Coping with Uncertainty in an Agile Systems Development Course

Toni Taipalus
Ville Seppänen
Maritta Pirhonen
Faculty of Information Technology
University of Jyvaskyla
Jyvaskyla 40014, Finland
toni.taipalus@jyu.fi, ville.r.seppanen@jyu.fi, maritta.pirhonen@jyu.fi

ABSTRACT

Uncertain and ambiguous environments are commonplace in information systems development (ISD) projects, and while different Agile frameworks welcome changes in organizational, technical, and business environments, the incurred uncertainty is known to negatively affect the development process and the quality of the final product. The effects of uncertainty on ISD projects have been studied in the past in real organizational contexts, but the effects of uncertainty on students in Agile systems development have received less attention from scholars. In this study, we measured the effects of experienced uncertainty on students' performance in an Agile systems development course and how uncertainty affected the quality of the system developed by the students using Scrum. We implemented the course using a problem based learning (PBL) approach and simulated uncertainty through various work environment reflecting concepts. Our study reveals that the effects of uncertainty are fairly similar among students and software professionals, and we identified three different coping strategies that students used with varying degrees of success. We present that learning approaches such as PBL enable a befitting environment for students to acquire hands-on experience in coping with uncertain environments, thus mitigating the problems students are likely to face in their work environments.

Keywords: Scrum, Agile, Software engineering, Problem-based learning (PBL), Student perceptions

1. INTRODUCTION

Uncertainty can be considered as one of the key issues in information systems development (ISD) projects; complex and continuously changing technical, organizational, and business environments pose a challenge to the project work. Compared to plan-driven methods, it can be argued that projects implemented using Agile approaches do not similarly restrict changes, and are thus more prone to face contingency factors that may induce uncertainties. Boehm and Turner (2004) state that Agile software development methods are more suited to projects with frequently changing requirements.

The effects of uncertainty in ISD projects have been studied in the past in real organizational contexts, but the effects of uncertainty in education, namely Agile systems development, have received less attention from scholars. In this paper, we examine how groups of computer science (CS) and information systems (IS) students experienced uncertainties they were forced to face over a Scrum-based systems development project and how uncertainty affected the development process and the quality of the system. Our research setting is inspired by the study by Jun, Qiuzhen, and Qingguo (2011) that indicates, among other results, that the level of a project's inherent uncertainty is negatively associated with both process performance and product quality. In addition, studies by Wallace, Keil, and Rai (2004) and Jiang, Klein, and Discenza (2001) provided us with ideas for our theoretical setting in which we set out to examine if there exist dependencies between experienced uncertainty, process performance, and product quality. Furthermore, we were interested in recognizing different strategies the students applied to cope in uncertain circumstances.

The remainder of this paper is structured as follows. In the next section, we detail the course setting in which our data was collected. In Section 3, we discuss the learning approach we used in the course. In Section 4, we describe our research method and data analysis; in Section 5, we discuss the implications and limitations of our study; and we present conclusions in Section 6.

2. COURSE SETUP

In this section, we first describe the course setup and the course assignment. We then discuss the various sources of uncertainty and how they were present in the course project.

2.1 Project Assignment

We built the project assignment so that it would introduce the students to Scrum with an engaging and memorable, as recommended by May, York, and Lending (2016), yet realistic and challenging experience of a systems development project. The course structure consisted of three distinct phases: initialization, execution, and reflection.

In the initialization phase, we presented the students with the intended learning outcomes of the course, i.e., comprehending the phases of systems development, applying Scrum in practice, and implementing a systems development project. We also introduced the assignment: that the students were required to develop a system to be used for managing academic research infrastructures and resources, such as research equipment and materials. The students were then presented with the roles, artifacts, and processes of the Scrum framework according to Schwaber (2004), and each of the students was instructed to choose their areas of expertise based on their skills, interests, or both. In this phase, we also informed the students of the learning approach, i.e., problem based learning (discussed in detail in Section 3); our roles as representatives of the client organization, rather than teachers; and other stakeholders representing personnel from the client organization and its partner organizations. Next, the students were instructed to form Scrum teams of five to six students. This could be done based on the students' personal preferences, social relationships, or in a more purposeful manner, e.g., choosing team members based on their areas of interest or expertise. We did not want to involve ourselves in the team forming process, as, contrary to work environments, one Scrum team formed the whole working unit, and there were no managers or recruitment personnel in addition to the development team. Lastly, we instructed the students to collaboratively use the Scrum framework with weekly sprints to develop the system, and, based on their expertise, identify what they need to learn to effectively solve the emerging problems.

The execution phase lasted six weeks, and each of the Scrum teams produced six system increments during six sprints (Figure 1). We required the students to turn in the system increments at the end of each sprint to oversee contingencies and emerging problems during the development process. In this phase, the students needed to familiarize themselves with the target domain by interviewing the representatives of the client organization, investigating the technical details concerning suitable database management systems and software development frameworks, and reflecting and refining their teamwork.

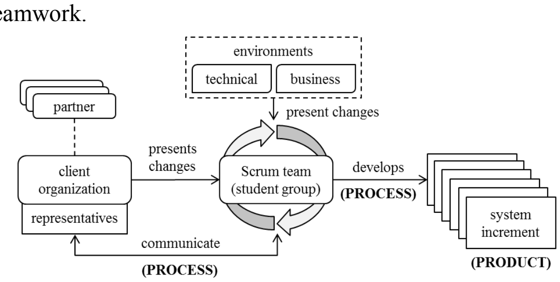

Figure 1. Course Setup

Lastly, in the reflection phase after the six weeks of development, we gave detailed feedback to the student groups as representatives of the client organization. In this phase, we considered the final system as a whole rather than focusing on smaller details as we did in the previous phase. After we had given feedback, we relinquished our roles as representatives of the client, re-assumed the roles of teachers, and gave more feedback to the student groups.

2.2 Uncertainty

As defined by Chu et al. (2014), ambiguity refers to situations with inexact and obscure cues. According to Chu et al. (2014), Budner (1962) classified the causes of ambiguity into three types: new (i.e., no familiar cues), complex (i.e., too many cues), and contradictory (i.e., conflicting cues). Additionally, we present that while ambiguity can be found in most situations, a certain level of ambiguity will lead to uncertainty. Therefore, by uncertainty we refer to the emotion caused by ambiguity. While traditional teaching methods such as lecturing strive away from ambiguity, ISD work environments are uncertain, as presented by Jun, Qiuzhen, and Qingguo (2011). To achieve a level of uncertainty in the course, we designed the systems development project assignment around ten causes of ambiguity (Table 1). Several similar items are also recognized in previous studies (e.g., Barki, Rivard, and Talbot, 1993; Jiang, Klein and Discenza, 2001).

#	Cause of Ambiguity	Type of Cause
1	minimum amount of teacher interaction	new
2	ambiguous target domain introduction	new, complex
3	using Scrum in practice	new
4	development skills are low	new
5	project size is large	new, complex
6	changes and conflicts in student groups	new
7	client and user experience is low	new, contradictory
8	changes in organizational, business, and technical environments	new, complex
9	technical complexity of the system is high	complex
10	many client organization representatives	contradictory

Table 1. The Ten Causes for Ambiguity in the Course – The Three Types of Causes for Ambiguity Presented According to Budner (1962)

First, we gave a minimum number of lectures in the course emphasizing problem solving and project work, and students were required to research relevant topics on their own. This procedure can be interpreted as a cause for ambiguity because students can miss key concepts or even rely on unreliable source material when learning about a topic such as a new data model. Second, we introduced the target domain on a general level during an introductory lecture, and the students were expected to investigate the details themselves. Third, for some of the students the concept of Scrum was new, and none of the students had used Scrum in practice. Even though we gave a lecture on Scrum, the students implemented it to different levels of completeness (e.g., daily Scrum meetings were infrequent.)

Fourth, students' development skills were relatively low. The students taking the course were mainly undergraduate, third-year students majoring in CS or IS, and the students majoring in CS usually had more experience in programming, whereas IS students were more focused on information technology management. Nevertheless, all the students had little or no experience in developing a system as a team.

Fifth, the project size, in terms of the number of students in a group as well as in the scope of the system developed, was relatively large compared to what the students had previously worked with. Sixth, as we expected based on our previous teaching experiences, a number of students dropped out from the course during the execution phase. In these cases, the remaining members of the group needed to redistribute the work within the team. Even though teachers have limited control over the fact that students drop out from courses, this could be seen to reflect personnel turnover in work environments in which human aspects are seen as the source of the majority of problems with software development (Hazzan and Hadar, 2008). Additionally, as we expected based on the study by Dunaway (2013), after the course several students reported social conflicts in their groups. Seventh, as we assumed the roles of representatives of the client organization, we strived to appear as real clients. Effectively, this appearance was concretized by several typical client characteristics presented by Moynihan (1996): client does not have the IT competence and experience, client does not know what they want, client does not understand the requirements of the system, and client does not have enthusiasm for the project.

Eighth, changes in the organizational, business, and technical environments (see Figure 1) presented the students with new and complex requirements. We communicated these changes in the course's mailing list, our e-learning platform's forum, and meetings, representing either the client organization or its partner organizations, or different third-party authoritative figures such as a standardization organization. The changes included, for example, requirements to replace all proprietary products with free open-source products due to budget cuts, to investigate how to integrate the developed system with one of the partner organization's systems, and to research how the system complies with a new, nationwide data security standard. Based on the chosen technologies and the areas of expertise, not all groups were required to react to all of the changes.

Ninth, the technical complexity of the system was not something the students were used to in previous courses. Even if the students chose to implement the system with the simplest of software architectures and software development frameworks, the complexity of the pre-existing systems of the client organization and its partners required complex integration strategies. Additionally, the changes in the technical environments constituted interconnecting technologies that students were unfamiliar with. Tenth, in addition to the course teachers assuming the roles of client organization representatives, we had four additional people from the university staff acting as future end-users with different job descriptions and use cases, and as partner organizations' liaisons. This resulted in conflicting cues in client meetings; some of the interviewees understood the purpose of the system differently or emphasized some features over others depending on their personal preferences. Additionally, due to the background of some of the client representatives, the students were required to communicate technical aspects of the system to non-technical members of the organization, as proposed by Taneja (2014).

3. PROBLEM BASED LEARNING

In this section, we discuss the problem based learning (PBL) approach and how it was realized in the course. During the initial design of this work and the course, we realized that the aspects of PBL, Scrum, and simulated uncertainty operate naturally together, and designing the course setup never felt contrived.

3.1 Background

PBL is based on a constructivist view of learning according to which learning occurs, not because of passively receiving information, but because of the learner's active cognitive and social processing of knowledge (e.g., Bell, 2010; Darus et al., 2016; Tynjälä, 1999; von Glasersfeld, 1984). PBL involves students in problem-solving tasks and allows students to actively build and manage their own learning. The underlying principle is the assumption that learning occurs during unstructured, complex activities (Helle et al., 2007).

PBL has proved to be an effective approach for learning skills and competencies demanded in working life, such as the development of communication skills (Pigford, 1992) and improving problem-solving skills (Gallagher, Stepien, and Rosenthal, 1992), along with team-building and interpersonal skills (Ross and Ruhleder, 1993). According to several studies, there is a positive relationship between problem-based learning environments and deep learning (Groves, 2005).

3.2 Implementing PBL in the Course

We chose Scrum as the framework due to its popularity in the Agile landscape (West and Grant, 2010) and approached the research problem by creating an environment in which we could simulate uncertainty while following the approaches of PBL. Table 2 summarizes how we implemented the aspects of PBL in the course, and next we describe these six aspects in detail.

We followed PBL approaches in the course as proposed by Hmelo-Silver (2004). The information system the client organization ordered from the student groups was an actual system ordered by the university, with the exception that a professional provider was developing the actual system. Consequently, the ordered system had a realistic target environment, purpose, and end-user base. Furthermore, as proposed by Hmelo-Silver (2004), one of the goals of PBL is to help the students become intrinsically motivated by providing intrinsic goals. Because CS/IS students have different professional goals, such as becoming system developers, project managers, or consultants, the problem of developing an information system was engaging, especially when using Scrum. This setting was also motivated by presenting the course's learning outcomes broadly and by reflecting a wide variety of competences (e.g., Colomo-Palacios et al., 2012; Turley and Bieman, 1995) required from software professionals. The students were then encouraged to define their personal learning objectives that correspond to their interests and would prepare them for their future professional ambitions, as proposed by Joham and Clarke (2012).

	PBL	How it was Realized in the Course
Problem	Realistic ill-structured problem	Real-life system ordered by the client organization with an ambiguous description. Scrum is utilized in the development process.
Role of problem	Focus for learning information and reasoning strategies	Students investigate information on the business domain and technical requirements by themselves, and refine their group work on their own, as proposed for sprint retrospectives by Schwaber (2004).
Process	Identify facts, generate ideas and learning, self-directed learning, revisit, and reflect	Students solve problems based on self-acquired information and self-identified facts. Scrum teams are oriented based on the areas of expertise.
Role of teacher	Facilitate learning process and model reasoning	Teachers mostly act as members of the client organization. However, students were given feedback on their work and pointers towards useful information resources per request, and at the end of each sprint, if needed.
Collaboration	Negotiation of ideas; individual students bring new knowledge to group for application to problem	Students act in different Scrum roles. Students have specialized areas of expertise as well as different educational and professional backgrounds. Students are self-directed in forming the Scrum teams.
Tools	Structured whiteboard; student-identified learning resources	A structured whiteboard is encouraged to facilitate the Scrum process. The use of other tools is under the discretion of the students.

Table 2. Different Aspects of PBL Realized in the Course – The Leftmost and Center Columns Adapted from Hmelo-Silver (2004)

The role of the assignment was designed to focus the students into learning information related to the problem and learning reasoning strategies. Learning new information was required in responding to changes in technical, organizational, and business environments. The Scrum conventions of daily scrums, sprint planning, and sprint retrospectives provided the students with a framework on developing reasoning strategies and facilitated collaboration inside the groups.

We allowed the students to choose for themselves how to implement their personal learning process. Based on the final system increments and reports, some groups relied on theory while others implemented parts of the system to reflect each change in the environments. In either case, the students solved problems based on self-acquired information and self-identified facts. Although the problems never had a single correct answer, on occasion some groups delivered subpar solutions. In these cases, we, as representatives of the client organization, expressed our concerns regarding the solutions and suggested that the students take some other approach. We gave lectures only on key topics (e.g., requirements elicitation and Scrum) and focused on facilitating the learning process rather than providing knowledge. Our roles as representatives of the client permitted us to move closer to the students' learning and development process while relinquishing our authoritative roles.

The student groups were heterogeneous in terms of majors, technical and management skills, and areas of interest. Since formulating an engaging problem for a heterogeneous group of students can be a challenge (Hmelo-Silver, 2004), we tried to mitigate this in two ways. First, as per Scrum conventions, each of the students had a role in their respective group. The students were encouraged to assume a role relevant to their interests or backgrounds – IS students could take the role of scrum master or product owner and CS students the roles of development team members. Second, we asked each of the students to report one or two areas of expertise, such as software development, project management, or IT security. These roles and areas of expertise allowed the students to each choose their own learning objectives and outcomes. The areas of expertise also brought new knowledge to their respective groups which helped in the problem solving. Both ways also helped to facilitate the self-directed problem-solving characteristic of PBL.

According to Hmelo-Silver (2004), the whiteboard is a forum for the students within a group to co-construct knowledge and provides a systematic approach to problem solving. Structured whiteboards are a known technique also used in Scrum, and we encouraged the students to use structured whiteboards as well as any resources they could find to facilitate their development process.

4. RESULTS

In this section, we describe the research setting of our survey. We then present the data analyses, and finally, to propose the answers to the research questions, the results in form of factor constructs and the concomitant correlation model.

4.1 Survey Setting

We collected the research data with an electronic survey instrument during a two-week period after the course was finished. We had two reasons for collecting the data only at the end of the course. First, we did not want the students to know that uncertainty was specifically induced. Additionally, we did not want the students to know that they were part of a study until the voluntary survey instrument was presented. In other words, we wanted the study to influence the students as little as possible. Second, we did not want to ask questions before or during the course that would imply the unexpected changes in the environments.

The students answered the questions (Appendix A) on a Likert scale (Appendix B) except for a question concerning the respondent's role in the Scrum team and two open-ended questions. Individual students could not be identified by the responses. Out of the 67 respondents (response rate 70.5%), 15 (22.39%) had worked in the role of Product Owner and 15 in the role of Scrum Master, while the remaining 37 (55.22%)

worked as team members. Overall, there was no noticeable bias causing differences between the grades of the students who responded to the survey and those who did not (see Figure 2).

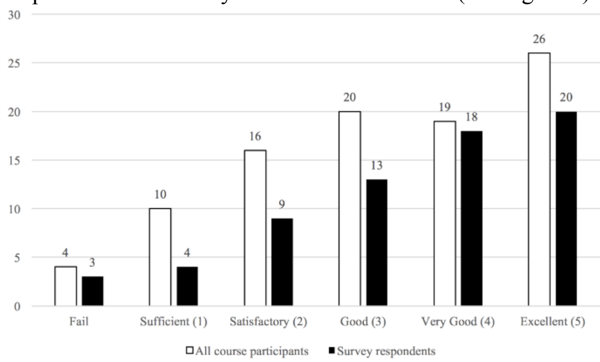

Figure 2. Distribution of Grades among all Course Participants and Survey Respondents

The data collection focused on two themes. We measured the level of uncertainty the students experienced and the performance of the student groups during the period of their six-week Scrum project using several variables. These were then combined into the two sum variables, namely Experienced Uncertainty (EU) and Process Performance (PP), which are both detailed in Appendix A. Additionally, we measured Product Quality (PQ) using the grade that was given for the project deliverables with the emphasis on the final product.

We wanted a higher value of the sum variable EU to signify higher value of uncertainty the respondent experienced, and a higher value of the sum variable PP to signify higher evaluated performance of the respondent's team. The tests on reliability using Cronbach's α resulted in 0.809 (EU) and 0.893 (PP), within acceptable boundaries (Tavakol and Dennick, 2011).

4.2 Analysis

First, we tested the correlations between the constructs EU and PP, EU and PQ, as well as PP and PQ using two-tailed Spearman's rank correlation coefficient (Figure 3). There was a negative correlation between the constructs EU and PQ and between EU and PP, suggesting that uncertainty negatively affects the process performance as well as the product quality.

The strongest correlation in the setting was the positive correlation between PP and PQ, and the results conform with our observations on students' work during the course. They are also in line with the results of Jun, Qiuzhen, and Qingguo (2011), and therefore we can assume that we succeeded in establishing a PBL setting that simulates reasonably well the real-life situations and problems, and that the grade given by the teachers reflected the student group's PP.

Next, we conducted principal component analyses for the items of EU (EU1-EU5 reversed) and PP (PP14 reversed) using Varimax with Kaiser Normalization as a rotation method. The ratios of sample size and number of items subjected to factor analysis were 6.44 and 6.09, respectively, which is an acceptable ratio (Gorsuch, 1983), and the resulting factor loadings were high (see Appendix C and Appendix D).

First, we were interested in whether the factoring of sources of uncertainty itemized in the data (EU1-EU9) would provide additional insights. The analysis resulted in an evident and rather unsurprising two-factor structure (Appendix C) with

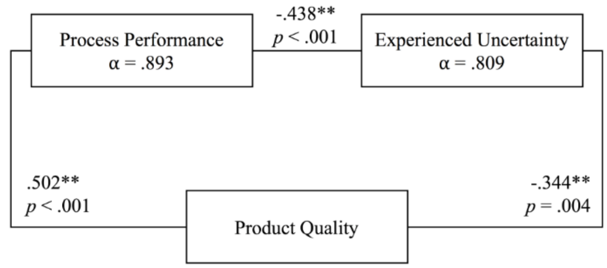

Figure 3. Correlations between EU, PP, and PQ

respective Eigenvalues of 3.641 and 1.684. The factors of *Assignment-induced uncertainty* and *Change-induced uncertainty* explained 59.17% of the variance. The Kaiser-Meyer-Olkin measure of sampling adequacy was 0.778, which is above the commonly recommended value of 0.6, and Bartlett's test of sphericity was significant (χ^2 (36) = 197.535, p < 0.001).

Second, we analyzed the items PP1-P11 to understand the dependence between PP and PQ in more detail. The items PP12-PP14 were dropped because they asked for the overall, and rather unfocused and subjective, evaluation of the respondent's personal performance and his or her group's performance. A three-factor structure for the remaining 11 items explaining 66.02% of the variance was identified (Appendix D). The Eigenvalues of the three factors were 4.465, 1.695, and 1.102, respectively. The Kaiser-Meyer-Olkin measure of sampling adequacy was 0.777, and Bartlett's test of sphericity was significant (χ^2 (55) = 301.475, $p < 0.001$).

We labeled these factors as *Versatile*, *Obedient*, and *Determined performers*. The factor *Versatile performers*, characterized as "*we do what needs to be done*," contains the items that represent the students who stepped outside their designated Scrum roles (PP5) as well as their areas of expertise (PP6) if the situation required. The items loaded onto the second factor, *Obedient performers*, characterized as "*we do what we've been told*," most closely typify the guidelines of Scrum. The students performed in their designated roles (PP3), e.g., the Product Owner mostly worked in the interface between the client and the developer team, and the main responsibility of the Scrum Master was to facilitate the work of the other team members (Moe, Dingsøyr, and Dybå, 2010). The third factor, *Determined performers*, characterized as "*we do what we know*," characterizes the students to whom it appeared early in the project what each member of the team was supposed to do (PP7) and they worked on these areas throughout the project (PP8). The students performed according to their areas of expertise while not necessarily according to their designated Scrum roles (PP4). Finally, we reconstructed the previous correlation structure according to the underlying factors. Figure 4 presents Cronbach's α of each factor and the correlations between the final constructs.

Both constructs created from the items of EU correlated negatively with PQ. The correlation in both cases is moderate, yet it is slightly stronger between *Assignment-induced uncertainty* and PQ. Neither *Assignment-induced* nor *Change-induced uncertainty* correlated with the *Versatile performers*. The negative correlation between *Determined performers* and *Assignment-induced uncertainty* was significant, and a moderate negative correlation exists between both components of uncertainty and the construct of *Obedient performers*.

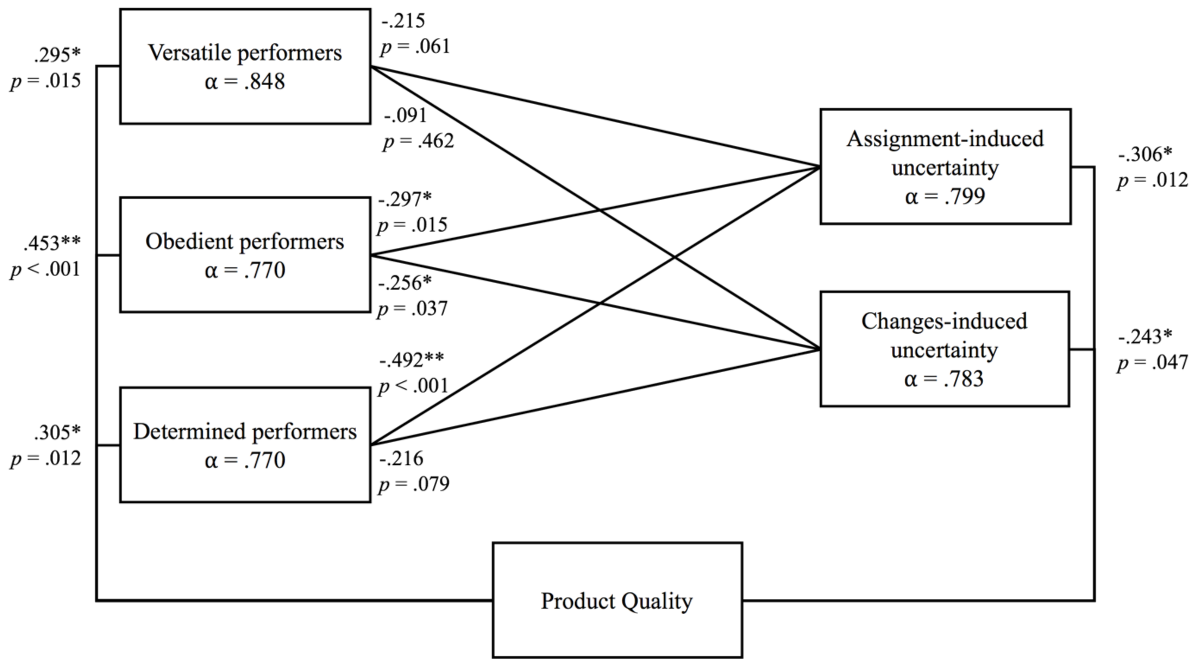

Figure 4. Final Construct Correlations – The Three Leftmost Constructs Created from the Items of PP and the Two Rightmost from EU

5. DISCUSSION

In this section, we discuss the implications of our results. Finally, we present the limitations of our study and address future research topics.

5.1 Implications of the Results

Our study showed similar relationships between uncertainty, process performance, and product quality as reported in previous studies (e.g., Jiang, Klein, and Discenza, 2001; Jun, Qiuzhen, and Qingguo, 2011; Wallace, Keil, and Rai, 2004). The key contribution of this study is the three strategies the student groups applied in the face of uncertainty. The correlation between *Assignment-induced uncertainty* and PQ was slightly stronger than the correlation between *Change-induced uncertainty* and PQ, which suggests the importance of identifying and tackling the possible sources of uncertainty early in the project life cycle and carefully managing the requirements (Ebert and De Man, 2005). Consequently, we suggest that teachers emphasize the importance of addressing emerging problems as early as possible.

The strong negative correlation between the constructs *Determined performers* and *Assignment-induced uncertainty* indicates that the students who knew what they are capable of did not experience the original project assignment as ambiguous as other students did. It is somewhat evident that self-assured developers are more capable of coping with ambiguous project assignments and making decisions even while facing uncertainty (Li et al., 2011). However, the correlation towards the construct *Changes-induced uncertainty* was not statistically significant. As these students mostly worked on their areas of expertise, it was coincidental whether said expertise could be effectively applied in a changed situation. Also, it is worth noting that regarding the construct *Versatile performers*, this study cannot reveal whether these students are flexible, multi-talents interested in various topics or if they rather reacted hastily to the approaching weekly sprint deadlines, allocated the immediate tasks at hand, for example via e-mail or instant messaging (PP1), and then did whatever was needed to complete the sprint's objectives. Regardless, unclear definition of roles and the situation where everybody is responsible for everything can be considered as an issue while using Scrum (Ayed, Vanderose, and Habra, 2014).

Overall, the construct *Obedient performers* seemed to present the best strategy to continuously cope with the project uncertainties. The students who rigorously followed the Scrum guidelines and practices were better equipped to deal with the changes in requirements and other sources of uncertainty. Each construct created from the items measuring the process performance correlated positively with the product quality. However, the construct *Obedient performers* had the strongest positive correlation with the product quality. Again, it appears that strictly following the Scrum guidelines yielded the best results for the students in our Agile project setting. While it is common to adapt a method by selecting and tailoring suitable practices and techniques (Ayed, Vanderose, and Habra, 2012) to reach a better fit with the project and organizational environment, this was out of the scope of the course because none of the students had previously used Scrum in practice. Consequently, we suggest that teachers emphasize the importance of following the chosen method, especially if the students are inexperienced, since according to Boehm and Turner (2004), Agile methods are typically better suited for experienced developers.

The students representing the construct *Obedient performers* had frequent face-to-face meetings (PP2) to plan the sprints, to play the Planning Poker (Haugen, 2006) to prioritize the sprint backlog items, to perform retrospective reviews, and to discuss acute and troublesome issues. In these groups, each member was continuously aware of the status of their project (PP9), they independently studied information needed to deal with new and changed requirements (PP11), and they did not hesitate to contact the client if something remained unclear (PP10). Furthermore, as the obedient performers performed most effectively, it suggests that we as teachers should emphasize the importance of the working methods of obedient performers for other students as well. Approaches such as PBL enable students to find, apply, and evaluate different coping strategies, and although it appears that these strategies produce different grades, trying out new strategies is also a part of the overall learning experience if the students can reflect on their experiences.

5.2 Limitations and Further Research

To allow anonymous responses, our survey was constructed so that we could not connect the respondents to their respective groups. Although the questions related to process performance asked the respondent to evaluate his or her Scrum team, it is possible that student groups were not homogenous in terms of *Versatile*, *Obedient*, and *Determined performers*. Second, our data collection instrument was rather simple, and the data was collected only at the end of the course as argued for in Section 4.1. We acknowledged several different sources of uncertainty identified in previous studies and implemented many of these in the course setup, yet the survey variables did not fully cover nor separate these sources, and, consequently, the analysis generalized uncertainty into two constructs. We examined uncertainty as a rather abstract concept instead of trying to understand its sources on a fine-grained level. These may be considered as limiting factors that should be addressed in further studies. Additionally, follow-up studies could reveal whether the students can make use of the lessons learned to better cope in uncertain environments in the future.

6. CONCLUSIONS

In summary, this study contributes to understanding how students' experienced uncertainty affects the process and product quality in an Agile system development course. The results also indicated three distinct strategies of working in uncertain environments: versatile, obedient, and determined. Organizational, business, and technical environments are complex, and software development teams can never predict and be fully prepared to meet all the possible contingencies and ambiguities. However, based on our study, and because uncertainty affects aspects such as risks and quality of the project (Geraldi, Kutsch, and Turner, 2011; Ward and Chapman, 2003), we suggest preparing the students for uncertainty through practical training and experiences by making the learning environments induce uncertainty. By making the learning environments more closely reflect real work environments, it can be ensured that the students do not face an uncertain situation for the first time when entering their first real development project.

7. REFERENCES

Ayed, H., Vanderose, B. & Habra, N. (2012). A Metamodel-Based Approach for Customizing and Assessing Agile Methods. *8th International Conference on the Quality of Information and Communications*. Lisbon, Portugal.

Ayed, H., Vanderose, B. & Habra, N. (2014). Supported Approach for Agile Methods Adaptation: An Adoption Study. *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering – RCoSE 2014*, 36-41.

Barki, H., Rivard, S., & Talbot, J. (1993). Toward an Assessment of Software Development Risk. *Development, 10*(2), 203-225.

Bell, S. (2010). Project-Based Learning for the 21st Century: Skills for the Future. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas,* 83(2), 39-43.

Boehm, B. & Turner, R. (2004). *Balancing Agility and Discipline*. Addison-Wesley.

Budner, S. (1962). Intolerance of Ambiguity as a Personality Variable. *Journal of Personality*, 30, 29-50.

Chu, W. H., Lin, D. Y., Chen, T. Y., Tsai, P. S., & Wang, C. H. (2015). The Relationships between Ambiguity Tolerance, Learning Strategies, and Learning Chinese as a Second Language. *System*, 49, 1-16.

Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., García-Peñalvo, F. J., & Tovar-Caro, E. (2013). Competence Gaps in Software Personnel: A Multi-Organizational Study. *Computers in Human Behavior*, 29(2), 456-461.

Darus, N., Mohd, H., Baharom, F., Saip, M., Puteh, N., Matt, Z., Husain, M., & Yasin, A. (2016). Factors Influencing a Problem-Based Learning Implementation: A Case Study of IT Courses. *AIP Conference Proceedings*, 176(1).

Dunaway, M. M. (2013). IS Learning: The Impact of Gender and Team Emotional Intelligence. *Journal of Information Systems Education*, 24(3), 189-203.

Ebert, C. & De Man, J. (2005). Requirements Uncertainty: Influencing Factors and Concrete Improvements. *ICSE'05*, 553-560.

Gallagher, S. A., Stepien, W.J., & Rosenthal, H. (1992). The Effects of Problem-Based Learning on Problem Solving. *Gifted Child Quarterly*, 36(4), 195-200.

Geraldi, J. G., Kutsch, E., & Turner, N. (2011). Towards a Conceptualization of Quality in Information Technology Projects. *International Journal of Project Management*, 29(5), 557-567.

Gorsuch, R. L. (1983). *Factor Analysis (2nd Ed.)*. Hillsdale, NJ: Erlbaum.

Groves, M. (2005). Problem-Based Learning and Learning Approach: Is there a Relationship? *Advances in Health Sciences Education: Theory and Practice*, 10, 315-326.

Hazzan, O. & Hadar, I. (2008). Why and how can Human-Related Measures Support Software Development Processes. *The Journal of Systems and Software*, 81(7), 1248-1252.

Haugen, N. C. (2006). An Empirical Study of Using Planning Poker for User Story Estimation. *Proceedings - AGILE Conference 2006*, 23-31.

Helle, L., Tynjälä, P., Olkinuora, E., & Lonka, K. (2007) 'Ain't Nothing Like a Real Thing'. Motivation and Study Processes on a Work-Based Project Course in Information Systems Design. *British Journal of Educational Psychology*, 77(2), 397-411.

Hmelo-Silver, C. (2004). Problem-Based Learning: What and how do Students Learn? *Educational Psychology Review*, 16(3), 235-266.

Jiang, J. J., Klein, G., & Discenza, R. (2001). Information System Success as Impacted by Risks and Development Strategies. *IEEE Transactions on Engineering Management*, 48(1), 46-55.

Joham, C. & Clarke, M. (2012). Teaching Critical Management Skills: The Role of Problem-Based Learning. *Teaching in Higher Education*, 17(1), 75-88.

Jun, L., Qiuzhen, W., & Qingguo, M. (2011). The Effects of Project Uncertainty and Risk Management on IS Development Project Performance: A Vendor Perspective. *International Journal of Project Management*, 29(7), 923-933.

Li, Y., Yang, M., Klein, G., & Chen, H. (2011). The Role of Team Problem Solving Competency in Information System Development Projects. *International Journal of Project Management*, 29(7), 911-922.

May, J., York, J., & Lending, D. (2016). Play Ball: Bringing Scrum into the Classroom. *Journal of Information Systems Education*, 27(2), 87-92.

Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A Teamwork Model for Understanding an Agile Team: A Case Study of a Scrum Project. *Information and Software Technology*, 52(5), 480-491.

Moynihan, T. (1996). An Inventory of Personal Constructs for Information Systems Project Risk Researchers. *Journal of Information Technology*, 11(4), 359-371.

Pigford, D. V. (1992). The Documentation and Evaluation of Team-Oriented Database Projects. *ACM SIGCSE Bulletin*, 24(1), 28-33.

Ross, J. & Ruhleder, K. (1993). Preparing IS Professionals for a Rapidly Changing World: The Challenge for IS Educators. *Proceedings of the 1993 Conference on Computer Personnel Research*, 379-384.

Schwaber, K. (2004). *Agile Project Management with Scrum*. Redmond, WA: Microsoft Press.

Taneja, A. (2014). Enhancing Student Engagement: A Group Case Study Approach. *Journal of Information Systems Education*, 25(3), 181-188.

Tavakol, M. & Dennick, R. (2011). Making Sense of Cronbach's Alpha. *International Journal of Medical Education*, 2, 53-55.

Turley, R. T. & Bieman, J. M. (1995). Competencies of Exceptional and Nonexceptional Software Engineers. *The Journal of Systems and Software*, 28(1), 19-38.

Tynjälä, P. (1999). Towards Expert Knowledge? A Comparison between a Constructivist and a Traditional Learning Environment in the University. *International Journal of Educational Research*, 31(5), 355-442.

von Glasersfeld, E. (1984). An Introduction to Radical Constructivism. In Watzlawick, P. (ed.) *The Invented Reality. How do we Know what we Believe to Know? Contributions to Constructivism*, New York: Norton, 17-40.

Wallace, L., Keil, M., & Rai, A. (2004). How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model. *Decision Sciences*, 35(2), 289-321.

Ward, S. & Chapman, C. (2003). Transforming Project Risk Management into Project Uncertainty Management. *International Journal of Project Management*, 26(1), 73-79.

West, D. & Grant, T. (2010). Agile Development: Mainstream Adoption has Changed Agility. *Forrester Research*.

AUTHOR BIOGRAPHIES

Toni Taipalus is a teacher at the University of Jyvaskyla. He

teaches databases, data management, application programming, and system development. His research interests are in the pedagogical aspects of query languages, data models, and Agile software development.

Ville Seppänen, Ph.D., is a teacher at the University of

Jyvaskyla. His current research interests are information systems education and enterprise architecture.

Maritta Pirhonen, Ph.D. (Econ.), is a teacher at the University

of Jyvaskyla. She has over 20 years of experience in using project-based learning in IS project management studies. Her research interests include ICT project management, specifically the role of the project manager within project success; and project-based learning in teaching.

Appendix A: Survey Instrument

All the questions below were answered on Likert scale.

Construct	Item	Question
Experienced Uncertainty (EU)	1	The project assignment was easy to understand.
	2	I could easily understand by the project assignment why the customer ordered this software.
	3	I could easily understand by the project assignment the customer's goals for the software.
	4	The software requirements were clearly expressed.
	5	The project assignment gave me enough guidance to start working.
	6	The changes in software requirements during the project were confusing.
	7	The changes in software requirements during the project complicated my work.
	8	The changes in software requirements affected negatively to the realization of the project assignment.
	9	The changes in software requirements made it difficult understand the project goals.
Process Performance (PP)	1	My Scrum team actively communicated using electronic communication channels during the project work.
	2	My Scrum team had frequent face-to-face meetings during the project work.
	3	My Scrum team members worked actively according to the tasks assigned to their Scrum role.
	4	My Scrum team members worked actively according to the tasks corresponding to their areas of expertise.
	5	My Scrum team members worked outside their designated Scrum role if needed.
	6	My Scrum team members worked outside their areas of expertise if needed.
	7	The roles of my Scrum team members were clear at the beginning of the project.
	8	The roles of my Scrum team members were clear at the end of the project.
	9	My Scrum team members were continuously aware about the status of the project.
	10	My Scrum team contacted the client to clarify the project assignment details if needed.
	11	My Scrum team searched actively information to cope with the new and changed requirements.
	12	I am satisfied with my own performance as a member of my Scrum team.
	13	I am satisfied with the performance of my Scrum team.
	14	I believe that I would have achieved better results by working in another Scrum team.

Appendix B: Survey Answers

	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
EU1	8 (11.94%)	35 (52.24%)	10 (14.93%)	12 (17.91%)	2 (2.98%)
EU2	1 (1.49%)	11 (16.42%)	5 (7.46%)	35 (52.24%)	15 (22.39%)
EU3	1 (1.49%)	14 (20.90%)	7 (10.45%)	33 (49.25%)	12 (17.91%)
EU4	3 (4.48%)	25 (37.31%)	9 (13.43%)	22 (32.84%)	8 (11.94%)
EU5	12 (17.91%)	21 (31.24%)	11 (16.42%)	17 (25.37%)	6 (8.96%)
EU6	2 (2.98%)	15 (22.39%)	8 (11.94%)	30 (44.78%)	12 (17.91%)
EU7	4 (5.97%)	20 (29.85%)	16 (23.88%)	19 (28.36%)	8 (11.94%)
EU8	7 (10.45%)	26 (38.80%)	15 (22.39%)	15 (22.39%)	4 (5.97%)
EU9	6 (8.95%)	24 (35.82%)	22 (32.84%)	12 (17.91%)	3 (4.48%)
PP1	3 (4.48%)	1 (1.49%)	3 (4.48%)	23 (34.33%)	37 (55.22%)
PP2	6 (9.00%)	8 (11.94%)	7 (10.45%)	26 (38.81%)	20 (29.85%)
PP3	9 (13.43%)	15 (22.39%)	6 (8.96%)	25 (37.31%)	12 (17.91%)
PP4	9 (13.43%)	15 (22.39%)	13 (19.40%)	15 (22.39%)	15 (22.39%)
PP5	1 (1.49%)	3 (4.48%)	4 (5.97%)	25 (37.31%)	34 (50.75%)
PP6	1 (1.49%)	4 (5.97%)	2 (2.98%)	18 (26.87%)	42 (62.69%)
PP7	6 (8.95%)	22 (32.84%)	3 (4.48%)	31 (46.27%)	5 (7.46%)
PP8	3 (2.99%)	16 (23.88%)	8 (11.94%)	24 (35.82%)	17 (25.37%)
PP9	12 (17.91%)	11 (16.42%)	5 (7.46%)	30 (44.78%)	9 (13.43%)
PP10	5 (7.46%)	9 (13.43%)	11 (16.42%)	24 (35.82%)	18 (26.87%)
PP11	3 (4.48%)	10 (14.93%)	9 (13.43%)	25 (37.31%)	20 (29.85%)
PP12	2 (2.98%)	7 (10.45%)	5 (7.46%)	27 (40.30%)	26 (38.81%)
PP13	5 (7.46%)	14 (20.90%)	5 (7.46%)	17 (25.37%)	26 (38.81%)
PP14	20 (29.85%)	15 (22.39%)	11 (16.42%)	12 (17.91%)	9 (13.43%)

Appendix C: The Factor Structure of Experienced Uncertainty

Item		Assignment-induced	Change-induced	Communality
EU1	The project assignment was easy to understand.	0.689		0.490
EU2	I could easily understand by the project assignment why the customer ordered this software.	0.731		0.538
EU3	I could easily understand by the project assignment the customer's goals for the software.	0.812		0.674
EU4	The software requirements were clearly expressed.	0.755		0.654
EU5	The project assignment gave me enough guidance to start working.	0.676		0.464
EU6	The changes in software requirements during the project were confusing.		0.748	0.562
EU7	The changes in software requirements during the project complicated my work.		0.834	0.705
EU8	The changes in software requirements affected negatively to the realization of the project assignment.		0.731	0.631
EU9	The changes in software requirements made it difficult understand the project goals.		0.737	0.607

Appendix D: The Factor Structure of Process Performance

Item		Versatile performers	Obedient performers	Determined performers	Com-munality
PP1	My Scrum team actively communicated using electronic communication channels during the project work.	0.751			0.689
PP2	My Scrum team had frequent face-to-face meetings during the project work.		0.769		0.629
PP3	My Scrum team members worked actively according to the tasks assigned to their Scrum role.		0.573		0.632
PP4	My Scrum team members worked actively according to the tasks corresponding to their areas of expertise.			0.631	0.635
PP5	My Scrum team members worked outside their designated Scrum role if needed.	0.884			0.796
PP6	My Scrum team members worked outside their areas of expertise if needed.	0.897			0.826
PP7	The roles of my Scrum team members were clear at the beginning of the project.			0.801	0.682
PP8	The roles of my Scrum team members were clear at the end of the project.			0.874	0.801
PP9	My Scrum team members were continuously aware about the status of the project.		0.701		0.603
PP10	My Scrum team contacted the client to clarify the project assignment details if needed.		0.633		0.416
PP11	My Scrum team searched actively information to cope with the new and changed requirements.		0.588		0.552