Fraunhofer Institut
Experimentelles
Software Engineering

Eight Years of Delivering Professional Education and Training for Software Engineering at Fraunhofer IESE: An Experience Report

Authors:
Ludger Thomas
Patrick Waterson
Sonja Trapp

Submitted for publication at
CSEET

IESE-Report No. 092.05/E
Version 1.0
October 2005

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the Fraunhofer Gesellschaft.
The institute transfers innovative software development techniques, methods and tools into industrial practice, assists companies in building software competencies customized to their needs, and helps them to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach (Executive Director)
Prof. Dr. Peter Liggesmeyer (Director)
Fraunhofer-Platz 1
67663 Kaiserslautern

Abstract

In this paper, we reflect on our experiences of delivering professional software engineering (SE) education and training over the course of the last eight years. We begin the paper with a summary of current developments in SE education and training, followed by a brief description of the educational framework that has guided our work in this domain. We then move on to describe four case studies of SE education and training delivery together with as set of lessons learnt. We end the paper with a summary of the wider lessons learnt gained from our experience in the domain, these consider how SE education should be delivered and facilitated, as well as other considerations such as changes to organizational roles and responsibilities brought about by the introduction of technology-based learning.

Keywords: software engineering education, education, professionalism, e-learning

Table of Contents

1	**Education and Training in Software Engineering**	**1**
2	**A Framework for Professional Learning, Education and Training in SE**	**2**
3	**Four Case Studies of Professional Learning**	**4**
3.1	Workplace-integrated learning: The APO Project	4
3.2	Blended Learning: Teaching UML	5
3.3	Communities of Practice: The "Software Competence Centre"	6
3.4	Collaborative Learning: The CORONET-Train System	7
4	**Conclusions: Putting it all together – what have we learnt?**	**9**
4.1	Adapt methods and media used for delivering training to the needs of SE professionals	9
4.2	Facilitate professional SE education in day to day work	10
4.3	Wider considerations and the big picture	11
References		**12**

1 Education and Training in Software Engineering

Over the last few years, many efforts have been undertaken to define and harmonize the structures and contents of software engineering (SE) education at undergraduate and graduate level (e.g., [1][1][13]). Most of these contributions were motivated by problems or perceived gaps within SE curricula, as well as external pressures such as the growing demand for well-educated software engineers. Others (e.g., [16]) have argued that current approaches rely too much on traditional as compared to engineering-focused approaches and do not equip students and learners with essential non-technical alongside technical skills and knowledge (e.g., team working and communication skills). Perhaps the most damaging criticisms are those that argue that many courses prepare students to work as scientists but not as engineers (e.g., [11]).

Whilst debates regarding the nature of undergraduate and postgraduate SE Education have proved popular amongst academics, corresponding discussions centred on topics such as continuing professional training and the lifelong formal/informal learning of software professionals, are less frequent. On the one hand this isn't surprising given the fact that Universities rely heavily upon such courses for incomes and as a source of prestige. On the other hand, it has long been that claimed that approximately 70% of professional skills are gained informally (e.g., [8][5]), for example, through reading off-line and on-line materials (articles, books, information available from newsgroups or the web more generally) and face-to-face interaction (e.g., communicating with colleagues). In this paper we attempt to partially redress the balance by describing the experiences we have gained over the last eight years working in this area and providing education and training for work-based software professionals. We first review the educational framework that has guided our work before moving on to describe four case-based examples and the lessons we have learnt from them. We end the paper with a summary of our experiences and the larger lessons they have taught us, the purpose of which is to act as guidance for similar workers and researchers in the area.

2 A Framework for Professional Learning, Education and Training in SE

Over the years there have been many attempts to describe the essential character of learning and the various cognitive, social and technological factors that promote and foster learning (e.g., [2][4]). Educational theorists consistently point to five essential components of successful learning (Figure 1). Firstly, effective learning is only possible through active involvement of the learner (i.e., the learner takes over responsibility and organized and controls their learning process) Secondly, learning should be constructive, that is it should build on existing knowledge structures and experiences. Thirdly, learning primarily takes place in social settings and fourthly, the learner's context and situation play a major role. Finally, learning is strongly influenced by performance-related and social emotions, the emotional component being very important in regard to motivation for example.

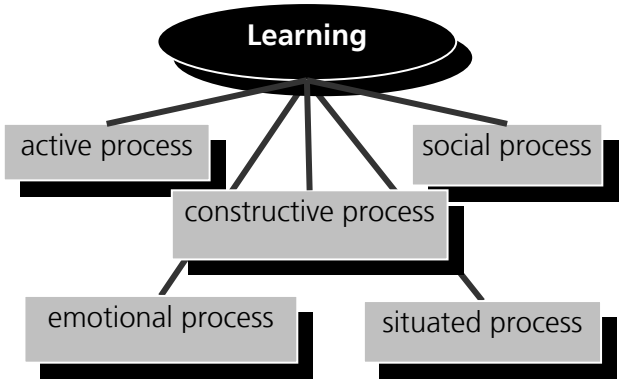

Figure 1: Five characteristics of learning

Many influential theories and approaches are built around these components; for example, the Cognitive Apprenticeship approach [6] stresses the importance of situated learning, social context, intrinsic motivation and cooperation for learning. The approach is oriented around the idea of a master helping a set of students to become masters themselves. The master provides initial guidance, especially in the early phases of the learning process, on how to solve real world problems and encourages students to cooperate, discuss, reconsider and share their knowledge. It also emphasises the importance of situated and active learning that help learners to transfer their skills and competencies to new situations. Supporters of the work-based learning take a similar philosophy of learning and education, emphasizing the following characteristics of learning which are embedded within work processes [15] :

- Learning is acquired in the midst of action and dedicated to the task at hand;
- Knowledge creation and utilization are collective activities wherein learning becomes everyone's job; and,
- Learners demonstrate a "learning-to-learn" attitude, which frees them to question underlying assumptions of practice.

In the context of our work on training and education for software professionals we have tried to use the framework and its components to guide the design of professional learning for SE. In addition, the framework has acted as an aid with which to interpret the results from the various projects and systems we have worked on over the years. We return to this framework in section 4 of the paper following a description of four case studies based on our work.

3 Four Case Studies of Professional Learning

3.1 Workplace-integrated learning: The APO Project

The APO-IT[1] project in Germany attempted to combine the definition and certification of Information and Communications Technology (ICT) related job profiles with so-called workplace-integrated learning. Following frequent demands from ICT business, that employees should be able to handle complex processes, work in interdisciplinary teams, cope with changing tasks and show customer-oriented thinking and acting, this ambitious approach aims at improving not only ICT competencies, but also social and managerial skills. The method relies on the concept of the reflective practitioner, mixing on-the-job learning with elements from formal learning. While formal training makes only subject-matter knowledge available, workplace-integrated learning helps to acquire hands-on process competence. Trainees remain at the workplace, they are involved in real projects, and acquire technical as well as process knowledge. This phase of informal learning is being supervised by a coach who helps trainees to critically reflect on their learning process, to write down what they have learned and how they learned it, which problems they encountered and what kind of material they used for acquiring expertise. The documentation process is central to this kind of informal learning, as it provides the basis for later formal certification.

First results showed that this method empowers individual learners and leads them to better overview complex tasks by themselves, they are enabled to analyse and evaluate own experiences, identify their own knowledge gaps, plan and organise operations, they acquire and consolidate knowledge and learn how to express and document results. The qualification procedure becomes significantly more structured and allows learners to overview the broader context. They get insights into ICT systems as a whole and acquire higher-level on-the-job competences. To sum up the experiences:

- Work-based learning results in more sustainable learning effects than many other kinds of trainings. This is partly due to the fact that it appeals to a high level of self-learning competence and self-discipline, and leads to a strengthening of methods and process expertise. Trainees take over responsibility for their own learning process and for the work process, they gain self-confidence and are prepared for further change and learning processes that may come up in their professional future.

[1] In German APO stands for "Arbeitsprozessorientierten Weiterbildung" (Work Process-Oriented Further Education)

- The most challenging task for the trainees, who in most cases do not have an academic background, seems to produce the documentation of their own learning process in relation to the respective work process.
- Moreover, work-based learning requires time and effort and can only succeed if trainees receive help and encouragement through individual coaching and if they are strongly supported by management.

3.2 Blended Learning: Teaching UML

Blended Learning is not a clearly defined concept but a synonym for a mixture of various methods that mainly involve self-paced e-learning and traditional face-to-face training. One of the main motivations to mix methods is that people tend to feel isolated and miss social contact in pure self-paced e-learning scenarios and hence the drop-out rate is high. In one of our projects with a large German company, we designed and implemented a blended learning course that was made up of face-to-face elements, self-paced learning using a web based training, together with individual face-to-face coaching. The goal of this collaboration was to help the company's software developers to get familiar to the paradigm of object orientation and to spread the use of software engineering principles as well as the Unified Modeling Language (UML) within the companies software development process (the course is available at http://www.uml-kurs.de/) [5]

One of the activities that were carried out during the project was an analysis of the usage of the self-paced e-learning component. The results showed that the contents are heavily used, especially at the end of a working day (around 4 pm) and on Fridays. One of the main conclusions from the study was that learning time does not appear to conflict with normal work and, in general, self-paced learning helps to optimize the co-existence of working and learning [5]. The course has been delivered several times to different groups at the company; the main lessons learnt so far include:

- We observed that the blended learning approach is widely accepted among the participants. The course has proved to be popular within the company and over time more and more people have registered to take the course. Informal interviews with participants, and more general feedback from the company, indicate that blended learning is efficient in terms of changing learner behavior, especially when it is enriched with additional transfer supporting activities, such as individual coaching and consulting. One interesting finding is that course participants take more responsibility for their learning and educational needs over time.
- There is a great demand for examples and course material that makes use of information that can easily be integrated with routine work tasks and is not solely based on theory or from existing textbooks. In building and extending

the course we have come to appreciate the need to enrich self-paced learning with specific transfer supporting actions that can be adapted to a specific domain and individualized to the learners day-to-day work. Based on our experience, such transfer-oriented efforts help the participants to apply the new knowledge more easily.

- In self-paced scenarios special attention has to be paid to learner's motivation. It is very important to provide further support in the application of the new knowledge and encourage learners to try out their knowledge in new situations (i.e., encourage them to transfer their knowledge). Furthermore, motivation is increased when the results of completing the course are acknowledged and recognized within the company by some form of certification. The full support from the management and the provision of success stories and a trustworthy "champion" is also vital in ensuring that motivation levels are raised and can be sustained over time.

3.3 Communities of Practice: The "Software Competence Centre"

The third case study involves a German project known as the Software Competence Centre (http://www.software-kompetenz.de/). The overall aim of the project is to bring together industrial and research-based professional groups in software engineering (SE) and encourage technology transfer between the research and industrial groups. The portal currently contains more than 3500 pages covering most of the SWEBOK areas, each of them e.g. containing knowledge, experiences and literature. A survey carried out in June/July 2003 showed that 60% of respondents worked for small or medium enterprises in the software industry. The survey also indicated that the majority of visits to the portal were for the purpose of finding out more about SE techniques, tools and methods, alongside other searches for information on trends within SE. In addition to the portal related activities, the project initiated several local communities of software engineers to support the transfer and application of SE research results and know-how into industry, particularly small- to medium-sized German companies [12]. Most of the regional communities were formally launched in 2003 after the knowledge base of the portal has been set up. Currently each of them provides several events a year, each attended by about 40 people. Experience with the online and regional communites so far has led to a number of lessons learnt, these include:

- At the launch of the community it was hoped that members would spontaneously volunteer information and the community would grow in an "organic" manner. In reality, persuading users to contribute information (e.g., experience reports, empirical data) has proved to be a major challenge throughout the existence of the community. As a result the community is largely "passive" in the sense that users tend to retrieve information rather than actively contribute. Nevertheless, individual learning is one of the moti-

vations for the participation in the communities and users tend to use the community to acquire new knowledge and ideas informally.

- Perhaps the most important lesson that was learnt is that communities-of-practice in a domain such as software engineering require a large amount of re-sourcing at the beginning and, needless to say, a good deal of effort has to be placed on maintenance activities in order to sustain interest (e.g., keeping content up-to-date). Whilst this may appear obvious, there is a widespread tendency to assume that general SE communities are similar to other forums and communities of practice (e.g., open source communities) where the motivation to provide and make use of information may be very different and the focus of the community, and therefore its functioning may be much harder to specify and manage [17] [14].

3.4 Collaborative Learning: The CORONET-Train System

The final case study is an example of a collaborative learning methodology (CORONET-Train) that was developed within a large-scale European research project CORONET (Corporate Software Engineering Knowledge Networks for Improved Training of the Workforce). The overall goal of the project was to facilitate web-based collaborative work-based learning amongst software engineers. The main components of CORONET-Train consist of a collaborative learning infrastructure in the form of a web-based learning platform, together with a set of methods that are used to facilitate collaborative learning and are implemented within the platform. These include: learning methods that define and facilitate particular types of learning processes (e.g., case-based learning, theme-based learning); knowledge transfer methods for the delivery of training (e.g., tutoring and mentoring); and, knowledge engineering methods covering a range of activities including authoring structuring, administration and learning management. CORONET-Train has been successfully used and evaluated within two companies (see [3] for further details). The main lessons that were drawn from this evaluation were:

- Users needed some time to handle the new learning environment before they could effectively focus on a specific learning activity. For example, shifting from the conventional presence learning mode to using the Internet is not obvious for learners who have not yet had experience with or have not been sufficiently prepared to using the new learning and knowledge transfer processes offered by the learning methodology. As a result preparation in the form of initial training followed by a pilot period for exploratory usage may have been advisable.
- Similarly, adjusting to the types of interpersonal relations typically occurring in conventional classroom settings by interactions between the learner and the web-based learning environment requires new competence on the part of trainers, tutors, and authors of learning materials.

- Finally, using a large-scale collaborative learning platform may require changes to the organization and specification of roles and responsibilities within the organization to be carried out in parallel. The use of CORONET-Train in the evaluation settings made it clear that the platform has to be capable of being tailored to complement norms and practices (e.g., who is responsible for training? who carries it out? etc) within the organization. Likewise, use of the platform and its methodology may also mean that these roles and responsibilities have themselves to be re-thought and in some case redesigned.

4 Conclusions: Putting it all together – what have we learnt?

Looking back on our experiences gained through working in the area of professional SE Education and Training, there are a number of wider lessons learnt that have implications for SE Education and Training. In the following sections we list the following points.

4.1 Adapt methods and media used for delivering training to the needs of SE professionals

One general experience with delivering training to SE professionals is that no single method, be it technology-based or otherwise, guarantees successful learning outcome. Rather the method of delivery depends critically on the characteristics of the educational situation and should take into account the more general components of learning described earlier in section 2 and figure 1. Hence, a multidimensional approach integrating multiple methods, disciplines and media is in our view the "silver bullet".

In recent years, for example, we have been involved with projects that have assumed that building a database or a community-of-practice for software engineers will automatically empower individuals to learn SE methods and techniques. Much of the initial discussions regarding the set up of the community have focused on, at this point new, technical questions (e.g., design of the platform, what functionality it should have, SCORM etc) and only later considered questions such as the type of content that should be delivered and how this should be integrated into the learner's work tasks. In some cases it may have been more appropriate to consider other options earlier and consider their relative didactical advantages and disadvantages. For example, communities of practice may be more suitable for interpersonal exchange on general SE topics (e.g., Software Design) and experiences, whereas a self paced or blended learning approach may be more appropriate for communicating clearly defined techniques and methods (e.g, UML). Likewise, individuals working in small companies may benefit from workplace-integrated approaches like APO-IT as compared to blended learning where support for coaching or training courses may not be available due to the costs involved. In other words, it is especially important to identify the specific goals and the target group for learning as early as possible and to adapt the method and media used to deliver learning.

4.2 Facilitate professional SE education in day to day work

A theme that runs through many of the case studies we have described is that when implemented in the right way integrating learning with work tasks can reap many benefits for the learner. These benefits include increased self-confidence, higher levels of motivation and feelings of independence, which come about through the individual taking responsibility for the content and timing of their learning. For example, in the APO-IT project we found that when provided with some initial guidance regarding how to structure and organize their learning activities, learners begin to move away from a reliance on these materials and take responsibility for their own learning needs. This learning is very much in the style of the "learning-to-learn" type described in section 2.

The preconditions that facilitate this kind of learning or attitude change involves, as with both the APO-IT and Blended learning examples, providing some sort of "support mechanism", which complements activities carried out alone (as in APO-IT) or with technological support (Blended learning). In both cases, we would conjecture that coaching and consultancy provided a form of "cognitive scaffolding" of the type described by Bruner ([4]), which supports the learner both during the learning process and in the application of the new knowledge. In the same manner, this support mechanism may help learners to reflect upon the materials and to question their own practice. Taken a step further, learning thus becomes more than just the acquisition of technical skills and moves towards the learner reframing what they have learnt and how they have learnt, in order to create new knowledge (i.e., learning as an active and constructive process). In many respects, it was the lack of such a support mechanism that caused problems for users of the CORONET-Train system and made it difficult for them to adjust to technology-support collaborative learning.

A second theme running through the examples is importance of motivation and identifying ways in which this can be encouraged and sustained over time. In the case of the Blended learning example this may take the form of providing some form of reward for completing a course (e.g., a certificate). In others, increasing motivation may be much more subtle and harder to facilitate. With communities-of-practice for example, initial motivation to take part in the community may simply be a matter of providing as much relevant content at the outset. Later on, encouraging users of the community to actively share information and discuss SE issues may be much harder. Deciding at the outset what the function, scope and overall goal of the community is critical.

4.3 Wider considerations and the big picture

A final issue that is worth mentioning related to wider organizational and social changes that may be brought about, or may be influenced, by the various mechanisms which exist for delivering and facilitating professional SE. In the CORONET-Train example we saw that introducing the system brought about changes to organizational structures in the form of training and educational roles and responsibilities. Part of the problem with introducing the system was that existing roles within the companies in which CORONET-Train was evaluated, did not match the assumptions implicitly designed into the system. In some cases, as with the Blended Learning example, it may be enough to rely upon management support and the role of a "champion" to make sure that roles are adjusted to complement the new technology being introduced. However, this is unlikely to be successful and more often than not leads to the rejection of the system by its end users.

In an ideal situation, development of any form of technology-based learning whether it be a collaborative-learning platform, a web-based training, or other supportive actions, needs to be carried out in parallel with inclusion of the end users and a consideration of how it will impact upon the complex social structures and wider systems existing within the organization in which it is being introduced. At the very minimum technology-based learning systems need to be flexible enough to cope with such demands and thereby complement, and not conflict with, other process improvement or competency development activities taking place within the wider software organization.

References

[1] Abran, A., Moore, J.W. (eds.), Guide to the SWEBOK 2004 <http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf> (22.10.2005)

[2] Anderson, J. R., Rules of the Mind. Hillsdale, NJ: Lawrence Erlbaum, 1993.

[3] Ankasaputra, N., Pfahl, D., Ras, E., Trapp, S. The Collaborative Learning Methodology CORONET-Train: Implementation and Guidance. In Proceedings of Learning Software Organisations (LSO 2002), 13-24

[4] Bruner, J., Acts of Meaning. Harvard: Harvard University Press, 1990.

[5] Bunse, Chr., Grützner, I. Ochs, M., Peper, Chr., Steinbach-Nordmann, S., Applying a Blended Learning Strategy for Software Engineering Education. CSEE&T, pp. 95-102, 18th Conference on Software Engineering Education & Training (CSEET'05), 2005

[6] Collins, A., Brown, J. S., and Newman, S. E.. Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), Knowing, learning, and instruction: Essays in honor of Robert Glaser (pp. 453-494). Hillsdale, NJ: Lawrence Erlbaum Associates, 1989

[7] Duggins, S., Bernal Thomas, B., An Historical Investigation of Graduate Software Engineering Curriculum. CSEE&T 2002 pp. 78

[8] Faure, E., Herrera, F., Kaddoura, A. R., Petrovsky, A., Rahnema, M., and Ward, F,. Learning to be: The world of education today and tomorrow. Paris, France: UNESCO, 1972.

[9] Furth, H. G., Piaget for teachers. Englewood Cliffs, NJ: Prentice-Hall, Inc, 1970.

[10] Grützner, I., Hebestreit, C., Ochs, M., Vollmers, C., Waterson, P.E., Wie wird E-Learning genutzt? - eine empirische Untersuchung. DeLFI, 2003, pp. 382-391

[11] Hilburn, T.B., SE Education: A Modest Proposal. IEEE Software 14(6), pp. 44-48, 1997

[12] Hofmann, B. and Wulf, V., Building Communities among Software Engineers: The ViSEK Approach to Intra- and Inter-Organizational Learning. LSO 2002: 25-33 <http://cse.unl.edu/LSO2002/Hoffmann-Wulf.pdf>, 2003

[13] IEEE, ACM, Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. <http://sites.computer.org/ccse/> (22.10.2005)

[14] Jensen, C., Scacchi, W. Collaboration, Leadership, Control, and Conflict Negotiation in the NetBeans.org Community. Proc. 4th. Workshop on Open Source Software Engineering, Edinburgh, Scotland, May 2004

[15] Raelin, J.A., Work-based Learning: The New Frontier of Management Learning. New Jersey: Prentice-Hall, 2000.

[16] Shaw, M., Software engineering education: a roadmap. Proceedings of the Conference on The Future of Software Engineering, p.371-380, June 04-11, 2000

[17] Waterson, P.E., Avram, G., Kerr, M.P. and Punter, T., On-Line Communities: What motivates people to take part? In D. Remenyi (Ed.), Proceedings of the European Conference on Knowledge Management (ECKM 04), CD-ROM (ISBN- 0-9547096-5-9), London: Academic Conferences, 2004.

Document Information

Title:	Eight Years of Delivering Professional Education and Training for Software Engineering at Fraunhofer IESE: An Experience Report
Date:	October 2005
Report:	IESE-092.05/E
Status:	Final
Distribution:	Public