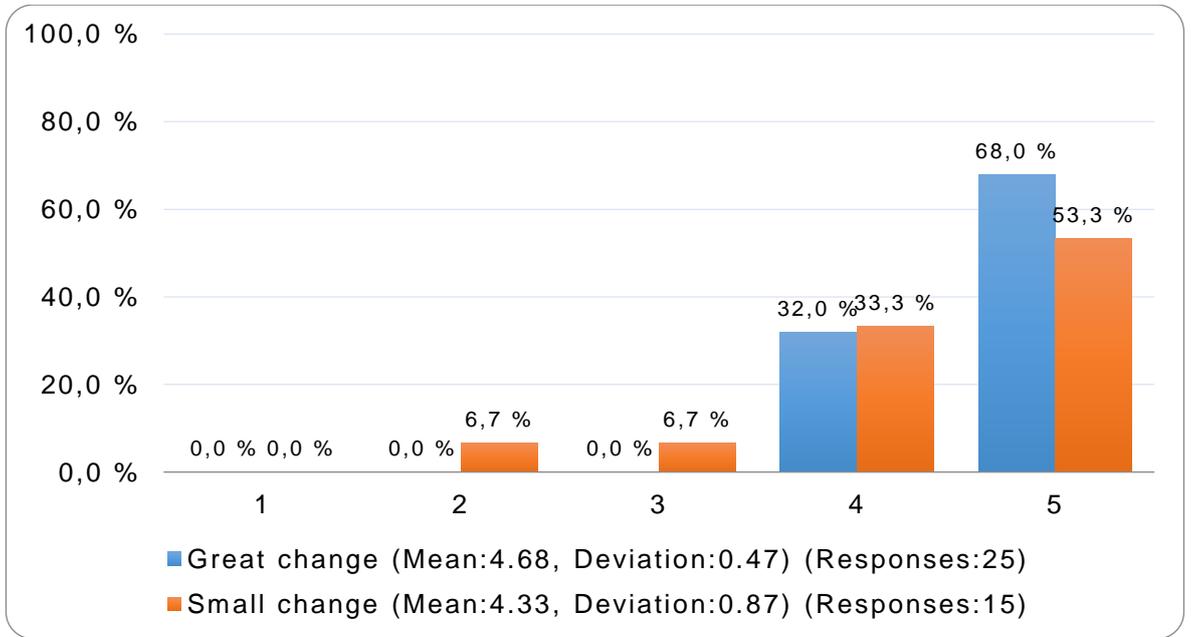


Figure 32. Motivation compared to change. (1 = Weak, 5 = Strong)

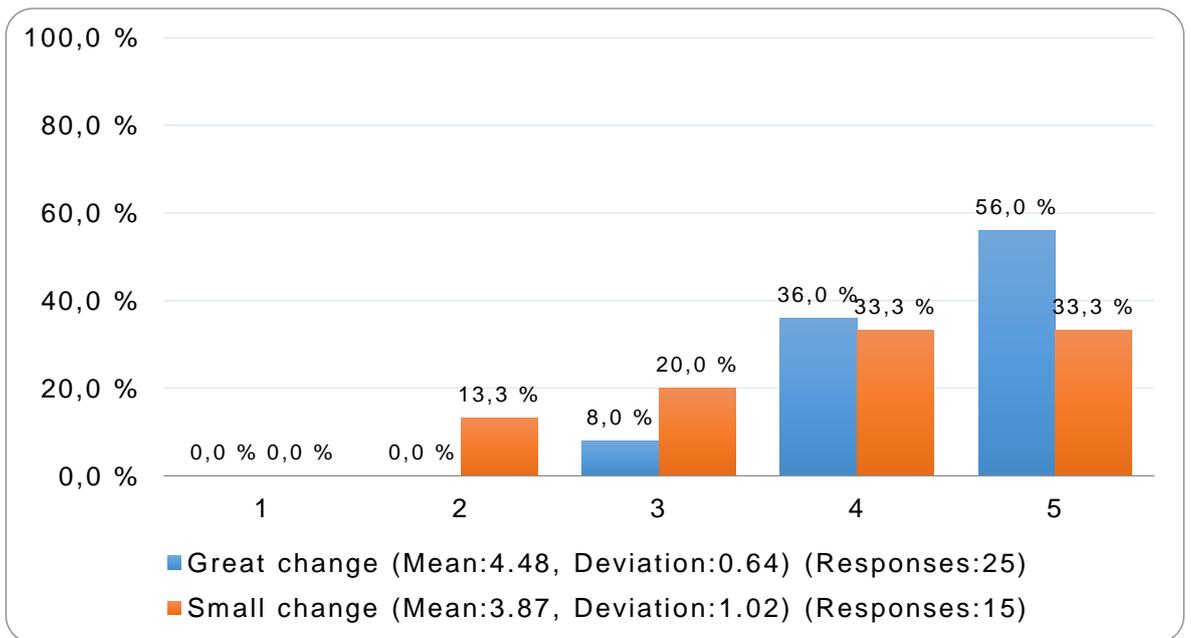


Figure 33. Engagement compare to change. (1 = Weak, 5 = Strong)

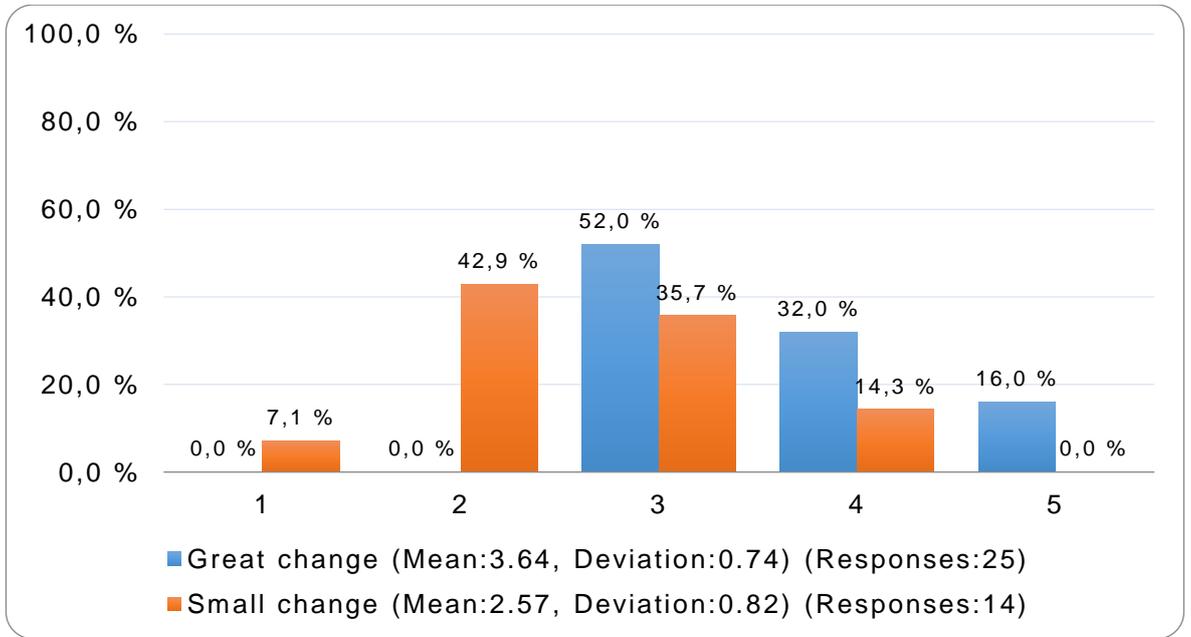


Figure 34. Reflection compare to change. (1 = Little, 5 = Much)

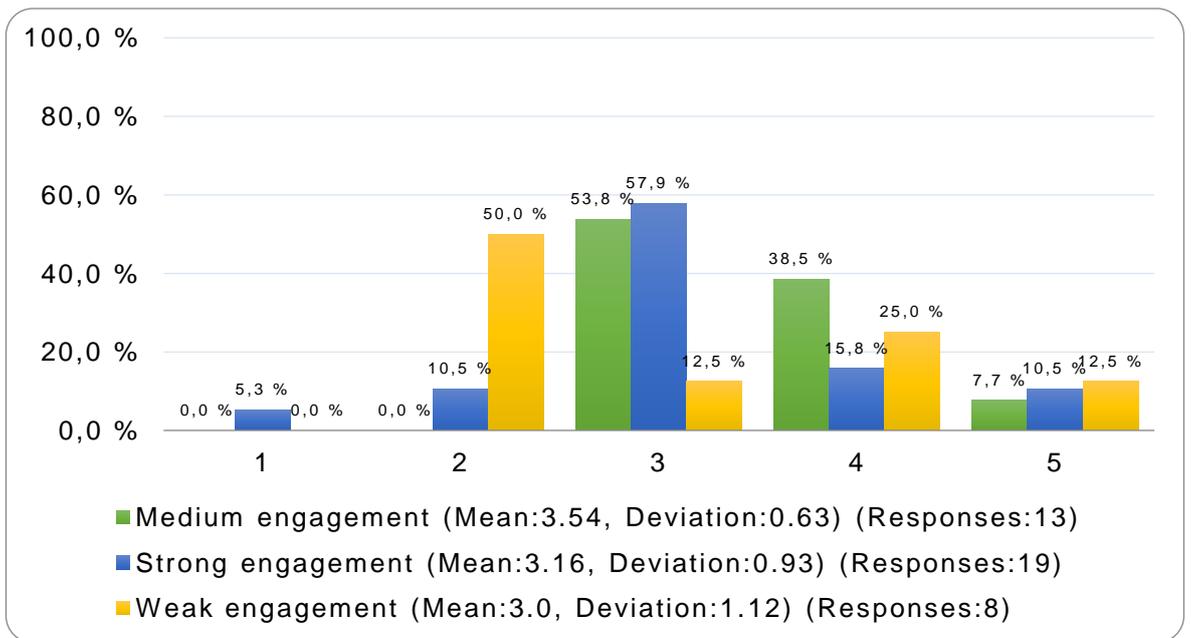


Figure 35. Reflection compared to engagement. (1 = Little, 5 = Much)

4 Conclusion

This chapter contains the analysis of the results and answers to the last research question: How can we improve our model of agile education? There are also some ideas for future research presented.

4.1 Analysis of the research results

Going through the financial data it seems that our agile model is very competitive compared to the traditional classroom training. In the first adaptation we managed to make a decent operating profit and after that, it has only improved since we have been able to cut the development costs. Most of the expenses are generated from the personnel costs and in our model it means guidance from the teachers, but a lot of it also comes from planning the structure of the education and generating material for the students. Since most of the material is web based and we lean heavily on online courses to teach the theoretical material, the courses can be more easily scaled for a bigger audience than traditional classroom training. Thus, our model enables a bigger number of students with less cost than a training model that is mostly held in a classroom. On the other hand, it will get more expensive if the number of students gets smaller since it will take more time to keep the online material up to date than to prepare for classroom training.

The number of students dropping out of the course has been quite high compared to our other courses. Also, the number of graduated students is low at the moment, but it seems that we can fix this number to be quite decent. This means that the graduation times are longer than in our classroom-based studies, but the number of graduates should be better than in other courses that need real work projects. Lab based simulations are far more efficient to get the degrees, but the educational system as whole is moving away from those and the government aims to move all demonstrations into real work. One of our main goals in this project from the beginning has been creating a motivating study environment. Looking in these drop out numbers it seems that we have not succeeded in this goal. Still, I don't think that we have completely failed in this goal, and it would be interesting to conduct another

research on reasons behind dropping out of school. The motivation of the students that have answered to the survey seems to be extremely high and maybe it is a good thing that student's with low motivation drop out rather soon. One thing that was not included in the survey is, how hard the students need to work.

The research shows that overall our students are very happy with the education that Amiedu ICT provides. The overall evaluation is a slightly lower for our agile projects than it is for more traditional projects. We have succeeded in creating a free atmosphere with active social interaction in our courses, but we are still struggling with the efficiency of the project. The biggest factor to overall evaluation seems to be the change that the students achieve from their education. The students in more traditional courses seem to feel that they get more out of the courses than our agile students. Reasons behind the smaller level of change seem to be in the level of engagement and reflection. Both of which are core values in our system.

Overall it seems that our model has the right values and we are aiming to do the right things, but we need to find a way to improve the engagement level of the students, which leads to better social interaction, better reflection, and greater change. The things that seem to improve the level of engagement are the independent student's role, high motivation, active social interaction, free study atmosphere, and practical approach. So these are the areas that we need to improve to make our model excellent.

4.2 Ideas for future research

This one-time survey of a relatively small number of students is not big enough material to draw concrete conclusions about how our model works. For future research it would be good to make a little simpler survey that asks for things like how motivating the education is rather than asking how motivated the students are since there are a number of things that affect to the motivation of the student that we as teachers are not able to change. One thing missing from this survey was the amount of work required from the students. Including it might give interesting correlations. The survey should be conducted to every group of students two or three times during their

education project, so that we would see if the students in the beginning of their studies feel differently of those that are finishing it.

Another important thing to research is the reasons behind dropping out. For this research, I have only numbers, but when I talk to the students that are dropping out, the reasons are normally from their personal life rather than from the studies. It would be important to know the reasons and find out if there are anything the school can do about it.

Finally, I would like to know how the students feel about these studies sometime after graduation. It would be interesting to conduct a research for the students that have been out of school for a year or two and ask how they feel about the education project then. Especially how they feel about the change that the project brought to their lives. I think that this feeling of change would be more important to know than the one, they feel when they are still studying.

References

Amiedu. 2016 Projektien tulokset

Auer, A., Auer, L., Heinäsmäki, M., Hölttä, J., Kalliala, E., Laanti, M., Laine, K., Lekman, L., Miinalainen, P., Naski, H., Piiharinen, T., Puhakka, H., Pyhäjärvi, M., Pääkkönen, T., Räisänen, S., Sora, H., Taipale, M., Talvio, J., Tanninen, A., Toikkanen, T., Toivola, T., Toro, K., Valsta, A., Väyrynen, V. & von Weissenber, M. 2013. Finn Lectura

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. 2001. Manifesto for Agile Software Development. URL: <http://agilemanifesto.org/> . Accessed 13 Sep 2016.

Delhij, A., van Solingen, R. & Wijnands, W. 2015. The eduScrum Guide. URL: http://eduscrum.nl/en/file/CKFiles/The_eduScrum_Guide_EN_1.2.pdf . Accessed 18 Sep 2016. eduScrum Team

eduScrum Ashram College. URL: <http://eduscrum.nl/en/fotogalerij/album/2> . Accessed 1.10.2016

Kontiainen, S. 2002. Dynamic Concept Analysis (DCA). University of Helsinki Department of Education. Helsinki.

Oxford Dictionaries. 2016. Definition of agile in English. URL: http://www.oxforddictionaries.com/us/definition/american_english/agile . Accessed 13 Sep 2016. Oxford University Press.

Pichler, R. 2010. Agile Product Management with Scrum. Pearson Education Inc. Boston.

Repo-Kaarento, S. 2007. Innostu Ryhmästä – Miten ohjata oppivaa yhteisöä. Dark Oy. Vantaa

Schwaber, K. & Sutherland, J. 2016. The Scrum Guide. URL:
<http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf> .
Accessed 17 Sep 2016. Scrum.Org

Screenmedia. What is Agile Development? An Introduction. URL:
<http://www.screenmedia.co.uk/blog/2014/08/what-is-agile-development-a-brief-introduction/> . Accessed 1.10.2016

Waters, K. 2007. What Is Agile? (10 Key Principles of Agile). URL:
<http://www.allaboutagile.com/what-is-agile-10-key-principles/> Accessed 13 Sep 2016.

Wikipedia. Scrum. URL:
[https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)) / . Accessed 1.10.2016.