Diversity in Subject Content Transfer Styles Improves Software Engineers’ Analysis Skills

Maria-Isabel Sanchez-Segura1, Fuensanta Medina-Dominguez1, Ana Sanz2 and Javier Saldaña3
1Department of Computers Science, Carlos III Technical University, Madrid, Spain
2Softtek, Madrid, Spain
3Ayiming, Madrid, Spain

Abstract

The learning profile of each student must be taken into account to offer the learning resources that best suit him/her and thus can benefit him/her more. In software engineering field, the learning process can be improved if lecturers use multiple content transfer styles to teach the same concept. The authors provide a case study to demonstrate that diversity in content transfer styles oriented to teaching software analysis techniques allows lecturers to improve students’ analysis capabilities. To validate the hypothesis, a case study was developed using a control group which learned software analysis techniques using the traditional method based on slides; for the second group the lecturers provided different subject content transfer styles in order to observe the improvement in the learning process when students have a choice of content transfer styles. Statistic techniques were used to validate the results. Due to the results obtained and analyzed, the authors can affirm that diversity in subject content transfer styles improves software engineers’ analysis skills, reflected in the quality improvement of software products developed.

Introduction

Research into the way people learn is becoming more and more advanced. That is, research is showing that everybody has a learning profile which influences how they learn [1-6]. In addition, the advancement in information and communication technology (ICT) has changed the way teaching and learning are conducted, facilitated by the effective combination of different delivery modes, teaching methods and learning styles [7]. So, the combination of knowledge about how people learn and ICTs provide opportunities to create interactive student-centered learning environments, which are easily accessible and flexible [8]. Apart from technology, it is necessary to consider the learning profile of each student to offer the learning resources that best suit him/her and thus can benefit him/her more [9].

Considering that not everybody learns in the same way, it is advisable to use certain pedagogical elements according to each person’s learning profile. Training course material should not only focus on one type of educational resource, typically slides and bibliographic texts, but several tools must be provided to meet individual learning styles [10]. Therefore, it is necessary to integrate ICTs and pedagogical elements to design and develop training material [11].

This work focuses on the improvement of software engineers’ learning analysis process. For software engineers, analysis of the information system is one of the main processes that constitute the software development lifecycle. As a result of this process, a detailed software product requirement specification is developed [12]. The requirements specification must satisfy the user’s needs and forms the basis of designing the system [13]. Failures introduced by software engineers during the analysis phase in the software under development will persist throughout the product lifecycle, increasing the cost and time of rework to correct them when they are detected later [14]. So, the better the system requirements specification skills software engineers have, the fewer the mistakes made during the system analysis phase.

The reason why the authors focused on the learning process of analysis techniques is because, as explained before, it is a critical process in software development and the loss of requirements, which causes a lack of functionalities in the final product, is still a common problem [14]. This loss of requirements can be clearly identified in the relationship between use cases and requirements specifications. When software engineers define use cases, they often forget that each requirement has to be included in at least one use case description or scenario, resulting in a loss of requirements [15]. In an incremental-iterative use-case-driven lifecycle, this is currently more widespread; the final system will lose the functionality that has not been included in use cases [16].

The main goal of this paper is to improve software engineers’ capability to perform information system analysis tasks, more precisely, use cases and requirements specifications in order to decrease the percentage of volatilized requirements in software projects by developing a teaching-learning process that suits the profile of every software engineer; in other words, a process that takes into account the two key elements of learning mentioned previously: ICT and pedagogical factors. The degree of improvement in the teaching-learning process where each student can choose the format he/she is more comfortable with or feels that he/she can progress more quickly with will be evaluated.

This paper is organized as follows. Section 2 contextualizes this work describing existing learning theories and styles and several technology systems which support learning. Section 3 describes related works. Section 4 describes the method to improve the teaching-learning...
process and the case study which was carried out to test whether the personnel's training had improved. Also, at the end of this section, the authors present and analyze the results. Finally, section 5 presents the conclusions of this work.

**Context of this Work**

**Learning Theory:** Learning theories help to understand and predict human behaviour from the point of view of how knowledge is acquired. Their goal is to define theoretical models that specify the human learning process. These theories, however, do not provide a solution; they only draw lecturers' attention to essential variables. Some learning theories are cognitive information processing [22], cooperative learning [23], objectivism [24], sociocultural Learning [25], constructivism [26] etc. The most recent one, constructivism [26], establishes that students have to build new knowledge based on previously acquired knowledge [18], instead of memorizing everything as proposed in other theories. Constructivism was selected for this work because it best fits the way software engineers think and learn. For the constructivist, knowledge is not the result of a copy of reality; it is an interactive dynamic process through which the individual interprets and reinterprets external information, progressively building more complex and powerful explanatory models [27].

**Learning style:** Learning style is defined as the way a person perceives, processes, integrates and remembers information; that is, behaviours that indicate how to learn [28]. Students have different levels of motivation, attitudes, and learning styles [29]. Knowledge of learning styles can be used to increase self-confidence and awareness of strengths and weaknesses [30].

Several learning styles have been defined to support students' needs. Table 1 summarizes the most widespread learning styles and their characteristics.

<table>
<thead>
<tr>
<th>Learning Style</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canfield’s Learning Styles Inventory</td>
<td>Generic Education</td>
</tr>
<tr>
<td>The Canfield Learning Styles Inventory (CLSI) is a 50-item assessment using a 4 point rank order procedure for each item. Students rank these choices in order that best describes their preferences or reactions. A ranking process is used to obtain the raw scores. Thus, the lower the score, the stronger the preferences [31].</td>
<td></td>
</tr>
<tr>
<td>Dunn and Dunn Learning Style Inventory</td>
<td>Generic Education</td>
</tr>
<tr>
<td>Learning strategies are included in the methods thorough which teachers teach and/or learners learn. Methods and strategies which match the different types of learners are, for example, contract activities packages (CAP), program learning sequences (PLS) and multisensory instructional packages (MIP) [32].</td>
<td></td>
</tr>
<tr>
<td>Kolb’s Experiential Learning Model</td>
<td>Generic Education</td>
</tr>
<tr>
<td>Kolb classified students into four types: having a preference for concrete experience or abstract conceptualization (how they take information in), and active experimentation or effective observation (how they process information) [33]. Preferences on this scale are assessed with the Learning style inventory [34] or the Learning type measurement [35].</td>
<td></td>
</tr>
<tr>
<td>Honey and Mumford’s learning style model</td>
<td>Business and Management studies</td>
</tr>
<tr>
<td>Based on Kolb’s theory, this model identifies four types of students: reflector, theorist, pragmatist and activist [36]. It was developed in an attempt to apply learning style theory to business and management studies measurement [37]</td>
<td></td>
</tr>
<tr>
<td>Felder-Silverman Learning Style Model (SPAL method)</td>
<td>Engineering students</td>
</tr>
<tr>
<td>It was designed to capture the most important learning style differences among engineering students, and provide a good basis for engineering instructors to formulate a teaching approach that addresses the learning needs of every student [38]. The model classifies students according to the following four dimensions: sensitive or intuitive, visual or verbal, active or reflective, and sequential or global [38]. This model has been already implemented in software engineering courses and it can be tested using different technologies.</td>
<td></td>
</tr>
<tr>
<td>Keefee’s Learning Style Profile</td>
<td>Middle and senior high school students</td>
</tr>
<tr>
<td>The Learning Style Profile (LSP) was designed to give teachers an easy way to determine the learning styles of middle and senior high school students. LSP diagnoses students’ cognitive styles, perceptual response tendencies, and study instructional preferences by means of 23 variables [39]</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Characteristics of learning styles.
Related Works

The importance of considering different learning methods has been widely studied [46]. Murrel and Claxton [47] cited several studies in this area that form the basis for all subsequent research on learning styles. Dewey [48] pointed out that students learn best if you include a component of experience in the learning process. Similarly, Lewin [49], found that an active learning environment plays an important role.

Several studies confirm the relationship between learning styles and academic achievement as a result of students' response to different teaching methods [46]. Different researchers have found evidence to show that current information using different approaches lead to more effective instruction [50]. The overview of studies on academic performance and learning styles is very extensive: analysis of academic performance, in general; relation to learning styles; relation among learning styles; teaching methods; methods and academic performance, etc. Most of the works establish a relationship between the influence of learning styles and student performance [50-57]. After reviewing different studies Alonso et al. [Alonso, 99] concluded that it seems well established that students learn more effectively when learning styles are adapted to their needs. According to Figueroa [58], in engineering classrooms, students are extremely practical, oriented toward facts and procedures and prefer visual presentation of material, so the lecturer should empower these dimensions. However, as Alonso [55] pointed out, there is great difficulty in adapting teaching to learning styles. Not only must students' learning styles be taken into account but also the teaching style of lecturers.

This is why the authors of this paper have defined and validated a method to adapt a learning style, with a set of specific content transfer styles and integrated it into lecturers’ way of teaching.

Adapting Learning Material To Student Needs: A Case Study

The goal of this case study is to apply a method that takes into account both pedagogical and technological perspectives to make knowledge transfer more efficient. As mentioned in the previous section, the SPAL [38] learning style was selected to develop this case study. SPAL defines the sequence of activities that should be carried out by students and lecturers to achieve course objectives. This model was used to create learning resources to transfer subject contents, called "content transfer styles", to represent different learning resources being used: slides, electronic process guidance (EPG), wikis, videos, and interactive exercises.

These content transfer styles were selected for two reasons:

- they are mentioned in the work of Caper Jones [56] as one of the major channels for software personnel (people involved in the software development process) to acquire new information, evaluated in terms of cost, learning efficiency, learning effectiveness and currency information.
- they comply with the features of SPAL, which requires that the content transfer styles must be intuitive, reflective, sequential, verbal and visual (Table 2).

<table>
<thead>
<tr>
<th>Content transfer styles designed and used in this case study</th>
<th>Content transfer style description</th>
<th>Content transfer style example</th>
<th>SPAL method student traits covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slides</td>
<td>Slides with information about the topic used by the lecturer in class.</td>
<td><a href="http://seldata.sel.inf.uc3m.es/docs/mdv/CasosDeUso.pdf">http://seldata.sel.inf.uc3m.es/docs/mdv/CasosDeUso.pdf</a></td>
<td>Intuitive Reflective Sequential Visual</td>
</tr>
<tr>
<td>EPG</td>
<td>Electronic Process Guidance that describes the software development process to be followed by students.</td>
<td><a href="http://dhipe.sel.inf.uc3m.es/PFC-guiaprogprocess/WS_AnalisisOrientadoObjetos.htm">http://dhipe.sel.inf.uc3m.es/PFC-guiaprogprocess/WS_AnalisisOrientadoObjetos.htm</a></td>
<td>Active Intuitive Reflective Sensitive Sequential Visual</td>
</tr>
<tr>
<td>Videos</td>
<td>For each class there was a video which shows the previous class covering the same topic.</td>
<td><a href="http://www.youtube.com/watch?v=OTxi3aMXTVA&amp;list=UUD-BQ8n7isuihmHbIZK6T6Q&amp;index=10&amp;feature=plcp&amp;noredirect=1">http://www.youtube.com/watch?v=OTxi3aMXTVA&amp;list=UUD-BQ8n7isuihmHbIZK6T6Q&amp;index=10&amp;feature=plcp&amp;noredirect=1</a></td>
<td>Global Intuitive Reflective Sensitive Sequential Visual</td>
</tr>
<tr>
<td>Wiki (product patterns)</td>
<td>A wiki with the description of every product pattern that could be used by students.</td>
<td><a href="http://productpatternswiki.sel.inf.uc3m.es/mediawiki/index.php/Metodo_de_Craig_Larman">http://productpatternswiki.sel.inf.uc3m.es/mediawiki/index.php/Metodo_de_Craig_Larman</a></td>
<td>Active Global Intuitive Reflective Sensitive Visual</td>
</tr>
<tr>
<td>Interactive Exercises</td>
<td>Exercises which allow students to test their knowledge.</td>
<td><a href="http://seldata.sel.inf.uc3m.es/docs/mdv/CasosDeUso.htm">http://seldata.sel.inf.uc3m.es/docs/mdv/CasosDeUso.htm</a></td>
<td>Active Global Intuitive Reflective Sequential Visual</td>
</tr>
</tbody>
</table>

Table 2: Content transfer styles designed and used in this case study.
The description of student traits in the SPAL learning style appears next:

1. Active: learns by trying things out, enjoys working in groups.
3. Intuitive: abstract thinker, innovative, oriented toward theories and underlying meanings.
4. Reflective: learns by thinking things through, prefers working alone or with a single familiar partner.
5. Sensitive: concrete thinker, practical, oriented toward facts and procedures.
6. Sequential: linear thinking process, learns in small incremental steps.
7. Verbal: prefers written and spoken explanations.
8. Visual: prefers visual representations of presented material, such as pictures, diagrams and flow charts.

**Definition of the Case Study**

The case study was carried out in a software engineering training course in the last year of a large state university in Madrid, Spain. This course is addressed to improving software engineers’ abilities to perform software engineering tasks, specifically those developed in the analysis and design phases. The authors focused on the analysis phase because the goal of this case study is to improve the training of future software engineers in the system analysis phase, more precisely, in the use of techniques; use cases and requirements specifications. The achievement of this goal will be demonstrated by a decrease in the percentage of volatilized requirements in software projects and improvement in the quality of the products obtained by the participants. The use cases technique was selected because it is a very widespread technique, common to all development paradigms. Use cases are a key element and if there is a breakdown between system requirements and use cases, then the system will lose a part of its functionality, specifically the one related to the requirements lost when use cases were modelled. Improvement in the software engineers’ training was carried out by means of different subject content transfer styles instead of restricting the learning process to just one style.

To develop the case study, two software projects were developed consecutively from September 2013 to February 2015 in order to observe the improvement in software engineers’ analysis capability. Each software project developed corresponds to an iteration of the case study (two iterations in total). Table 3 summarizes the main features of each iteration.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Subject content transfer styles used</th>
<th>Number of students involved</th>
<th>Project developed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration 1</td>
<td>Slides</td>
<td>36 (12 groups, 3 students each)</td>
<td>Project 1: Development of an application that allows the management of the family account.</td>
</tr>
<tr>
<td>September 2013 to February 2013</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iteration 2</td>
<td>EPG, Exercises, Interactive Slides, Videos, Wiki (product patterns)</td>
<td>30 (10 groups, 3 students each)</td>
<td>Project 2: Development of an application that allows to manage bibliographical resources.</td>
</tr>
<tr>
<td>September 2014 to February 2015</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The authors considered the analysis phase in the two software projects in Table 3, with similar features and complexity, performing the same set of software product development activities. The same lecturers trained the participants in the two projects. The way in which the set of activities proposed by the authors was developed on each iteration is summarized in Table 7, section 4.2.

It is important to highlight that, as this is a real case study, there were different students on each iteration, but similar sample sizes were maintained and the students on each iteration had the same profile, knowledge level and capabilities. Before starting the subject the lecturers developed a test to determine the students’ knowledge of a set of concepts studied in previous years. The distribution of the grades in the samples is always the same, so on both iterations there is a similar distribution of student with grades between 5 to 7, 7.1 to 9, and 9.1 to 10... This is how the lecturers maintained the variable related to the students’ capabilities. Also, although the students were not the same on each iteration, two different practicals were prepared, one for each iteration in order to prevent students from sharing their practicals across iterations. Each practical corresponded to a software project developed on each iteration.

The training phase, where students learn the subject content, was different on each iteration. Therefore, in the first iteration the training was performed in the traditional way (on-site classes, class materials such as PowerPoint slides), making this the control group in this case study. In the second iteration, the content transfer styles used were slides, EPG, videos, wikis and interactive exercises. Each student chose the means that best suited his/her learning characteristics. Although this paper focuses on requirements specification and use cases techniques, it is important to point out that complete software projects were performed and they lasted 6 months on average. Of this time, one month was devoted to defining requirements specifications and use cases; for this purpose two one-and-a-half-hour training sessions were carried out weekly.

The parameters studied, being directly related to the previously stated objectives, improvement of the students’ performance of use cases and requirements analysis task were:

- Volatility of requirements: percentage of coverage of requirements regarding use cases. Traceability matrix between requirements and use cases are analyzed in order to identify the percentage of requirements that do not trace with any use case.
- Number of reviews performed to achieve the required quality level: number of reviews necessary to perform the software requirement specification until the required quality level is achieved.

![Table 3: Main features of each iteration.](image-url)
As a secondary aim, the authors wanted feedback on the participants' satisfaction when they were able to choose the way they learnt requirements analysis.

- Participants’ satisfaction: Degree of satisfaction of software engineers who took part in the project during the second iteration (2014-2015).

The following section presents a description of activities, defined in The Student Preference Adapted Learning Method, that the authors propose in this work. The activities describing the method followed can be applied to any area, but they have been illustrated for the specific case of software engineering students. First, the activities of the proposed method to be developed on each iteration of the case study are described in section 4.2. After, Table 7 shows the specific way in which each activity was carried out on each iteration.

Description of activities developed

**Activity 1: Elaborate learning resources**

The first activity consists of designing and elaborating learning resources to be used in the training course to improve the knowledge acquisition process.

Learning resources influence actively the process of student learning. These resources facilitate knowledge transfer, so they should be elaborated with special care and concern. It is important that they motivate students and not have the opposite effect.

In this activity, the material used to implement the course was created when none existed, or the existing material was updated. Table 4 summarizes the actions and results obtained from the execution of activity 1.

**Activity 2: Select technology**

In order to support the knowledge acquisition process it is recommended to use a technological platform that facilitates a collaborative and interactive learning environment. Each time this activity is performed the technology must be reviewed to check that it complies with the established requirements. Table 5 shows a summary of actions and results obtained from the execution of this activity in this case study.

**Activity 3: Establish course goals and schedule**

The third activity consisted of establishing the course goals. Before the course began, the lecturer established the course goals, determined the learning resources to be used, planned scheduled contents and established the method to assess students. Table 6 shows the goals identified for this case study in which the method defined in this work was applied as well as an excerpt from the schedule where the kind of tasks and their sequences are shown. Goals are common for the three iterations and the example corresponds to iteration 1.

**Activity 4: Select teaching-learning strategy**

The strategy used for the classes is described below.

Step 4.1: The lecturer explains concepts that software engineers can learn how to apply the knowledge to solve problems and teaches them how to use the learning resources.
Step 4.4: The software engineers develop a project assigned by the lecturers that covers the development of use cases and requirements specification techniques. This work focuses on analyzing the improvement obtained in the analysis phase. Although the students developed the entire project, this paper only considers products related to software requirements and use cases.

Activity 5: Assess degree of learning

The lecturer assesses the software engineers work developed, bearing in mind several quality criteria previously defined. The assessment is carried out in a continuous way, taking into account products obtained in each phase of the software development lifecycle contemplated in the deployment of the method proposed by the authors. In the context of the case study developed and reported in this paper, the products under consideration are requirements specification and use cases.

Table 7 shows the differences between activities developed in the iterations described in this work.

Analysis of results

The results obtained from the experiment were analyzed from three perspectives: volatility of requirements, reviews needed to achieve the quality level required and, as a collateral effect, the participants’ level of satisfaction using the content transfer styles proposed.

Volatility of requirements

The percentage of requirements that are not related to any use case was calculated for every project, considering the average rate each team achieved. Therefore, the traceability matrix between requirements and use cases was performed for these, which allows to detect requirements that do not trace with use cases; these are lost requirements. An T-test and the representation of the interval plot were used to analyze data.

- Data Analysis of Iteration 1 versus Iteration 2

H0: number of volatility requirements in both iterations is the same.
Iteration 1: number of volatility requirements in iteration 1.
Iteration 2: number of volatility requirements in iteration 2.

With a p-value of 0.009, lower than 0.05, H0 can be rejected. As Figure 1 shows, the number of volatility requirements was reduced when the content transfer styles proposed were used.

To conclude, it can be said that the use of the content transfer styles provided improved their analysis capability because the number of requirements that are lost when use cases were modeled decreased.

Reviews needed to achieve the level of quality required

The quality of requirements is a significant factor that affects the success of a project. A suitable software requirement specification will contribute to improving the resource and time planning, and the development of the system. Therefore, it is necessary to ensure that requirements are clear, precise, unambiguous and complete. By means of this variable, the number of reviews that were necessary to carry out until the software requirement specification fulfilled the level of quality required was analyzed. The quality criteria used to validate the software requirements specification are accessible at: http://seldata.sel.inf.uc3m.es/docs/LearningExperiment/criteria/Criteria.pdf. A T-test and the representation of the interval plot were used to analyze the data.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity 1</td>
<td>The effort was addressed to develop and improve the slides used in the master classes. Product patterns and the EPG began to be developed.</td>
<td>Pedagogical material was reviewed and updated. Further, interactive exercises were created.</td>
</tr>
<tr>
<td>Activity 2</td>
<td>The technology selection process was carried out.</td>
<td>The technology was reviewed.</td>
</tr>
<tr>
<td>Activity 3</td>
<td>This activity is performed in the same way in each iteration.</td>
<td></td>
</tr>
<tr>
<td>Activity 4 Step 4.1</td>
<td>This activity is performed in the same way in each iteration.</td>
<td></td>
</tr>
<tr>
<td>Activity 4 Step 4.2</td>
<td>With the objective of reinforcing the knowledge acquired, several optional exercises were proposed to be solved at home.</td>
<td>With the objective of reinforcing the knowledge acquired by students, several optional exercises were proposed to be solved at home. These exercises were interactive in order to increase students’ motivation to perform them.</td>
</tr>
<tr>
<td>Activity 4 Step 4.3</td>
<td>This activity is performed in the same way in each iteration. The only difference is the explanation lectures gave for the learning resources to be used on each iteration.</td>
<td></td>
</tr>
<tr>
<td>Activity 4 Step 4.4</td>
<td>Students developed a case study in groups using slides and the recommended library resources.</td>
<td>Students developed a case study in groups. For this purpose all the resources were available: slides, EPG, product patterns, videos, texts and library resources.</td>
</tr>
<tr>
<td>Activity 5</td>
<td>The lecturer assesses students. For this task, the lecturer grades the work and the students’ presentation.</td>
<td>The lecturer assesses students. For this task, he/she gives a mark to the works and carries out an individual exam in order to identify if students have achieved the learning goals previously defined. To prepare the exam, the resources used during the work (slides, library resources, the EPG, videos, product patterns, interactive exercises and library resources) were available.</td>
</tr>
</tbody>
</table>

Table 7: Summary of activities and how they have been developed on each iteration.
With a p-value of 0.004, lower than 0.05, H0 can be rejected. As can be seen in Figure 2, the number of reviews needed to achieve the quality required in systems requirements is lower in iteration 2 when the software engineers used the content transfer styles proposed.

To conclude, it can be said that the use of the proposed content transfer styles improves the quality of the system requirements because the number of reviews needed to achieve the quality required decreased.

- Data Analysis of Iteration 1 versus Iteration 2

H0: number of reviews needed to achieve the level of quality required in both iterations is the same.

Iteration 1: number of reviews needed in iteration 1.
Iteration 2: number of reviews needed in iteration 2.
In addition, the reduction in the number of reviews needed to reach the required quality of the system requirements using the proposed content transfer learning styles positively affects the verification and validation process because the time and effort and, in consequence, the cost of this process is reduced.

Participants’ satisfaction

In the second iteration, a survey was carried out in order to assess participants’ satisfaction as regards content transfer styles used in the course. This information was addressed to determine the usability of such resources.

Figure 3 shows that the differences between valuations obtained for resources are not significant. Therefore, no resource was valued negatively or with a really low mark. This means that the software engineers considered that these tools helped them positively to improve their skills in software requirements and use cases. Moreover, the fact that all the resources got similar marks proves that there are several learning styles and therefore every individual requires the pedagogical resources that best suits his/her learning profile. Thus, the usability of each resource depends on the software engineer and his/her learning style, but there is no evidence that a resource is not usable or better than the rest. It depends on the software engineer’s preferences.

Threats to Validity of the study

The major internal validity threats are listed here.

Related to the influence of teachers, in both iterations there were two teachers but they were different in each iteration. The teachers who taught in the iteration 2 did not participate in iteration 1. The only common person in both iterations was the coordinator of the subject, who did not teach class in any of the iterations and was not communicating with students. It is the reason which authors can affirm that there was no influence by teachers in the outcome of the data related to the common material in both iterations, the authors can ensure that the slides, which are the only common material in both iterations, were not improved nor updated; teachers used the same slides in both iterations.

Conclusions and Future Works

Requirements are one of the key elements to develop software systems in planned time and with required quality levels. These must be clear, precise, unambiguous and complete. However, the low quality of requirements and their loss are common problems. One way to solve these problems is to improve software engineers’ qualification through the use of multiple content transfer styles. This work proposes to develop training materials which integrate both pedagogical and technical perspectives to improve software engineers’ capability to perform information analysis tasks. The authors have developed a case study where the statistical analysis of its results reveal that the percentage of volatilized requirements decreased and their quality increased in the software analysis phase.

Due to the fact that each individual has a different learning style, the authors have developed a method which involves the use of several learning resources in the format of content transfer styles so that each software engineer can select the one which best suits his/her profile. To assess the improvement with this method, a case study was carried out and the results of the different iterations were analyzed.

As a conclusion of the results, the percentage of volatility requirements decreased as well as the number of reviews to achieve
the level of quality required, which means that the system analysis tasks were performed better, using the different content transfer styles proposed by the authors.

This means that not only has the requirement quality increased but also the verification and validation process has improved due to the time reduction.

It worth's the cost of producing the additional material involved in the development of the case study not only because the results of the case study are positive but also because they can be reused on following academic years.

The students valued content transfer styles according to their learning profile. Therefore, it can be concluded that all the content transfer styles have contributed to the software engineers' learning, one cannot be classified as better than the others because all of them have been useful, depending on the learning style of the student.

This work is focused on Use Cases and requirements specifications in order to improve analysis skills. However, it is possible to replicate this experiment in other development and management processes in the software engineering field and even in engineering areas in general.

Competing Interests

The authors declare that they have no competing interests.

References

5. Dihua, X (2012) Personalized Learning Path Recommender Based on User Profile Using Social*: Fifth International Symposium on Computational Intelligence and Design (ISCID), 511 - 514.


