An agile approach to teaching and learning

A handbook for teachers to encourage student participation in secondary schools

Valborg Sturludóttir

May 2019

Contents

1 Introduction

This project attempts to improve teaching by combining it with practices from software engineering. Student participation and collaboration are major themes in this handbook and that's what sets the method apart from other traditional methods of teaching. By involving the students in how the learning takes place and encouraging them to participate in a community of learning the aim is to empower them, make them aware of their own efforts and capabilities.

This handbook describes a method for the teacher. It is a product of a masters program at the University of Iceland. The program is called "The education of secondary school teachers" and is an interdisciplinary degree combining educational science and computer science. Argumentation and further discussion can be found in the accompanying thesis along with citations and references. The handbook is divided into five chapters: chapter 1 an introduction, chapter 2 a summary of agile methodology in a software engineering context, chapter 3 why agile methodology may be used in an educational setting, chapter 4 the description of the agile educational method, and chapter 5 resources. The second chapter an introduction to what agility means in a software engineering context. An overview of agile processes is given with pertinent explanations of key concepts, those key concepts will then be superimposed on education with the method. The third chapter's focus is what agility means in an educational context and how a software engineering methodology might enrich teaching. The fourth chapter then, constituting the main focus or backbone of this handbook, describes the method and how it should be applied, what it takes to create a course with team work, and how to value student input on course development. Further reading and resources are listed and laid out in chapter five.

There are many ways to combine computer science with teaching, such as creating educational software, but the agile methodology is of particular interest to the author. Major influences on the computer science side comes from working in software development under agile principles, and teaching courses in software development where students chose to adhere to agile practices. Major influences on the teaching side include social constructivism and the approach that students come with their preconceived notions, thoughts and ideas into the classroom, and that there is a need for conversation to make sure that the student has learned what was intended.

This handbook is meant for teachers who want to incorporate student feedback in the lesson plan and assessment. For teachers who want to engage students in a novel way and give them an active role in their education. The aim is to help teachers create an atmosphere of learning, and effectively assess progress by utilizing a method that values the input of students.

Key features are continuous communication, responsiveness and collaboration.

2 What is the Agile Methodology?

At its core agile methods are about creating software through teamwork in an iterative process, where the customer, and communication with the customer, is at the center. See figure 1 to get a better feel to the iterative nature of agile methodology. At first the requirements, the expectations of the system in concrete terms, are mapped out, and broken down. These requirements are then taken into consideration and implemented in a logical order as deemed desirable or necessary. Implementation means planning, designing, building, testing and reviewing it before launching it. After launch it might be apparent (or before) that a particular requirement that was to be implemented is not met. So it is taken into the next iteration of planning to launching.

Organizations that develop software in this manner usually have overall management in accordance with agile practices, utilizing defined roles. When developing software it is imperative to have software developers, and those are at the center of agile practices. By focusing on the development team and maintaining a continuous conversation with the customer agile practices attempt to make the people who are creating value a focal point. This focus, and the fact that adaptability and responsiveness are key features, sets agile methodologies apart from other software development practices.

The different roles and actors differ between versions of agile methods and the main focus in this introduction will be on the development team, the product owner and the customer, and omit other roles of significance in some agile methods, such as the role of an agile coach or Scrum master. That is not to say that they are redundant in other situations or contexts. There are other roles, but for the sake of simplicity and relevance the following three *the team*, *the product owner* & *the customer* will be at the center of our agile approach to the planning, designing, building, testing, and reviewing of teaching and learning.

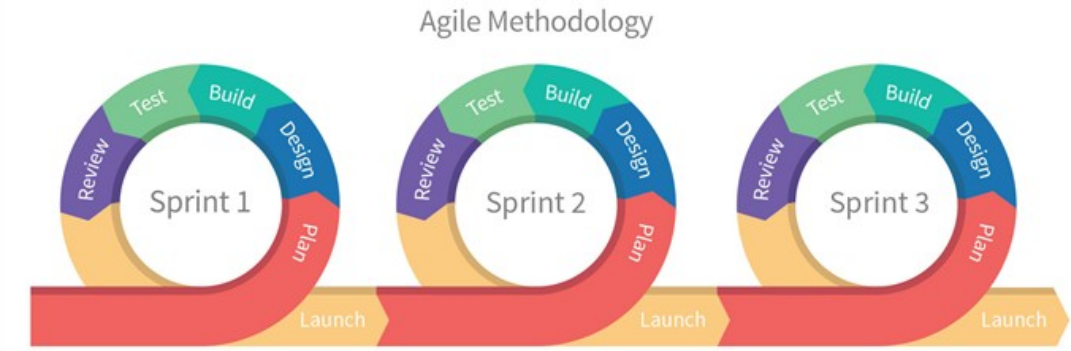

Figure 1: The usual workflow of agile software development is to always work in increments, iteratively. The different roles play different parts in each step of the process. Planning is where the customers wishes are prioritized and the product owner sets the focus. In the design, build and test steps work on the project is in the hands of the developers under the guidance of the product owner, and at the end the result is reviewed before being launched. It depends on the level of involvement of the customer where their feedback comes in, usually at launch of the product so their priorities and input is then the substance for the next planning phase [1].

The developer-team self-organizes and delegates work according to capability. The work is divided into blocks, usually called *sprints*, and a sustainable pace is set to create what is asked for, no more or less. The person organizing these sprints is the *product owner*. They choose what functionality goes into each sprint and what its focus is. They have the pivotal role of overseeing the developmental process, and by working closely with the development team they are more likely to adhere to democratic ways. This role is an involved one and vital when transferring the agile methodology to education. The teams reflect on their work, both to improve the work done and also themselves and the teamwork. This reflection leads to reiterating, refactoring or continuing the software. It also leads to personal growth, because the wellbeing of developers is crucial, as will be explained in more detail later on, as we introduce the manifesto of agile development (see principles 5, 8, 11, and 12 in figure 2).

The customers are the key stakeholders, being the people for whom the software is being created. They might not know the details of what they want right at the start, they might have only a vague sense or a rough idea. By developing the software incrementally and getting input from the customer throughout the process a clearer picture of the vision presents itself. By being informed and made aware of that the process allows for a change of plans or requirements the customer is empowered.

[1] Image credit http://newproductvisions.com/blog/wp-content/uploads/2017/12/agile.jpg

The different roles in an software engineering setting:

- **The development team** consists of the essential people who are creating the software. The team is self-organizing such that each person gets work suitable for their pace and skillset. Some agile flavors require a customer representative be on the team.

- **The product owner** is a person responsible for the product. They decide what the focus of the sprint should be, maintain the backlog and choose which items from the backlog go into the sprint.

- **The customer** is the reason the work is being done. According to agile philosophy, their voice is the most important one and continuous conversations need to be maintained in order to satisfy their needs. The conversation here is about getting information¸ about requirements and priorities from the customer.

Agile is a name given to a group of different software development methods that all align with the agile manifesto as it was written in 2001 by 17 software practitioners, see figure 2. The four propositions of the manifesto might seem simple or even a tautology at first but it turns out that there are many different ways to interpret them. The 12 principles listed in the manifesto go into detail on how the propositions should be implemented. When looking further into how agile methods are implemented many different flavors emerge. What differentiates these flavors (for example Scrum and Crystal) is how they interpret the manifesto and principles, and roles also differ between them. Our focus here is however on the broadest interpretation of agile and not any one implementation in particular.

Here are some examples and explanations of some of the key concepts or terms in agile methodology and what they might constitute in an educational context.

- **A story** is a definition of what needs to be done. The description is from the student's point of view e.g. Learn concept X or Understand the significance of Y. In this method stories will be called *items*, and they represent, in the simplest terms, what the teacher will teach (tasks of learning or instruction). They cannot be broken down any further and are evaluated at some "cost", which is the story's estimated effort, in e.g. story points or work hours. The cost is decided in a planning meeting.

- **The backlog** is a list of stories that are yet to be completed. The backlog is a sorted, prioritized collection of stories.

- **A milestone** is a major progress point, which signifies an end of a certain context.

- **A sprint** is a block of time devoted to a certain context and in software development those last from 1 to 4 weeks, in this method 2-4 weeks is encouraged. There is a rhythm to a sprint. It starts out with a planning

meeting, daily stand up meetings and ends with a retrospective meeting. The development team members divide among themselves the work that the product owner decided was to be completed this sprint.

- **A visual board** or a task board is a visual aid. It is a tool where the current sprint's tasks, for each team member, are visually represented in columns. The tasks are moved along the columns e.g. To do - Doing - Done. The reasoning is to get a visual feeling for all the work done by each member.

Figure 2: The agile manifesto as it was written in 2001 and the twelve principles. These are the agile guidelines referred to in this handbook [2].

[2]Image credit: Agile Alliance

3 Why Agile for Educators?

The reason to try to combine agile with teaching is to bring into the field of education some valuable experiences and expertise made in the field of software development, to make teachers more responsive to the needs of their students, to create a method that lets students know that their voices are being heard, and make the process of teaching and learning more dynamic. This agile educational method looks to a philosophy that is based on the agile manifesto, it looks to the structure of roles of the stakeholders involved in agile methodology, and the benefits of software development structure around communications and processes. The real reason to get familiar with any new teaching method is to benefit the student. Here are some of the reasons this method should benefit them and the teacher.

- A number of resources cited in the thesis, based on research in education, sleep and other fields, show a correlation between preferable study outcomes, health benefits to the student, and good student participation, with assessments that this method looks to.
- This method encourages teachers to aim for context-based learning. By getting teachers of different subjects to work together (for interdisciplinary and context based approaches) on course design and synchronizing their schedules.
- A flexible schedule divided into independent units. This is not a new concept to most, but these units come with more structure than just time division.
- It gets students invested in their own success by making their collaboration important. Social constructivist writing talks about the need to cultivate metacognitive skills, and by getting students involved in the planning and implementation of the course they are taking, those will be enhanced.

The hope is that this method helps to maintain a meaningful conversation and give support to teachers who want to be more responsive, taking student input more into account.

In figure 2 the manifesto and principles are laid out from the software engineering perspective. The following propositions and principles are an "educational agile manifesto" or this method's philosophy for teaching and learning contexts. Notice how some are unchanged from the original document. The reasoning is that some of them are general enough to be already applicable to teaching and learning. A more detailed look at each one and their argumentation can be found in the accompanying thesis.

The Agile Teaching Manifesto

1. Individuals and interactions over processes and tools
2. Meaningful work over rote learning
3. Student collaboration over authoritative ways
4. Responding to change over following a plan

The Agile Teaching Principles

1. Highest priority is for the student to gain mastery through engagement in class
2. Be ready for change, even late in the course. The agile method is responsive so that the student's benefit is always at the forefront
3. Assess student mastery frequently with a preference to a short timescale
4. Teachers must work together with other people frequently throughout the course
5. Build an environment that supports students in their efforts and trust them to work to get the job done
6. The most effective way of getting information across is with face-to-face conversation
7. Primary measure of progress is the students' ability to apply knowledge
8. The method promotes that a sustainable pace be set for the course
9. Continuous attention to professional excellence enhances agility
10. Simplicity is essential
11. The best projects come from self organizing teams
12. At regular intervals the students and teacher reflect on how to become more effective.

These are the principles and manifesto of agile as this method attempts to translate or localize them into the context of teaching and learning. The next chapter goes into detail on implementation.

4 The Agile Educational Method

An visual representation of the method is shown in figure 3, and each element will be described in detail in this chapter.

4.1 Roles

In chapter 2 the different roles of agile are presented as they appear in software development. To fit them into teaching some rework is due. Namely that of defining what roles fit the teacher and the students.

- **The teacher** is the **product owner**. They decide what items from the backlog are added to the sprint. This is equivalent to organizing blocks of the syllabus as the course progresses. They are responsible for the direction the class takes and at the end of each sprint evaluate its results. The teacher should also be on a team of peers when designing the course. That team work is tailored more to traditional course design theory than agile methodology and therefore the teacher team is therefore not equivalent to a team of developers However, an extension of this handbook might be an implementation of an agile framework for teacher-teamwork. See table 1.
- **The students** are both **the customer** and **the (developer) team**. This might seem contradictory but these different roles are applied at separate junctions in the method. On one hand it is the student who is asking for the product (their education) but they also master the subject through work. That is why the students have both roles. The customer role is tied with maintaining the conversation about priorities and expectations with the teacher. The developer role is tied with doing the course work and delivering projects. See table 2.

Teacher as the product owner	**Teacher as a member of a teacher team**
Chooses items from backlog to add to the sprint and maintains the backlog	Designs the course
Holds all the required meetings with students	Evaluates the course

Table 1: The different roles, and added responsibility, the teacher takes on in this method.

Student as a customer	Student as a developer
Maintains a conversation that helps enhance the learning experience	Participates in sprint planning and work
Reviews the sprint to align priorities	Delivers projects on time

Table 2: The responisibilites of the student as a customer and as a developer.

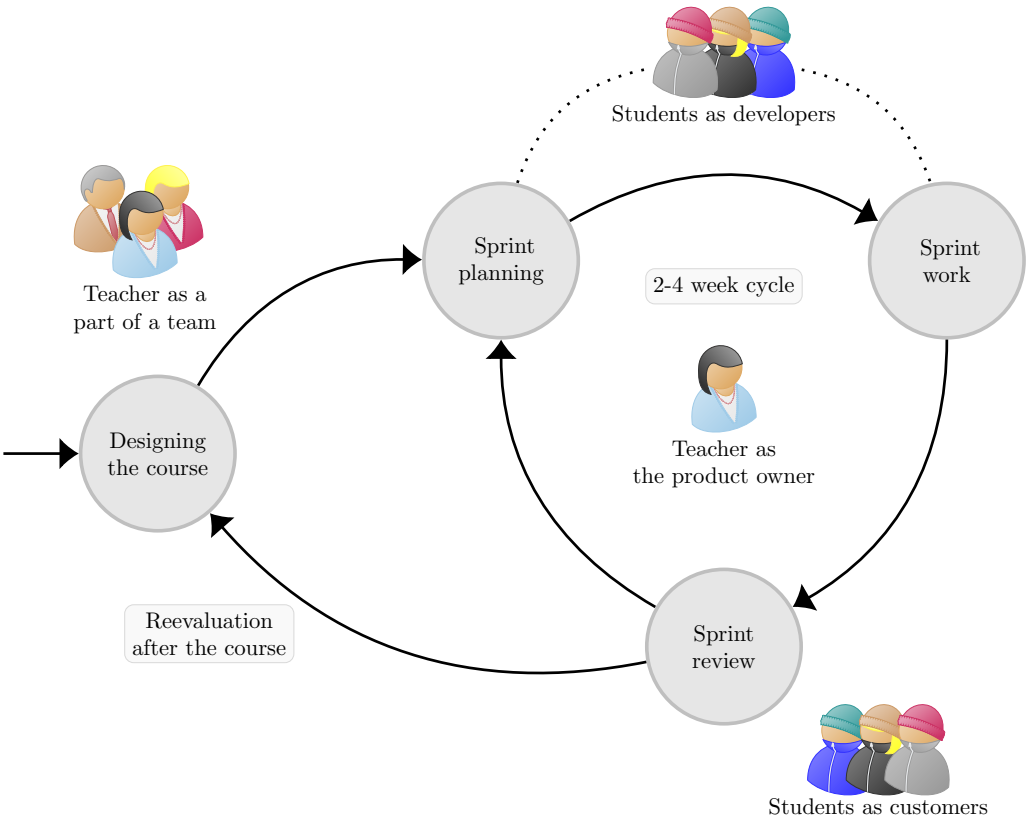

Figure 3: The visualization of the agile educational method for teachers

4.2 Course Design

A key element in this method is that the design and structure of the course itself is in the hands of the teacher. That teachers aren't handed a course that they try to apply agile methods on with patchwork. What is suggested here is a structured way of designing a course to get benefits from a successful software engineering methodology combined with an educational theory that revolves around conversations with students.

In Icelandic secondary schools, course design is usually in the hands of the teacher or a group of teachers that are covering the same material. Working in teams to design courses is familiar to teachers, who mostly work with other teachers from the same domain. The method presented here, however, recommends a somewhat different approach, a more interdisciplinary and context-based cooperation or collaboration.. What this method proposes is that there is a self organizing team working on the design and each member plays an important role based on their expertise and insights. A teacher might be on a number of such teams, working towards designing their own course or helping to design other courses they may or may not be participating in via interdisciplinary work. The motivation for this participation is to to maintain the teachers' own professionalism and to benefit the student with a more context rich education.

The course design is done in team meetings before the course begins, where a facilitator leads the way.

Roles on teacher teams:

- Facilitator is a person that serves the role of team leader, has a background in teaching (other than the subject at hand) and preferably course design work. Since this person doesn't have the background in the field they may ask questions that would otherwise not come up and can bring to light items that might have been overlooked. This role could sometimes seen as the voice for students (if no former students are on the team) because they don't have expertise in the subject and bring up things that a student could get confused over. This is the person that keeps people on point, on track and goal oriented.

- Educators; people with background in education theory that have intimate knowledge on how to go from theory to practice.

  - Domain specific: The educators with expertise on the subject to be taught, the teachers who will teach the course. These are the people who have the most to say on which instructional items to add to the backlog and how to informally evaluate them.
  - Interdisciplinary: Teachers of other subjects who are there to work out schedules to fit their own courses for context based learning. These are the people who find items from the backlog they can work with so that context can be established between subjects.

- Support staff; a person from the office of the institution that knows details

about practical matters, budget and other constraints or opportunities. This is the person who knows about all things logistic.

- A novice; a former student of a class of this type. They would voice concerns that teachers and staff maybe hadn't thought of. This role is not vital but could yield valuable data nonetheless if included.

The outcome of this work should be a course that fits this method with the following being clear:

- A syllabus ordered into sprints and milestones. The sprints cover the time of the course, their duration may vary depending on context and need, but within the limit of 2-4 weeks each. It could help to name the sprints but they can also just be assigned numbers. The sprint schedule should be able to accommodate enough of the items from the backlog, for the educational purpose of the course to be met. If multiple sprints share a project grade or exam grade a milestone should be included. The milestone signifies a certain context and it's use is to highlight how sprints are connected (see chapter 4.2.2). This initial planning is done to estimate how much can fit in each sprint, but the sprints are empty until sprint planning starts.

- Accompanying this sprint schedule is a backlog of educational items that have been broken down into understandable terms. Any item that is not clear needs to be either made understandable or broken down further. The items are then informally given a cost (time, effort, etc.). If an item's cost exceeds any one sprint it should be broken down further. The total cost of the backlog is evaluated by the team designing the course and the cost should exceed the time available by a small margin to make sure there is more material to cover if all goes better than expected. The items then need prioritizing and can be placed under milestones, the prioritization is to give the teacher a clear way of discarding items if necessary.

- A list of resources that are available to the students, and cover the material of the course, allowing the students to use materials that fit their needs.

- The type of assessment to be used and the grading scheme. Whether the grading is tied to the individual student or a group, partly or for all the course's assessment.

- The way in which the teacher plans on holding all the required meetings with students. A rough draft for preparations needed for each meeting, which will be introduced later.

The course then meets the requirements of this method. When the course is over the method requires it to be reevaluated and improved with each new installment. This will not be news to teachers, who are continually improving their courses.

4.2.1 The Syllabus.

Informal evaluation of the backlog is important for flexibility. The teacher team estimates what cost the students are going to spend on each item when designing the course. So to kick off the course the teacher should give the students some room for adjustment by only planning for 80-85% of each sprint's time.

By giving the students this leeway, they are given extra time for what they struggle with and not what the teacher team believes they might struggle with. However if the students are not struggling with anything there is always the possibility of inviting them to grapple with more advanced material.

To work in an agile way students must be able to give feedback that influences the course and how the lesson plans play out. This helps maintain a knowledge-centered atmosphere where students are encouraged to participate on their terms. Meaningful feedback is therefore not just a number or letter on an assignment but a conversation on what can be done based on the project-work and meetings.

This method proposes two ways of doing this on the syllabus level, although they are not mutually exclusive and by no means the only solution.

- **Assigning weights** to the syllabus. To get a certain level of engagement at the start of the course the teacher provides the syllabus with the breakdown of each item (or group of items) without the value assigned to it. Coming up with that value is in the hand of each student or team of students if the course is designed for team work. This way the students are allowed, within some logical boundaries set by the teacher, use their own strengths by assigning more weight to their forte or challenge themselves by placing them inversely. The boundary would be so that the weights couldn't exclude some important educational requirement. The prioritized backlog should determine which are the minimum weights for each item.

- **Reiterate** some part or parts. Give the students a chance to work on items from a previous sprint. This could be allowed if a certain minimum average result was not achieved on some item. Here it would help and be important to already have a minimum threshold established (preferably with input from the students).

In designing the course it is very important that the syllabus be divided into sprints, a core element in agile implementations, with each sprint having its own theme and concrete concepts to work with, not overlapping other sprints except to build upon previous ones. A sprint looks familiar to most teachers who organize their syllabus into blocks. The first few sprints are to find out the velocity of the students, and to find a good working pace, getting used to the method's need for meetings and so on. By dividing time in such a way a valuable time-frame is created, in which meaningful work can take place and adequate reflection upon that work can be done. This is to satisfy the eighth principle: „The method promotes a sustainable pace be set for the course". The sprint

length is decided when the course is designed. The sprints in this method are preferably between a minimum of two weeks and a maximum of four weeks. In software development the teams are usually working full time on their project, in the educational setting it is only a few hours a week and that needs to be taken into account. With one week there are at not that many hours with the students and with the overhead of meetings and conversations there is not enough time to get any teaching done. If the sprint is longer than four weeks the feedback from both students and the teachers might not be precise enough and might not seem to matter that much. For instance the typical semester in an Icelandic secondary school is 18 weeks, so with sprints of five weeks there would only be three full sprints. Different context might might require different sprint lengths, but they should all be within the two to four week limit.

4.2.2 Assessment

Assessment might be the most important part of this method. Students' progression is important and successfully assessing their achievements is instrumental, as principle seven demonstrates: „Primary measure of progress is the students' ability to apply knowledge". Formative assessment is highly encouraged, because that is rooted in continuous conversation about the success, performance, and goals of the student. Formative assessment fits well with organizing the syllabus into sprints, with each sprint having its own focus. It fits well with the third principle: „Assess student mastery frequently with a preference to a short timescale". A final exam is strongly discouraged for a number of reasons, and an exam is only encouraged if it applies to the current sprint or the sprints under one milestone, to make sure students work steadily and in a sustainable manner. Frequent assessments with lower stakes are beneficial to the students' sleep, health, and memory. Research has shown that continuous assessment has a positive impact on academic outcomes as well as student satisfaction. High stakes make students more prone to sleep deprivation which negatively impacts health and memory. The method of assessing should be fair, transparent, and lead to a conversation on future work. A final exam isn't inherently unfair but the feedback given to the student is a single grade and not as a continuous conversation about their own interests and potential in the subject.

When working with the students on attaining knowledge there is always a need to make sure that mastery of the subject is at hand. There are two primary ways in which to ensure this. First there is the definition of mastery. The students should, at the beginning of the course and of each sprint, know what is expected of them, so that they are working in a organized way toward the same goal as the teacher. By using formative assessment goal-setting is done in cooperation with the students. Secondly there is the need to keep things simple and relevant. By defining mastery for one item at a time the students can get a better grasp of what is asked of them. By only looking at the current item's criteria for success the students can get a better sense for setting their own metric for achievement.

Before the end of a sprint, the sprint's workload should be completed for

assessment. Regardless of whether it was a project, exam, portfolio or other way. This is to make room for the retrospective. If the students are satisfied with their work they'll want to continue, if not they might want to repeat some parts. It is possible to push items from the current sprint into the following sprint, and therefore make it a part of the latter's assessment. Pushing work into the following sprint and assessing it again is a way of showing the student their success, input and participation is important. This however might make the next sprint more costly or risk cascading work until the end of semester. The flexibility in time planning should handle that possibility. To answer the different needs of students, fluid sprints might be needed (looked at in chapter 4.2.4), to push parts of the previous sprint on an individual or team basis into the next sprints assessment.

Assessing teams is a different challenge because the process needs to be fair both for each member and for the whole team. Here are some ideas on how to evaluate the work done by teams:

- The individual's grade is the mean of the grades of everyone in the team. This creates the incentive to help out members who are struggling instead of treading on and leaving them behind.
- Each student gets a vote at the end the course to evaluate the other team members' contributions as below average, average or above average. That will then be represented in lowering or increasing the weight of a team's project grade compared to an individual exam grade (or another grade if there is no exam e.g. having an individual part of the group project). This creates the incentive of making sure the student's contribution is proportional to their effort on the project and if it was lax then they'll need to make up for it somehow.
- Students get bonus points to award at the end of the course, where the student who receives the highest number of points gets a grade increase by some amount or percentage of points, while the person with the lowest number of bonus points gets no change to their grade. This will create a greater incentive to outperform the other team members than the vote described in the previous point because it is a more tangible way to affect the grade.

4.2.3 Interdisciplinary Sprints

One of the three major focuses in social constructivist writing cited in the thesis accompaning this handbook is to give a context-rich experience.

With an agile approach teachers can work with one another to integrate lesson plans of different subjects, to create context based learning experience for the students. This is much like the agile idea of cross-functional teamwork, where people with different skills and experience work together. This way one sprint would have a particular context to base it in, and the next sprint could

have a different context. Interdisciplinary education has gotten some traction recently and there are signs that it gives students an advantage.

The cross-functional teamwork of teachers is valuable because it benefits the students by making their education more immersed with other subjects.

4.2.4 Fluid sprint work

When the sprint is over and it's clear that a certain section of the students would greatly benefit from looking back but another from continuing it could be best to let each team or individual participate in their sprint of choice. This will only work with student teams, accompanying software or non linear learning, and a major shift in school policy. None the less it's an important point to make. This would mean that mastery of a subject happens at a pace fitting the learner, but not necessarily the school system. To assess mastery at a pace fitting the student the work done would supersede the semester and the course would be fluid. That's where major shift in policy comes into account. This has already been implemented in Fjölbraut við Ármúla where students can take a math introduction course, the material is divided into blocks and after each block there's a test. If the student gets a high enough grade they can start the next block. The course ends when the student finishes all the blocks.

Using fluid sprints, each team or individual would get a unique sprint with items chosen from the backlog for them specifically. The students would then each get tailored sprints. However the sprint schedules would only be fixed time-frames and not have predetermined themes and contexts. As already stated, this way would require major policy change since the time for assessing mastery of the course's requirements is not fixed.

4.2.5 Teaching Materials

An important part of learning is the materials used. Choosing materials in conversation with the students and being able to update them as needed is a very agile approach to teaching. Materials can be anything from test-papers to software. Of course it's important that the teacher is familiar with the material at hand but expertise on the subject should equip the teacher to use fairly new materials at the suggestion of the students. The teacher should get acquainted with a number of materials and give out a list of resources so students can find a suitable fit for themselves.

This is why the fourth principle is so important: „Teachers must work together with other people frequently throughout the course" There is no need for each teacher to rediscover the wheel.

4.2.6 Teaching Method

An agile teacher can change their teaching method if they see it's not working and in conversation with the students, engage them in a manner that better suits their needs. This is in accordance with the first principle: „Highest priority is for the student to gain mastery through engagement in class". If the

method they are using doesn't work towards the fifth principle, it should be altered: „Build an environment that supports students in their efforts and trust them to work to get the job done" Given enough feedback the teacher will have concrete ways in which to improve to better suit the needs of the students. This might be something simple as „please raise your voice" or something that requires more flexibility on the part of the teacher like „can I work alone?" Most important here is to engage the students in conversation about their experience and maintain the conversation for the remainder of the course.

Teaching methods that work alongside the course's materials need to be chosen. Some teaching methods offer more flexibility than others, these listed are encouraged and reading materials on them is in the resource chapter (chapter 5):

- Effective group work
- Flip teaching
- Project based learning
- Problem based learning
- Team based learning

Student Teams. An integral part of agile is the development team, and this method proposes using teams as a unit for students. The class should be divided equally into teams of 4-5 people. Teams must be chosen either based on an initial assessment test where students of varying ability and expertise are chosen together (students with background that complement each other) or at random. The team then chooses for itself the weights on the syllabus (if that option is available) and works together on all projects for the remainder of the course. To create an incentive for participation there are a number of ways of assessing the team and its individuals. Some ideas for assessing teams may be found in chapter 4.2.2

To get the students to work together on a project without a carrot of added points for a final grade some arbitration is in order. That is the name of the project-work itself, and it has to be a collaborative term (co-creation, co-design, co-...) according to the people behind Effective group work.

Explaining to the teams in the beginning that teamwork is hard will hopefully get them to understand what's in store. Team based learning with the right setup however does not require incentivizing because the team becomes self policing. See further reading on Effective group work and Team based learning in chapter 5.

Even when the team approach is not chosen the pair-programming method might fit, when two students with different skill sets work together on a single workstation (a computer, or single paper to hand in). This is only for so called blitz-projects and is not a good fit for an entire course. Some agile experts even deter others from this approach, in the case of senior-junior pairings. The teams should not include such wildly differing abilities if it can be avoided.

4.3 Sprint Planning

The teacher decides the sprint's scope, before it starts, by choosing what items from the backlog to cover. This can be done for each team but for the remainder of this sections it is assumed the whole class is in sync.

Items not finished from the previous sprint could bleed into the current sprint or be discarded. The items taken into consideration must be doable in the time frame allotted. If an item in question is larger than the time available it must be broken into parts. When there are student teams there is an allowance to take on more cost and allocate the work within the team, see also chapter 4.2.6.

Then the class goes over the items and plans how to fit them into the sprint by deciding their effort in the planning meeting. There are a number of ways to get the students to help in evaluating the cost of the items. Without software to help out it might be too time consuming to do so accurately. A simple way would be to have three types of difficulties - small, medium and large. The students would then, with a majority vote of hand raising, choose what class describes the time needed for the item. The teacher would subsequently translate those types into time division for the items for the rest of the sprint. With software it would be possible to ask each student to divide the sprint's time on its items, then average all the students answers to get a democratic result.

After the scope has been evaluated the goal setting comes in, where the students set themselves goals and help shape how the formative assessment of that sprint will look.

The element of sprint planning is focused on work that enhances metacognition and deals with preconceived notions (how they view the items), which touches upon two of three main focuses in social constructivist writing cited in the thesis.

4.4 Sprint Work

The sprint work is referring to the part of the method that deals with actual educational content being covered by the teacher and students working on projects.

Here the so called daily meetings and class work take place. These meetings are to satisfy the propositions of the manifesto; to place value on the interaction of people, responding to change and make sure meaningful learning is taking place. It also addresses principles two, five and nine. *The daily stand-up* is a meeting that is held at the beginning of each lesson. The daily meeting is about assessing what the student is doing right now and what they need help with. Helpful here is to use a visual board for a placeholder for each student. The meetings purpose is usually fulfilled with a series of simple questions:

1. What did I do towards the sprint's goal?
2. What will I do toward the sprint's goal?
3. What do I need help with in order to achieve the sprint's goal?

Questions should be hand picked for maximum relevancy. A general example would be:

1. What concept/problem did I work with?
2. What will I do to continue my work?
3. What do I need help with?

By getting this information at the start of the class the day's lesson should be tailored to the needs of the students. To do this with a large group of students the teacher would either have to use software tools or organize the class around student teams that could then each have their own stand-up.

After the daily stand-up is finished the lesson continues according to the chosen method, covering the items on the backlog. Here the aim is to maintain a sustainable pace. Workload in each sprint should be equal to what the class is capable of in a sustainable way, so that the cost of the sprint items is a number that depends on the class in question. If this class is full of high achievers they might handle a heavier workload, or more cost.

Items on the agenda for the sprint should always be visible. A helpful tool with organizing items from the sprint are vision boards, see further in chapter 5.

4.5 Sprint Review

The feedback given in review by students, now in the role of customers, influences how the rest of the course plays out.

At the end of a sprint, after the assessment, there is the retrospective meeting. No teaching takes place but instead the class reflects upon the work done in the finished sprint. These meetings influence metacognition skills, create a knowledge-centered atmosphere and cultivate conversation with the students, all of which are vital in teaching according to social constructivism. The goal is to make sure that the focus of the sprint resulted in benchmark mastery or whether there is need to reiterate some subject matter, ensure the method and material were fitting, and last but no least to make sure no one is being left behind. The retrospective conversation about the assessment, previous work and next steps is very important, and should give the student the impression that their work matters and that the teacher is invested in their success. There are many different ways structure this meeting and a valuable resource is to be found in chapter 5.

Feedback from the retrospective is what the teacher uses to continue with the course. Students are given a chance here to voice opinions on the progression of the course. The teacher reconciles the feedback from students with the course's plan and is then able to choose items from the backlog to make the scope for the next sprint.

Reviewing all the other feedback is important as well. It might not be apparent on a daily basis but looking back at the stand up data, with the help

of software, a trend might emerge. That trend would indicate, along with the more substantial evidence from the retrospective, whether to continue or to repeat some parts.

Feedback should be digitized or stored in such a way that after the course is complete it may be used to reevaluate the course for future installments.

5 Resources

This is an iterative method and using action research alongside it could streamline the process. Applying action research to the implementation is not only a good way to gauge the method's performance but is also very agile in and on itself. For a teacher to also be a researcher, studying the effectiveness of their work, is encouraged. Hafþór Guðjónsson has written extensively on the matter from the perspective of chemistry and science teaching in secondary schools in Iceland, and from the view of the academia. As has significant work been published from the works of the secondary school teachers at Menntaskólinn við Sund available on the school's website

The handbook of formative assessment, edited by Heidi L. Andrade and Gregory J. Cizek, is a good read, particularly what Dylan Wiliam wrote, who has been a prolific contributor to research on formative assessment.

A handy guide for how to conduct and organize meetings, what to do with teams and how to empower people is at the Atlassian website

A list of resources for the mentioned teaching methods:

Jonathan Bergmann's and Aaron Sams' book *Flip Your Classroom: Reach Every Student in Every Class Every Day* on flip teaching is available in its entirety here, and it is short, informative and concise.

To get familiar with the the idea behind Effective groupwork the book *Effective Groupwork, 2nd editon* by Michael Preston-Shoot is a very good start.

Sigríður Helga Sigurðardóttir wrote a comprehensive masters thesis from the School of Education from University of Iceland on project based learning. In it she cites multiple sources on the benefits and usefulness of this method. Available here

An accessible account of problem based learning is in John Barell's *Problem based learning: an inquiry approach.*

Team based learning has all its resources available online at teambasedlearning.org/ and I had the privilege to sit a very interesting seminar for this method.

A valuable insight into all the research that has been made on sleep and it's impact on life and health by Matthew Walker is in his book *Why we sleep: unlocking the power of sleep and dreams.*

And last but not least the thesis accompanying this handbook, available at skemman.is here