Assessment of continuing educational measures in software engineering: A view from the industry

Olaf Radant

BearingPoint Germany

Kurfürstendamm 207-208, 10719 Berlin, Germany

olaf.radant@bearingpoint.com

Ricardo Colomo Palacios

Østfold University College, Norway

B R A Veien 4, 1783 Halden, Norway

ricardo.colomo-palacios@hiof.no

Vladimir Stantchev

SRH Hochschule-Berlin

Ernst Reuter Platz 10, 10587 Berlin, Germany

vladimir.stantchev@srh-hochschule-berlin.de

Abstract

In a field like Software Engineering, highly dependent on knowledge, continuing education is key to warrantee business sustainability. In this scenario, counting on with continuing educational measures is crucial to support organizations in their competence management programs. In order to provide with this set of measures, in this paper an interview study regarding continuing education measures is reported. The study was carried out in late 2014, early 2015 with some of the most experienced SE professionals from the consulting company BearingPoint. Study findings suggest an increasing relevance of such educational measures. Furthermore, authors provide insights about relevant educational measures, improvement areas in employees' training, commonly experienced problems, as well as instruments for control and evaluation.

Keywords

Software Engineering, Educational Measures, Continuing Educational, Training

1. Introduction

The software industry is a powerful wealth creator, generating billions in revenues, millions of jobs, countless new companies and innovative business models (Slaughter, 2014). In spite of its importance and overall impact, as any other industry in the world, it is facing threats and challenges. Forced to be in the

loop of continuous innovation (R. Colomo-Palacios, García-Crespo, Soto-Acosta, Ruano-Mayoral, & Jiménez-López, 2010; García-Moreno et al., 2013; Hernández-González, García-Moreno, Rodríguez-García, Valencia-García, & García-Sánchez, 2014), one of this challenges is the scarcity of various goods and resources (Booch, 2009). Given that software development is highly intensive in human capital, the key factor for industry is personnel. Not in vain, software development is a human centric and sociotechnical activity influenced by personnel factors (Ricardo Colomo-Palacios, Casado-Lumbreras, Misra, & Soto-Acosta, 2014).

Maybe the biggest issue the industry is facing is the shortage of IT professionals all over the world, which has been pointed out by many works and reports e.g. (Mithas & Krishnan, 2008). The problem is rooted on the erosion of its student base (Hirschheim & Newman, 2010; Sabherwal, 2010), on the retirement of babyboomers (Stone & Deadrick, n.d.) and on the relative low success of initiatives like Global Software Development (Casado-Lumbreras, Colomo-Palacios, Ogwueleka, & Misra, 2014; R. Colomo-Palacios, Casado-Lumbreras, Soto-Acosta, Misra, & García-Peñalvo, 2012). Thus, in a perspective of demographic change skilled software practitioners will continue to be hard to find due to decreasing birth rate (Radant, 2014). While systematic resource governance approaches (Vladimir Stantchev, Petruch, & Tamm, 2013) and project portfolio management paradigms (Vladimir Stantchev & Franke, 2009; Vladimir Stantchev, Franke, & Discher, 2009) can provide higher utilization of available human resources, this provides only a limited amelioration of the resource problem in the field.

In any case, apart from the problem of manpower, there is also an issue in the scarcity of specialized knowledge and competence in software industry as many studies have underlined (Casado-Lumbreras, Colomo-Palacios, Hernández-López, & Soto-Acosta, 2011; R. Colomo-Palacios, Casado-Lumbreras, Soto-Acosta, García-Peñalvo, & Tovar-Caro, 2013; Ricardo Colomo-Palacios, Casado-Lumbreras, Tovar, & Soto-Acosta, 2011; Ricardo Colomo-Palacios, González-Carrasco, López-Cuadrado, Trigo, & Varajao, 2014; R. Colomo-Palacios, Tovar-Caro, García-Crespo, & Gómez-Berbís, 2010; Ruano-Mayoral, Colomo-Palacios, García-Crespo, & Gómez-Berbís, 2010; R. Valencia-García, García-Sánchez, Castellanos-Nieves, Fernández-Breis, & Toval, 2010). Skills obsolescence is especially important in a sector that witness fast paced technological, domain, and process changes leading to rapid skills obsolescence, unless these skills are updated often (Bapna, Langer, Mehra, Gopal, & Gupta, 2012). In other words, a constant development of the software practitioner's knowledge is fundamental as highlighted by previous works on the field (Agarwal, Pande, & Ahuja, 2014; Khemaja & Mastour, 2014). Main assets in software industry are not servers, buildings or machines. The main asset is knowledge capital. Due to the fluctuation of labour and the fact that available resources are not increasing along with the increasing needs, knowledge management, training and education in software engineering are even more important (Garcia-Alvarez, Suárez Álvarez, & Quiroga García, 2014; Rus & Lindvall, 2002).

Training and training measures (or continuing education and continuing educational measures) are part of the improvement programs in both small (Díaz-Ley, García, & Piattini, 2010) and big companies (Armbrust, Ebell, Hammerschall, Münch, & Thoma, 2008) and knowledge-based enterprises(V. Stantchev & Franke, 2010). However, to the best of authors´ knowledge and although the topic is quite popular in scientific literature, there is not a study devoted to identify these measures in organizational contexts. This paper is aimed to bridge this gap.

2. The study: Training measures in software engineering

The interview study reported here was carried out with some of the most experienced SE professionals from the consulting company BearingPoint. Six partners and senior managers from BearingPoint participated in the interviews. The sample consisted three Senior Managers and three Partners. Three of them worked as "Software Quality Assurance Engineers and Testers" and three of them as "Information Technology Project Manager". The mean age was 41.83 with a standard deviation (SD) of 6.74. They have work experience in Information Technology on average of 15.33 years with SD of 6.47 and experience in the field of software

engineering of 11.17 years with a SD of 5.91 years. The study was structured in two parts. In the first part, the interviewer hosts a round table which five of the six SE professionals attended. The goal was to identify the most common challenges for SE companies and continuing education and training. After that, the interviewer created four research questions, based on this event.

- RQ1: Which are the relevant educational measures and their benefits in praxis?
- RQ2: What could be improvements for companies regarding training of employees?
- RQ3: What are the common problems when companies are conducting these measures?
- RQ4: What are controlling instruments for these measures to evaluate the success?

In the second part, the interviews regarding the research questions were conducted and recorded by an interviewer, and later transcribed by him. The study took place as a qualitative interview study in the tradition of the qualitative research interview which allows the researcher to ask questions to different issues in the interviewees work life and experiences, including practical issues of how to do things and handle cognitive issues such as personal and professional epistemology (Sayrs, 1998). Open-ended questions were used and members of the team had freedom to describe at length their experiences. This data collection approach provides information that could not be obtained through a quantitative approach as it allows opinions, thoughts and feelings (Sayrs, 1998).

Atlas.ti 6 software was used for transcription and coding of the interviews. Subjects were selected from those who answered positively to a personal invitation sent by the authors. The total recorded time of interviews was 5 hours, 30 minutes with an average of 55 minutes and 3 seconds per interview.

2.1 Which are the relevant training measures and their benefits in praxis?

There are several types of trainings for IT- and software engineering employees. All of them have their specific elements and benefits (see Table 1). It is important to mention that the choice of which training type should be used, depends on the audience and their skills.

Type of training delivery	Elements	Intended audience and benefits
Instructor-led training (traditional classroom training)	– Classroom setting – For small audiences (up to 20 end users) – Provides walkthroughs – Storyboards and concept slides – Task-level work instructions – Quick reference guides – Hands-on exercises – Facilitator and learner guides – Allows end users to complete scenario-based, hands-on activities – Case studies – Simulations – Online performance support system	– Most appropriate for training groups on changes to concepts, procedures, and detailed functional and technical steps. – Provides participants with access to professional knowledge, allows participants to learn from each other, and emphasizes teamwork. – Excellent for communicating delicate topics. – Experience shows that using client resources to conduct the end-user training is instrumental in decreasing resistance and promoting adoption of new business processes.
Computer-based, web-based, and virtual training	– Self-paced and interactive learning process – Training accomplished by using a CD or via the internet – Online simulations – Exercises – Knowledge checks	– Self-paced – Online access to course materials – Hands-on practice – Useful for economical training geographically disbursed stakeholders – Facilitates course sessions over the intranet and internet – Useful when classroom training is not available or practical
Hands-on practice environment or sandbox	– Production-like environment that end users can use after training to	– Most appropriate for individuals who have completed the training course and would like to continue building their

Type of training delivery	Elements	Intended audience and benefits
	practice specific transactions and processes in the system – Environment uses simulated but realistic data – Self-directed learning	confidence using the new system or keep the knowledge and skills recently learned – Self-paced – Hands-on practice – Mirrors the production database – Safe environment in which to practice skills
One-on-one training	– On-the-job training – Personal coaching – Informal training	– Better suited for senior-level management – Highly customized to learner on-the-job presence – Provides prompt application of learned knowledge, skills or abilities – Feedback is extensive, contextual, and immediately relevant
Demonstrations and presentations	– Provides a general introduction to the system and components – Can be conducted in person or on the web – Employed prior to classroom training	– Ideal for large audiences – Cost-effective

Table 1 Types of Training and their corresponding Elements and Benefits

2.2 What could be improvements for companies regarding training of employees?

All experts stated that - due to the rapid change of requirements - learning and training are more important than ever for ensuring more effective and efficient operations in software engineering. This is particularly relevant in the context of societal and demographic changes. Companies need to develop and educate their employees in the best possible way, to retain and achieve the targeted level of production. In what follows, authors summarize the answers given by the participants.

It is important to achieve a certain basis of knowledge in a company on which the measures can build up on each other. So the first training for new acquired personnel should be a curriculum which teaches the standards and characteristics of the companies systems, processes and working habits. Afterwards specific trainings for the assigned tasks could be conducted, but it is always necessary to set standards for competence management and development. Regarding the organization of training, there should be one person who is in charge of the measures. Often, due to miscommunication between departments, trainings overlap not only in schedules but also in the skills taught. This can be prevented if a company uses a knowledge management strategy and a corresponding knowledge management system, which sets the standards, organizes and regulates the educational processes to improve the training time of the employees.

Further training for SE is often not provided with sufficient funding to conduct the necessary trainings nor to equip the labs with the adequate hardware. This is often due to lacking willingness of the leadership of a company to fund investments which have not a short-term business case. Also, the acquired skills and abilities should be used immediately after the training to receive the best possible outcome. Although the gained knowledge will not be lost if it is not applied shortly after the training, the learning process will indeed be far more successful this way.

2.3 What are the common problems when companies conducting these measures?

Risk	Mitigation
Lack of User Commitment to Training Activities	– Leadership must communicate that end-user training is mandatory for the required staff.

Risk	Mitigation
	- Class attendance will be tracked, and this data will be provided to the appropriate management staff. - The class schedule will be published well in advance of the training delivery date to reduce conflicts, personal commitments, and other scheduling limitations.
Users Do Not Have Prerequisite Knowledge and Have Skills Gaps	- Users should have basic PC and Windows knowledge before end-user training. This will be communicated to the appropriate management staff. - Managers should evaluate their user population to determine who needs additional PC and Windows training.
Technical Issues	- Software/database issues - A technical team member will be assigned to serve as a liaison with the Training Lead. - Hardware issues - Create a back-up plan in the event of projector/system issues (for example, a need for additional projector bulbs or an additional projector). - Provide hard copies of slides to end users.
Training Is Offered Too Far in Advance of Go-Live	- Instructor-led training (ILT) should ideally be conducted no more than 2 months prior to go-live (ideally 1 month prior to go-live) with other training activities potentially given throughout the project. - Timing of training delivery is important to increase retention of information learned in class and improve the trainees' ability to perform their jobs after go-live.

Table 2 Risks and Mitigation Strategies

The overall success of a software engineering department depends on the success of the training effort; processes and systems require knowledgeable people to realize planned benefits. While Table 2 summarizes common risks and corresponding mitigation measures, Table 3 presents a list of items which are critical to the success of the training effort, and actions that can be taken to ensure success.

Critical success factors for training	
Success factors	**Actions to ensure success**
Management commitment – Ensures the required time is committed on the part of employees to participate in required activities, training, etc.	– Top Management Commitment – Strong Leadership Communications – Engage supervisors to ensure priority – Consistency and visibility throughout the organization
Trainer commitment – employees selected as trainers will spend significant time and energy learning about the project and how to be effective trainers.	– Early identification of trainers – Engage immediate supervisors to ensure priority – Train- the-Trainer program provides training tools content and generates project excitement
User commitment – Users are ultimately accountable for their own training. Their quality of learning will depend on attitude and commitment.	– Consistent and visible communication – Training requirements communicated early – Engage supervisors to ensure priority
Knowledge sharing – Rapid development of training will rely on knowledge and information from the company team and Subject Matter Experts (SMEs).	– Establishing roles and responsibilities for training developers and core team – Identifying the best people as training developers
Process and System Design Decisions – Timely process and system design decisions with minor revisions. Significant and last-minute changes will reduce the effectiveness of training.	– Understanding the implications for training if/when changes are made – Close coordination with Functional Teams so changes are communicated quickly – Decision maker awareness of the implication of changes. – Communication of the Documentation Freeze Date.
Training environments – Access to necessary IT-environments during training development and delivery. Configuration and data must be consistent with the production environment. Refresh schedule	– Properly configured and data populated training environments – Refresh schedules established – Sandbox available to all users

Critical success factors for training	
Success factors	**Actions to ensure success**
must be adhered to and practice sandbox with realistic data should be available.	
Facilities and infrastructure – Training requires dedicated facilities and IT systems. All sites must have adequate network connectivity.	– Close coordination within the department – Early communication of requirements to sites – Adequate equipment in training rooms – On-call IT and Training Team support
PC workstations – End users must have PC workstations available that are properly configured in a timeframe that coincides with training delivery.	– Early identification and communication of requirements to the Technical Team
Technical support – Training delivery concentrated in a short period requires minimal rescheduling due to technology and application failures.	– Pre-training checks of infrastructure operation – On-call technical and functional support
User training materials – Training materials must be user friendly.	– Minimize printed materials; – Documentation needs to be simple and straight forward

Table 3 Critical Success Factors for Training

2.4 What are controlling instruments for these measures to evaluate the success of them?

Training instruction and content will be measured through an evaluation process. The primary use of training evaluation will be to support continuous improvement of training materials and instructional approaches. A strategy for evaluating the effectiveness of the training should be developed early in the design phase (Hettiarachchi, Huertas, & Mor, 2015). The approach could include four levels of evaluation (see Table 4):

Training Program Evaluations – Course participants evaluate every course taught in the program. The purpose of the evaluation is to gain feedback that can be used to improve subsequent training sessions. The evaluation provides trainees with an opportunity to indicate their level of satisfaction with the following:

- Training materials
- Course content
- Course design
- Facilities

Classroom Performance (participant) Evaluations – Trainee evaluations will measure retention via assessments, hands-on exercise completion, and case study understanding. During training, exercises will be structured so that each module builds on the previous module. A user cannot progress through the class unless skills are mastered in each module. Trainers can assess each participant based on his or her progress through the modules.

Behaviour Evaluation - This measures how well the participant is applying the new skills on the job (the degree to which training has changed job behaviour) and is collected after the participant has actually used the new skills on the job. Behaviour Evaluation can be linked to the individual's personal development plan.

Business Results Evaluation – It measures the impact of the training on the overall business unit performance and is collected after the participant has worked in the new environment for approximately nine to 12 months. A baseline for comparison will be needed for this evaluation to be effective.

Measures and methods of training evaluation		
Evaluation	Measures	Method of evaluation
1. Program (overall training experience)	– Training administration process (registration, facilities) – Trainer performance	– Course evaluation – Trainer debrief

Measures and methods of training evaluation		
Evaluation	Measures	Method of evaluation
	– Course material (quality, content and presentation of)	
2. Participant (classroom performance)	– Based on the predefined learning objectives which measure understanding of: o Facts – such as the new process and procedures o Techniques – such as how a transaction is performed to support my job role	– Course evaluation – Trainer evaluation
3. Behaviour (job performance)	– Behaviour (e.g. willingness to perform job role following processes and procedures) – Performance (e.g., the ability to use the new system effectively with few errors)	– Participant post training survey – Supervisor post training survey – Help desk trouble calls
4. Business Results (business unit performance)	– Improved quality – Reduced costs – Increased productivity	– Supervisor post training survey – Budget/ reports

Table 4 Measures and Methods of Training Evaluation

3. Methodology for the implementation and execution of training measures in software engineering

Also the expert panel suggested a rigid development methodology for the implementation and execution of continuing educational measures which follows a five steps approach (Figure 1).

Figure 1 Approach for the Implementation and Execution of Continuing Educational Measures

Conduct training analysis: The development and delivery of training must integrate a number of key elements. Before the training solutions are developed or delivered, a thorough requirements assessment and audience analysis will be conducted using skills assessments, job impact assessment, process redesign, and fit/gap analysis. Prerequisite role definitions, types of training, and new course development (for example, as is, from scratch, or off the shelf) are some of the issues that will be identified at this time.

Design training: The objectives of developing a training plan are to produce trainings which fit the strategic development and the needed skills of the company. To achieve the desired goals, the implementation of a flexible, blended training delivery solution that considers various training delivery mechanisms was also recommended.

Develop training: During this phase the required business, technical, and end-user training materials are build. This may include developing classroom training and train-the-trainer presentation materials, graphics, interactive elements and post-training job aids. When appropriate these materials will be business scenario based and product centric.

As part of the complete training plan for software engineering, detailed specifications for a training environment should be included, but not be limited to the following:

- Hardware architecture,
- Software architecture,
- Training configuration.

An effective training environment is crucial to an effective training. A simulated production-like environment or sandbox will contain the same functionality, views and reporting structure planned for the

production client. Thus it will mirror the real-life situations the trainees might encounter with new projects. This practice training environment will also enable users to practice running transactions and process scenarios during and after classroom training.

Deliver training: The experience of the expert panel has shown that a blend of training modes, such as instructor-led training (ILT) and computer-based training (CBT), enhances workforce learning and performance improvement. Instructor-led classes will include just-in-time training and frequent hands-on learning. Self-paced, technology-based training modules offer hands-on practice in the form of simulations, exercises, and knowledge checks. Although, different training strategies for different audiences are appropriate. In addition a variety of performance support resources for reinforcement, such as quick reference guides, desktop user guides, and online access to course materials should be added.

It was highly recommended, to provide a "dry run" of each class, manual and material, instructions and lesson plans to evaluate the success or implement necessary adjustments. Training aids, appropriate system manuals, quick reference guides or templates and other training materials should be provided for each participant in the training. Also, a train-the-trainer approach that uses an organization's own in-house resources to conduct end-user training and support has been instrumental not only in decreasing resistance and promoting adoption of the proposed solution and business processes, but also in enhancing knowledge transfer and self-sufficiency.

Conduct evaluation and maintenance: Critical to the ongoing training program success is confirming that employees in various roles have the necessary knowledge, skills, and abilities to perform their new role functions. In post-implementation, users will require ongoing support and additional training as they work to sustain changes and challenges. New employees will need to learn and understand the systems and processes. Proficiency requires more than attending a one-time learning event held prior to go-live. Research indicates that learning occurs not only through the acquisition of knowledge, but also through the successful application of that knowledge on the job (Mosheiov & Sidney, 2003). Ongoing operational support is just as critical to effective performance as the learning event itself. During this phase, the development of a method to assess the effectiveness of the initial training, which identifies strengths and the need for improvement of the plan to sustain training on an ongoing basis should be conducted.

The overall goal for training is effectiveness which can be measured in the following seven areas:

- Targeting -End users will understand the courses and be able to schedule appropriate courses.
- Effectiveness -End users who complete required training will be able to successfully perform their duties.
- User accountability - End users who complete training will be able to continue practicing and learning in their own environments and will know how to get additional help and support.
- Participation - Training participation will be recorded and users will be randomly tested prior to Go-Live to ensure successful knowledge transfer.
- Availability -Training will be delivered on schedule. Facilities, equipment, and systems will be available when needed.
- Customer satisfaction -Training efforts will focus on the needs of the end users being served and they will have a positive perception of their training experience.
- Cost effectiveness - Training will be delivered on time and in budget

The following Table 5 outlines and summarizes the key training activities and deliverables associated with each phase.

Phases	Activities	Deliverables
Analysis/Strategy	– Develop a concise strategy that involves the following key activities: – Identify and tailor approaches to be used in developing and delivering required training such as e-learning, self-study, and classroom options – Perform high-level affected population analysis – Identify facilities and infrastructure requirements by location – Evaluate and recommend training delivery approaches	– Training plan
Design/Design	– Develop a tactical plan that describes the development and delivery of training (how, what, when, where, and by	– Training plan

Phases	Activities	Deliverables
	whom). Develop training content outlines help the workforce learn new skills, technologies, and processes needed to bridge required competency gaps. Key activities associated with this deliverable are as follows: – Incorporate appropriate processes and procedures into each course – Incorporate course mapping to job role grouping – Develop room, trainer and class schedules – Develop course material templates and prototypes – Identify course evaluation tools	
Development/ Build	– Create courseware and develop training courses. Select trainers for instructor-led classroom delivery (train the trainer). Key activities associated with this deliverable are as follows: – Create training materials and exercises – For each master course, develop an instructor guide, participant guide, quick reference guide, and competency testing criteria – Assist in the design of the training technical environment	– Training plan project team training materials (project development team training)
Deliver/Deploy	– Begin knowledge transfer of new related content to the workforce to bridge the gap between the present level and the desired level of skills and behaviours. Key activities associated with this deliverable are as follows: – Conduct train-the-trainer sessions for each course – Develop training schedule and instructor assignments – Register training participants – Support training delivery through go-live – Deliver training – Evaluate and document training results	– User Training – Training classes, class feedback reports, training report for end users
Evaluation and Maintenance/ Operate	– Present summary of the learning metrics activities associated with this deliverable are as follows: o Collect and analyse learner evaluation data o Use instructor feedback o Review training program	– Final report on training effectiveness

Table 5 Phases, Activities and Deliverables of Training Development

After training needs are identified and solutions are designed and developed, the training team will work with their staff to support the just-in-time delivery of required training. The training will be delivered based on the pre-defined roles, thus enabling standardization across the organization. This standardization of roles, processes and training will help:

- Drive the initiative across multiple departments or stakeholder groups
- Reduce or eliminate stovepipe processes
- Reshape the workforce to leverage the new procedures, processes, and technologies

4. Discussion and limitations

Software development is crucial for every company in every industry and for the people itself (Hernandez-Lopez, Colomo-Palacios, & García-Crespo, 2013; Rashid & Roni, 2012; Siddiqui, Hussain, & Hussain, 2006). In a permanently changing business environment, companies and especially their IT-Departments have to adapt to changes in the market and be more agile and customer-oriented than ever before (Stoica, Mircea, & Ghilic-Micu, 2013). Due to a lack of proper techniques and a resulting lack of productivity within the organizations, millions of dollars are wasted (Cao, Gu, & Thompson, 2012; Sasankar & Chavan, 2011). Because of the increasing complexity of new developed software and a needed specialization of employees,

productivity of software development companies became more and more important issue in the last years and for the future (Corbin, Dunbar, & Zhu, 2007). Thus, training in SE engineering is one of the most important challenges for companies because it has to cover a wide range of possible issues and fields to cope with and fulfil the demands of the market (Liebowitz & Agresti, 2011).

The results of the study should be considered from two different points of view. At first, this study investigated some of the most common challenges for SE companies regarding the training and continuing education of their employees. The identified and currently relevant training methods in the area of SE are mostly consistent with the identified measures in the literature. Although a purely technical or functional training is considered insufficient especially in the international context of global software development and learning by doing approaches are more promising and thus trainings are nearly irrelevant (Aurum, Daneshgar, & Ward, 2008), the interviews showed, that formal trainings are a very important source of knowledge for employees, because they set the foundation to gain knowledge on an everyday basis. The authors would rather agree with the findings in the interview. Formal trainings are indeed relevant and set the basis for a further acquisition of knowledge as is informal training (García-Peñalvo, Colomo-Palacios, & Lytras, 2012; García-Peñalvo & Conde, 2014; Rafael Valencia-García, García-Sánchez, Casado-Lumbreras, Castellanos-Nieves, & Fernández-Breis, 2012). Additionally continuing educational measures should include a soft skills curriculum (Conn, 2002; Holtkamp, Jokinen, & Pawlowski, 2015; Sousa & Mouraz, 2014).

Also the authors suggested a standard process, which should help companies to gain a better understanding how trainings are executed from the design until the controlling of the success of the trainings. Similar approaches were published before for several specific requirements (Green, 2005; Kuhrmann, Mendez Fernandez, & Munch, 2013), (Vladimir Stantchev & Tamm, 2011). Due to a wide range of specific needs in software engineering, authors propose a more general methodology for the implementation and execution of training measures in software engineering. Companies can adopt this methodology and alter it to their actual requirements and situation. In connection with the other findings like risks, risk mitigation strategies and success factors authors are confident that companies can benefit from this methodology.

The main limitation of our study is the sample size. Although it is an introductory investigation, it uses a small sample size and, thus, conclusions and implications may not be broadly generalized. Future studies could include a more representative sample of the total population of the target group or the whole industry. Another limitation comes from the fact that all participants are experts from Germany, may exhibit regional or national bias, and that regional conditions are not completely considered.

5. Conclusions and future works

Successful, efficient and effective training needs to be standardized and centrally organized. If a company wants to achieve this, they need to anchor education of SE employees as one of the main strategic pillars of the organization. Furthermore, standard processes for the development of training, as proposed in this paper, are indispensable. Also adequate financial resources for the training departments are a non-negligible factor for success.

Continuing education of employees doesn't always promise a short term improvement of a company's performance. Thus, a mid- and long term evaluation of the controlling ratios is needed. But it can provide significant improvements in the areas of quality and other key performance indicators (KPIs). A one sided consideration of the investments while neglecting the possible improvements and advantages should be avoided. For further research, authors recommend a comparison of the amounts invested for staff training in SE companies with the corresponding corporate results. Thus it may be possible to identify a sector-specific correlation factor between the invested amounts in training and business results. Also common knowledge management methods in praxis should be reviewed to investigate how they can help a SE company to reduce its investments in training due to a better flow of information and knowledge.

REFERENCES

Agarwal, N., Pande, N., & Ahuja, V. (2014). Expanding the Kirkpatrick Evaluation Model-Towards more Efficient Training in the IT Sector: *International Journal of Human Capital and Information Technology Professionals*, *5*(4), 19–34. http://doi.org/10.4018/ijhcitp.2014100102

Armbrust, O., Ebell, J., Hammerschall, U., Münch, J., & Thoma, D. (2008). Experiences and results from tailoring and deploying a large process standard in a company. *Software Process: Improvement and Practice*, *13*(4), 301–309. http://doi.org/10.1002/spip.391

Aurum, A., Daneshgar, F., & Ward, J. (2008). Investigating Knowledge Management practices in software development organisations – An Australian experience. *Information and Software Technology*, *50*(6), 511–533. http://doi.org/10.1016/j.infsof.2007.05.005

Bapna, R., Langer, N., Mehra, A., Gopal, R., & Gupta, A. (2012). Human Capital Investments and Employee Performance: An Analysis of IT Services Industry. *Management Science*, *59*(3), 641–658. http://doi.org/10.1287/mnsc.1120.1586

Booch, G. (2009). Software Abundance in the Face of Economic Scarcity, Part 1. *IEEE Software*, *26*(5), 12–13.

Cao, Q., Gu, V. C., & Thompson, M. A. (2012). Using Complexity Measures to Evaluate Software Development Projects: A Nonparametric Approach. *Engineering Economist*, *57*(4), 274–283. http://doi.org/10.1080/0013791X.2012.729878

Casado-Lumbreras, C., Colomo-Palacios, R., Hernández-López, A., & Soto-Acosta, P. (2011). Personnel Performance Appraisal Coverage in ITIL, COBIT and CMMi: A Study from the Perspective of People-CMM. *International Journal of Knowledge Society Research*, *2*(2), 59–70.

Casado-Lumbreras, C., Colomo-Palacios, R., Ogwueleka, F. N., & Misra, S. (2014). Software Development Outsourcing: Challenges and Opportunities in Nigeria. *Journal of Global Information Technology Management*, *17*(4), 267–282. http://doi.org/10.1080/1097198X.2014.978626

Colomo-Palacios, R., Casado-Lumbreras, C., Misra, S., & Soto-Acosta, P. (2014). Career Abandonment Intentions among Software Workers. *Human Factors and Ergonomics in Manufacturing & Service Industries*, *24*(6), 641–655. http://doi.org/10.1002/hfm.20509

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. http://doi.org/10.1016/j.chb.2012.04.021

Colomo-Palacios, R., Casado-Lumbreras, C., Soto-Acosta, P., Misra, S., & García-Peñalvo, F. J. (2012). Analyzing human resource management practices within the GSD context. *Journal of Global Information Technology Management*, *15*(3), 30–54.

Colomo-Palacios, R., Casado-Lumbreras, C., Tovar, E., & Soto-Acosta, P. (2011). A step towards human capital management in the software industry based on generic competencies. *International Journal of Strategic Change Management*, *3*(4), 247 – 259. http://doi.org/10.1504/IJSCM.2011.044938

Colomo-Palacios, R., García-Crespo, Á., Soto-Acosta, P., Ruano-Mayoral, M., & Jiménez-López, D. (2010). A case analysis of semantic technologies for R&D intermediation information management. *International Journal of Information Management*, *30*(5), 465–469.

Colomo-Palacios, R., González-Carrasco, I., López-Cuadrado, J. L., Trigo, A., & Varajao, J. E. (2014). I-Competere: Using applied intelligence in search of competency gaps in software project managers. *Information Systems Frontiers*, *16*(4), 607–625. http://doi.org/10.1007/s10796-012-9369-6

Colomo-Palacios, R., Tovar-Caro, E., García-Crespo, Á., & Gómez-Berbís, J. M. (2010). Identifying technical competences of IT Professionals: the case of software engineers. *International Journal of Human Capital and Information Technology Professionals*, *1*(1), 31–43.

Conn, R. (2002). Developing software engineers at the C-130J Software Factory. *IEEE Software*, *19*(5), 25–29. http://doi.org/10.1109/MS.2002.1032849

Corbin, R. D., Dunbar, C. B., & Zhu, Q. (2007). A three-tier knowledge management scheme for software engineering support and innovation. *Journal of Systems and Software*, *80*(9), 1494–1505. http://doi.org/10.1016/j.jss.2007.01.013

Díaz-Ley, M., García, F., & Piattini, M. (2010). MIS-PyME software measurement capability maturity model – Supporting the definition of software measurement programs and capability determination. *Advances in Engineering Software*, *41*(10–11), 1223–1237. http://doi.org/10.1016/j.advengsoft.2010.06.007

Garcia-Alvarez, M. T., Suárez Álvarez, E., & Quiroga García, R. (2014). ICTs and Learning: A Challenge in the Engineering Education. *International Journal Of Engineering Education*, *30*(3), 636–643.

García-Moreno, C., Hernández-González, Y., Rodríguez-García, M. Á., Miñarro-Giménez, J. A., Valencia-García, R., & Almela, A. (2013). A Semantic based Platform for Research and Development Projects Management in the ICT Domain. *Journal of Universal Computer Science*, *19*(13), 1914–1939.

García-Peñalvo, F. J., Colomo-Palacios, R., & Lytras, M. D. (2012). Informal learning in work environments: training with the Social Web in the workplace. *Behaviour & Information Technology*, *31*(8), 753–755. http://doi.org/10.1080/0144929X.2012.661548

García-Peñalvo, F. J., & Conde, M. Á. (2014). Using informal learning for business decision making and knowledge management. *Journal of Business Research*, *67*(5), 686–691. http://doi.org/10.1016/j.jbusres.2013.11.028

Green, R. (2005). Effective Training Programs. *CADalyst*, *22*(3), 42–43.

Hernández-González, Y., García-Moreno, C., Rodríguez-García, M. Á., Valencia-García, R., & García-Sánchez, F. (2014). A semantic-based platform for R&D project funding management. *Computers in Industry*, *65*(5), 850–861. http://doi.org/10.1016/j.compind.2013.11.007

Hernandez-Lopez, A., Colomo-Palacios, R., & García-Crespo, A. (2013). Software engineering job productivity - a systematic review. *International Journal of Software Engineering and Knowledge Engineering*, *23*(03), 387–406. http://doi.org/10.1142/S0218194013500125

Hettiarachchi, E., Huertas, M. A., & Mor, E. (2015). E-Assessment System for Skill and Knowledge Assessment in Computer Engineering Education. *INTERNATIONAL JOURNAL OF ENGINEERING EDUCATION*, *31*(2), 529–540.

Hirschheim, R., & Newman, M. (2010). Houston, we've had a problem…… offshoring, IS employment and the IS discipline: perception is not reality. *Journal of Information Technology*, *25*(4), 358–372. http://doi.org/10.1057/jit.2010.23

Holtkamp, P., Jokinen, J. P. P., & Pawlowski, J. M. (2015). Soft competency requirements in requirements engineering, software design, implementation, and testing. *Journal of Systems and Software*, *101*, 136–146. http://doi.org/10.1016/j.jss.2014.12.010

Khemaja, M., & Mastour, T. (2014). Skill oriented Training Activity as a Service:: An Approach based on the e-Competence Framework to overcome the Fast Changing IT Profession. *International Journal of Human Capital and Information Technology Professionals*, *5*(4), 55–78. http://doi.org/10.4018/ijhcitp.2014100104

Kuhrmann, M., Mendez Fernandez, D., & Munch, J. (2013). Teaching software process modeling. In *2013 35th International Conference on Software Engineering (ICSE)* (pp. 1138–1147). http://doi.org/10.1109/ICSE.2013.6606665

Liebowitz, J., & Agresti, W. W. (2011). IT Education for Practitioners. *IT Professional*, *13*(6), 14–15. http://doi.org/10.1109/MITP.2011.101

Mithas, S., & Krishnan, M. S. (2008). Human Capital and Institutional Effects in the Compensation of Information Technology Professionals in the United States. *Management Science*, *54*(3), 415–428. http://doi.org/10.1287/mnsc.1070.0778

Mosheiov, G., & Sidney, J. B. (2003). Scheduling with general job-dependent learning curves. *European Journal of Operational Research*, *147*(3), 665 – 670. http://doi.org/10.1016/S0377-2217(02)00358-2

Radant, O. (2014). Demographic Change:: The Reasons, Implications and Consequences for IT Departments. *International Journal of Human Capital and Information Technology Professionals*, *5*(1), 41–54. http://doi.org/10.4018/ijhcitp.2014010104

Rashid, A. A., & Roni, R. A. (2012). Exploring the Approaches for COBIT Process in Malaysian 100 Top Corporate Governance Companies. Retrieved from http://www.internationalconference.com.my/proceeding/icber2012_proceeding/211_154_3rdICBER2012_Proceeding_PG3081_3106.pdf

Ruano-Mayoral, M., Colomo-Palacios, R., García-Crespo, Á., & Gómez-Berbís, J. M. (2010). Software Project Managers under the Team Software Process: A Study of Competences. *International Journal of Information Technology Project Management*, *1*(1), 42–53.

Rus, I., & Lindvall, M. (2002). Guest Editors' Introduction: Knowledge Management in Software Engineering. *IEEE Softw.*, *19*(3), 26–38. http://doi.org/10.1109/MS.2002.1003450

Sabherwal, R. (2010). Declining IS enrollments: a broader view of causes and strategies – a response to "Houston, we"ve had a problem … offshoring, IS employment and the IS discipline: perception is not reality'. *Journal of Information Technology*, *25*(4), 382–384. http://doi.org/10.1057/jit.2010.33

Sasankar, A. B., & Chavan, V. (2011). SWOT Analysis of Software Development Process Models. *IJCSI International Journal of Computer Science Issues*, *8*(5), 1694–0814.

Sayrs, L. (1998). Interviews: An Introduction to Qualitative Research Interviewing. *American Journal of Evaluation*, *19*(2), 267.

Siddiqui, M. S., Hussain, S. J., & Hussain, S. J. (2006). Comprehensive Software Development Model. In *Computer Systems and Applications, 2006. IEEE International Conference on.* (pp. 353–360). http://doi.org/10.1109/AICCSA.2006.205113

Slaughter, S. A. (2014). *A Profile of the Software Industry: Emergence, Ascendance, Risks, and Rewards*. (G. A. Atlanta, A. M. Costley, & G. I. of Technology, Eds.). Business Expert Press.

Sousa, A., & Mouraz, A. (2014). Promoting'Soft Skills' from the Start of the Engineering Degree and the Case Study of the Special'Projeto FEUP'Course. *International Journal of Engineering Education*, *30*(6), 1539–1548.

Stantchev, V., & Franke, M. R. (2009). Managing Project Landscapes in Knowledge-Based Enterprises. In *Best Practices for the Knowledge Society - Knowledge, Learning, Development and Technology for All; Proceedings of the Second World Summit on the Knowledge Society - WSKS 2009* (Vol. 49, pp. 208–215). Springer. Retrieved from http://www.springer.com/computer/general/book/978-3-642-04756-5

Stantchev, V., & Franke, M. R. (2010). Knowledge and learning aspects of project portfolio management. *International Journal of Knowledge and Learning*, *6*(2), 114–129.

Stantchev, V., Franke, M. R., & Discher, A. (2009). Project Portfolio Management Systems: Business Services and Web Services. In *ICIW '09: Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services* (pp. 171–176). Los Alamitos, CA, USA: IEEE Computer Society. http://doi.org/10.1109/ICIW.2009.32

Stantchev, V., Petruch, K., & Tamm, G. (2013). Assessing and governing IT-staff behavior by performance-based simulation. *Comput. Hum. Behav.*, *29*(2), 473–485. http://doi.org/10.1016/j.chb.2012.06.003

Stantchev, V., & Tamm, G. (2011). Addressing Non-Functional Properties of Services in IT Service Management. In *Non-Functional Properties in Service Oriented Architecture: Requirements, Models and Methods* (pp. 324–334). Hershey, PA, USA: IGI Global.

Stoica, M., Mircea, M., & Ghilic-Micu, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica*, *17*(4), 64–76.

Stone, D. L., & Deadrick, D. L. (n.d.). Challenges and opportunities affecting the future of human resource management. *Human Resource Management Review*. http://doi.org/10.1016/j.hrmr.2015.01.003

Valencia-García, R., García-Sánchez, F., Casado-Lumbreras, C., Castellanos-Nieves, D., & Fernández-Breis, J. T. (2012). Informal learning through expertise mining in the social web. *Behaviour & Information Technology*, *31*(8), 757–766. http://doi.org/10.1080/0144929X.2011.638397

Valencia-García, R., García-Sánchez, F., Castellanos-Nieves, D., Fernández-Breis, J. T., & Toval, A. (2010). Exploitation of social semantic technology for software development team configuration. *Software, IET*, *4*(6), 373 –385. http://doi.org/10.1049/iet-sen.2010.0043

Olaf Radant received a diploma in business administration (2009) from the University of Applied Sciences in Berlin, Germany. He works as a business consultant for the consulting company BearingPoint in the areas of strategy-, organizational development and business transformation. His research interests include demographic change, IT-Management and organizational development. He is a Ph.D. candidate in the Computer Science program at Universidad Carlos III de Madrid.

Ricardo Colomo-Palacios, Full Professor at the Computer Science Department of the Østfold University College, Norway. Formerly he worked at Universidad Carlos III de Madrid, Spain. His research interests include applied research in information systems, software project management, people in software projects, business software, software and services process improvement and web science. He received his PhD in Computer Science from the Universidad Politécnica of Madrid (2005). He also holds a MBA from the Instituto de Empresa (2002). He has been working as Software Engineer, Project Manager and Software Engineering Consultant in several companies including Spanish IT leader INDRA. He is also an Editorial

Board Member and Associate Editor for several international journals and conferences and Editor in Chief of International Journal of Human Capital and Information Technology Professionals.

Vladimir Stantchev is the executive director of the Institute of Information Systems at SRH University Berlin where he is a research professor. He is also an affiliated senior researcher with the Networking Group at the International Computer Science Institute (ICSI) in Berkeley, California, USA. Vladimir Stantchev studied law at Sofia University (Sofia, Bulgaria) and also earned his master's degree in computer science from the Humboldt-University in Berlin, Germany. He received his PhD (Dr. rer. nat.) in the area of system architectures from the EECS department of the Berlin Institute of Technology (TU Berlin). His major research interests are in the areas of IT-Governance, Cloud Computing architectures, IT strategy, as well as methods for service and software engineering.