**Introduction**

The University of Glasgow is currently developing a work-based degree in software engineering, aimed at improving the skills gap that exist between students and the expectations of employers in the industry. However, research shows that softer skills such as communication and writing although ranked high by employers, remain unprioritised within the engineering curriculum. This resulting in engineering students often are reported as lacking these skills by employers and are performing poorly in their dissertations.

The Patchwork text is an educational method that has proven successful in improving writing and communication. and has been implemented in fields such as social work and nursing which has a large practical element to it and where self-reflecting in practice is seen as a vital part of learning.

This project is to create a software prototype that implements the patchwork text method in the context of a work-based learning software engineering degree at the University of Glasgow. The objective is to integrate writing into computing science practice, and ultimately improve students communication-skills and academic writing.

**Background**

Research shows that students in engineering often lack competence in academic writing and communication, skills that are highly valued by employers (Ibrahim, 2017). The gap between the expectations of the employers and the students abilities is something that has been of concern for universities, and many have taken steps to improve students writing and communication skills in order to increase their students employability. (Itani & Sror 2016). However, efforts in trying to include more writing within the curriculum has rarely proven successful, often due to an inability to integrate the writing into the learning itself. These efforts have instead been in the form of separate modules that are not related to the subjects the students undertake (Goldsmith & Wiley 2016). The modules that teaches writing and communication are also often of an optional type, communicating to students that these skills are not as important as technical skills (Itani & Sror 2016).

**The Patchwork text method**

The patchwork text method is a method that addresses many problems of the teaching and assessment of writing in education. Due to the method, other forms of assessment such as the essay fails to accomodate students with different abilities and experiences of academic writing, resulting in many students achieving poorly in these modules. It recognises that writing is a skill that demands critical reflection and deep understanding of a topic, and argues that the essay format encourages superficial learning where students learn how to play the game rather than developing real knowledge.

The method aims to improve critical thinking and deep learning by proposing an alternative structure for assessment that unifies retrospective summations of knowledge with the view of knowledge as something gradually accumulated (Winter & Scoggins 1996). Trevelyan & Wilson (2011) summarising the key objectives of the patchwork text method being continuous learning, deep learning, integrated understanding of a topic and meta-cognitive self-reflecction.

**Structure**

The main characteristics of the method is that students generate patches in forms of small individual submissions throughout a learning module. This divides essay writing into more manageable pieces, making it easier for students who are less exposed to writing practices to complete their submissions.

Every patch are being shared and reviewed within small working groups, giving students a possibility to get feedback and learn from viewing other students submissions. They also receive formative feedback from their teacher on every patch, and get the opportunity to edit their original patch.

In the end of the module these patches are being stitched together into a unifying whole. Students get to choose what patches they want to weave into the final submission, underlining the autonomy of the learning progress . Students also submit a reflecting piece where they get to think about their progress and learning process. Due to the method, this forces an element of self-reflection, which is an important factor for critical thinking.

DIFFERENT IMPLEMENTATIONS

Apart from the basic structure of the original method, there are also differences in how the method has been implemented.

**Challenges in implementing the patchwork text method**

An important part of the patchwork text method is that students in every patch share their submissions within smaller groups and are offered formative feedback from their peers (Winter 2003). A challenge is that the success of such a peer reviewing is very much dependent on the quality of the feedback, which in its turn depend on the feedback literacy of the students. (Carless & Boud 2018). Some factors that can increase the quality of the peer review is to scaffold it by using directed questions. This way a teacher can guide students and make sure they assess their peers after certain criterias or rubrik, which improves the quality of the feedback even among less able students (Meek et. al. 2017), (Staubitz et. al 2016). Formative feedback from tutors for every patch have also been found to be an important factor for the success of the method. A challenge for implementing this however and an overall criticism of the method is the increase in resources it takes and the extra pressure it puts on teachers (Sapouna 2016). Assessment within the patchwork text method is also complex due to the evolving nature of the method. Teachers are not only assessing the final quilt, but the learning process and how well students have responded to feedback throughout the whole module. This means a lot more time and resources has to be spent in order to implement the method, which

**Adaptive Comparative Judgement(ACJ)**

Adaptive comparative judgement(ACJ) is method that is based on the assumptions that human judgement is relative, and that the value of an object therefore only can be retrieved in comparison to another object. The method has started to be used within education as an alternative marking method and has proved to be much more efficiant as well as reliabile than other methods such as criteria based marking and rubrics (KALLA) It also supports different formats and can favourably be used in areas where judgement of pieces is less objective such as arts and writing (Pollitt 2012).

**Method**

The method is a development of Thurstones comparative judgement, which states that the difference in quality between two objects A and B is equal to the logarithm of the odds that A will beat B, and is expressed in the following equation:

Scripts are randomly presented in pairs for judges who simply pick which one of them is better. If script is picked as a winner it adds on a 1 to its total score, and 0 if it loses. In the second round, scripts are paired with scripts that has a similar score as themselves. It is this that is the adaptiveness of the ACJ, in that the scripts only are paired against other scripts where the information of the odds of them winning is not given. After a few rounds of pairing scripts with similar scores to each other, the Maximum-Likelihood method is applied to optimise the value parameters. This is done by first calculating the Score of a certain object, which is the total number it has beaten the other objects. Then the Estimated score is calculated, which is the sum of the probabilities that an object beats the other objects. The following equation is then applied to update the estimated value parameters:

**Existing software implementations**

There are no existing software that are specifically designed for the patchwork text method. The previous implementations of the patchwork text method that have used digital tools have often used range of different mediums, such as blogs and online forums, but also personal learning systems and virtual learning environments like Pebblepad (Chesney & Marcangelo 2009), Blackboard(Marcangelo 2011) and Moodle (Hager et. al. 2016).

Interview 28/06 Derek Somerville and Jack Parkinson

Glasgow university is currently developing a 4 year work-based learning degree in Software Engineering that will start in October 2019. The following information has been gathered during an interview with Derek Sommerville and Jack Parkinson 28/06.

The structure of the degree is that students will have a mentor and a coach at their workplaces. The mentor would be in charge of the technical tasks and the coach of the students progress and general HR-aspects. There is an existing system being planned on Moodle that will work as a unifying bridge between the university, the student and the employer. The mentor can follow the students progression on there, as well as viewing a common timetable showing important dates so that employers does not assign students big tasks just before an exams for example. The current system will not contain any forum for students, but it is expected that will take place in an external application. There might be some reflective pieces being done also in this application.

Issues of confidentiality were discussed, how can employers be in the loop of the students writing progression without there being conflicts, for example a student might not feel free to express themselves.

Requrements

Users

The users in the system are teacher, student and judge. A potential user was also the mentor that students have on their work placements. It was decided based on information gathered from the interview that the main potential role of a mentor in the system would be to follow the progress of its student. The benefits of this would not outweigh the negatives, that the presence of a mentor might affect the students ability to apply critical thinking towards the work and workplace. Further, there would exist a separate system where the mentor would be able to track the students progress.

**Must haves**

The following Must have requirements are based on the earlier presented literature, and is representing the essential elements of the patchwork text method.

Student can view timeline of patches

Student can view individual patches

Student can make a submission to a patch

Student can view and give feedback on group-members’ patches

Student can view feedback on own patch

Student can edit original patch

Student can view all own previous patches

Student can write final submission

Student can write a reflective piece on own patches

Teacher can add students

Teacher can divide students into group

Teacher can add patches

Teacher can add instructions to patches

Teacher can view students patches

Teacher can view students final submission

**Should have**

The Should have requirements are features that though not essential for the patchwork text method, can be described as important for the method to work effectively as discussed in the literature review. Using rubrics as scaffolding for peer reviews increases the effectiveness of it, especially when students are unfamiliar with criticising each others work. In this context it could therefore be seen as a feature that the system should have in order for the essential element of peer review to be effective. Further, formative feedback from teachers is also describes as an important element for the patchwork text method, but a difficulty when it comes to the extra workload it puts on the teachers. A should have requirement of the system would therefore be to offer teachers the ability to judge students work through ACJ, since this has proved to be much more effective.

Teacher can give formative feedback

Teacher can add peer review rubrics

Student can write reflective commentary on own patches

Student can view own edit history of a patch

Student can view own edit history of all own patches

Student can pick what patches to use for final instruction

The could have requirements are based on flexible implementations of the patchwork text method, allowing the teacher more flexibility when designing the module as well as giving students more choices.

Could have

Student can view its own score relative to its group-members

Student can view its own score relative to the course

Student can choose order of patches

Will not have

The ability to upload multimedia have been less prioritised due to the aim of the project being to improve writing ability. Further, a mentor will not have access to the system at this time.

Student can upload multimedia

Mentor can view students submissions

Mentor can view students score

Non-functional requirements

System needs to be accessible from different locations

System needs to be accessible anytime

**Design**

System architecture

An important non-functional requirement of the system is that it needs to be accessible from different locations, due to the users being students on work placements. The nature of the patchwork text method also requires the ability for students to work on their patches when it suits them. The easiest way to accommodate this was to implement the application as a web application, since it would mean students could access it from any location any time.

Model-View-Controller (MVC)

The MVC design pattern separates the business-logic of an application from the presentation layer, with the aim of decoupling these modules.

ER-diagram



Implementation

Django

Since the must have requirements for this project was quite extensive, it was judged as important to use a framework in order to speed up the development of the application. Django was chosen over other frameworks since it facilitate rapid prototype development and brings a lot of features out of the box (Kaluž et. al. 2019). It also comes with an admin interface that makes management and testing of the database easier.

ACJ

The acj part was implemented for this project rather than using existing open source code. The reason was that it was seen as important to be able to update and save the states of the judgement process in the database while it was running. Being a web application means that the program is vulnerable to interruption of internet access, which would result in loss of data if the state was saved in the running program only. Since the judgement process is demanding in time and effort for the judges, it was seen as important to minimise the loss as much as possible, and that every judgement made therefore would be saved directly into the database.