FOCUS: An Adaptation of a SWEBOK-based Curriculum for Industry Requirements

S G Ganesh*, Girish Suryanarayana*, Arbind Kumar Gupta†, Raghu Nambiar†
* *Siemens Corporate Research & Technologies, India*
† *Siemens Corporate Development Center, India - Technology Evaluation & Competences*
Siemens Information Systems Limited, No. 84, Keonics Electronics City, Hosur Road, Bangalore-560100, India.
Email: {ganesh.samarthyam,girish.suryanarayana,arbind.gupta,raghu.nambiar}@siemens.com

***Abstract*—Siemens Corporate Development Center India (CT DC IN) develops software applications for the industry, energy, health-care, and infrastructure & cities sectors of Siemens. These applications are typically critical in nature and require software practitioners who have considerable competency in the area of software engineering. To enhance the competency of engineers, CT DC IN has introduced an internal curriculum titled "FOundation CUrriculum for Software engineers" (FOCUS) which is an adapted version of IEEE's SWEBOK curriculum. The FOCUS program has been used to train more than 500 engineers in the last three years. In this experience report, we describe the motivation for FOCUS, how it was structured to address the specific needs of CT DC IN, and how the FOCUS program was rolled out within the organization. We also provide results obtained from a survey of the FOCUS participants, their managers, and FOCUS trainers that was conducted to throw light on the effectiveness of the program. We believe the insights from the survey results provide useful pointers to other organizations and academic institutions that are planning to adopt a SWEBOK-based curriculum.**

***Keywords*-SWEBOK; ISO TR 19759; Software engineering education; Industry experience**

I. Introduction

Siemens Corporate Development Center India (henceforth referred to as CT DC IN) provides software solutions for four Siemens' sectors - Industry, Energy, Healthcare, and Infrastructure & Cities. Most applications in these domains are of a critical nature. For example, a power system management application deployed in a control center must be able to receive rapidly arriving data from field devices, process it, and make informed decisions in real time. Critical applications that perform poorly or do not work as specified can not only lead to loss in user confidence and credibility but can also lead (in severe cases) to serious injuries and loss of human life and property. Additionally, these applications are highly complex and sometimes span more than a million lines of code. Managing the complexity of developing and maintaining such kind of software is indeed a difficult task and requires practitioners with sound software engineering knowledge operating at a high skill level [4].

Some critical projects in CT DC IN are in maintenance phase. Project engineers are expected to perform maintenance tasks within a short turnaround time. Lehman observes that the quality of a system will appear to be declining unless it is rigorously maintained and adapted to operational environment changes [12]. In this context, containing "code decay" [6] or "design erosion" or "architectural erosion" [20] is a challenging task. Software engineers are trained in language features, but do not know how to effectively and safely use them in practice; inappropriate modification of code can lead to "code decay". Inexperienced engineers do not have good understanding of refactoring; when they make modifications that negatively affect the existing design of the software, it results in "design decay". Hence, in addition to software engineering (henceforth SE) knowledge, software developers need to be aware of topics such as refactoring and best practices in programming.

An informal survey of project managers of CT DC IN in 2008 revealed that according to the managers, there was a gap between the expected and actual SE knowledge and programming skills for many practitioners. However, the urgency to bridge this gap was most pronounced in the case of engineers with 2-4 years of experience. This is because this group of engineers was expected to shoulder more responsibilities and play a significant role in their projects in the future. The managers felt a distinct need for a suitable training for this group of engineers. This thought is also echoed in the "Mythical Man Month" where Fred Brooks [5] emphasizes the role of great practitioners in the success of software projects and suggests the use of short courses to help develop potential practitioners for future assignments.

While Siemens globally has an internal certification program for software architects [16], there was no training specifically aligned towards SE. CT DC IN, therefore, introduced the FOCUS (FOundation CUrriculum for Software engineers) program in early 2009 for training engineers with 2-4 years of experience. FOCUS is an adaptation of a SWEBOK-based curriculum [3]). FOCUS also includes topics such as refactoring and best practices in programming to enable development and maintenance of quality software. CT DC IN has conducted FOCUS trainings for more than 500 engineers in the last three years. In order to assess the effectiveness of the current FOCUS curriculum and

978-1-4673-1067-3/11/$26.00 © 2011 IEEE

trainings towards directing future improvements, CT DC IN has conducted a survey to seek feedback from the participants, their managers, and the trainers. The overall feedback indicates that the FOCUS curriculum and trainings have indeed met the basic organizational goals of increasing the SE knowledge and programming skills of software engineers with 2-4 years of experience. Additionally, a few clear directions for the future have emerged. We expect these will guide the further tuning of FOCUS program to make it more effective in the context of CT DC IN.

There are many accounts [10], mainly in academic contexts, of how a SWEBOK based curriculum has been adopted, for example, [19]. There are also some reports on experiences with using SWEBOK for training industry professionals [7], [15]. However, our study of the publicly available data and publications has revealed that there are very few software organizations who have adopted a SWEBOK based curriculum on such a large scale as CT DC IN. Our experience with adopting a SWEBOK-based curriculum to train a large number of CT DC IN engineers with 2-4 years of experience has clearly highlighted its usefulness. Our objective, therefore, behind reporting our experience with a SWEBOK-based curriculum is to attest to its potential to be effectively adopted on a wider scale in the industry[1].

The rest of the paper is organized as follows. Sec. II briefly covers the organizational context, and provides an overview of IEEE certifications and SWEBOK; Sec. III describes how the FOCUS curriculum was conceived, defined, and rolled out within CT DC IN. Sec. IV discusses the survey results and Sec. V reflects upon the lessons learned from the FOCUS training. Finally, Sec. VII outlines the limitations of the paper.

II. Background and related work

A. SWEBOK

IEEE's SWEBOK (SoftWare Engineering Body Of Knowledge) is a guide to the body of knowledge on SE that exists in the form of literature. The purpose of SWEBOK is to "provide a consensually validated characterization of the bounds of the SE discipline and to provide a topical access to the Body of Knowledge supporting that discipline" [3]. SWEBOK is also an ISO technical report (ISO/IEC TR 19759:2005) [1]. ISO/IEC 24773:2008 standard [2], which is a framework for comparison of schemes for certifying software development professionals, refers to a Body of Knowledge, for which SWEBOK is a reference. IEEE's CSDA and CSDP certifications conform to this standard [9].

SE 2004 [13] and GSwE2009 [18] respectively provide curriculum guidelines for undergraduate and graduate degree programs in SE; both curricula relate to Knowledge Areas (KAs) in SWEBOK. SWEBOK is an important step "to make software engineering respectable" [17] since it provides a guide to the body of knowledge in SE. Further, to quote from SWEBOK [3]:

> "In engineering, the accreditation of university curricula and the licensing and certification of practicing professionals are taken very seriously. These activities are seen as critical to the constant upgrading of professionals and, hence, the improvement of the level of professional practice. Recognizing a core body of knowledge is pivotal to the development and accreditation of university curricula and the licensing and certification of professionals."

B. IEEE certifications

IEEE has three certifications targeting software developers at different levels of expertise [8]:

- **CSDA:** The Certified Software Development Associate certification is meant for entry-level developers and was launched in 2008.
- **CSDP:** The Certified Software Development Professional certification is meant for experienced developers and was launched in 2002.
- **CSDM:** The Certified Software Development Master certification is meant for specialists and is yet to be launched.

In addition to these assessment-based certifications, IEEE has a training-based certification program for the SWEBOK called "SWEBOK Certificate Program" (SCP). The current CSDA and CSDP curricula cover five more KAs (totaling 15 KAs, listed in table II) than the ones covered in SWEBOK: SE Professional Practice, SE Economics, Computing Foundations, Mathematical Foundations, and Engineering Foundations.

C. Organizational background

CT DC IN[2] develops a wide range of software solutions for the four sectors of Industry, Energy, Healthcare, and Infrastructure & Cities. CT DC IN is spread across five locations in India and employs over 2,600 software professionals. CT DC IN is a part of Siemens Information Systems Ltd (SISL). All software engineers employed in CT DC IN have either a bachelor's or a master's degree and are mostly involved in developing, testing or maintaining the software. A considerable percentage of CT DC IN engineers come from disciplines other than computer science (such as mechanical or electrical and electronics engineering).

Software plays a key role in most of the Siemens products and hence developing high quality software is very important to the organization. An informal survey of CT DC IN project managers in 2008 reported that project engineers (with 2-4

[1] An article in the 2005 issue of the Communications of the ACM [14] made a strong statement that, "failure to endorse SWEBOK and the dearth of published uses are a shame to the industry."

[2] http://www.siemens.co.in/en/about_us/index/innovations/sisl.htm

years of industrial experience and henceforth referred to as "target group") were expected to increasingly shoulder more responsibilities in their projects in the near future; however, their existing SE knowledge and development skills were considered inadequate for this purpose. Hence, there was a compelling need to train these engineers on SE topics. It should be pointed out that every CT DC IN employee does undergo mandatory training on topics such as configuration management and process & quality. While these trainings provide necessary background on a few important SE topics, they cannot be considered sufficient to cover a wide range of important topics in SE.

III. FOCUS CURRICULUM AND TRAINING

A. Scope, objectives, and course contents

The main requirement that emerged from the informal survey of CT DC IN project managers was the need for a training that would help develop a high level of competency in SE in the target group of engineers. Our approach to meet this requirement consisted of the following steps.

First, we determined the general background and experience of the target group of engineers so that a relevant curriculum could be designed. The main responsibilities of the target group of engineers in CT DC IN relate to detailed design, software construction or maintenance, documentation, unit testing, and adhering to quality processes. Since they have some work experience, these engineers are already familiar with the software development environment. They also have some ideas about what makes SE difficult, especially in the context of complex projects that are also distributed geographically. They have also undergone various trainings (on both hard and soft skills), including some of the topics in SE.

Next, using the engineers' background and the project managers' requirements as the basis, we decided that the training program should have the following objectives:

- It should help them have a good grasp of various topics in SE with main focus on four topics: foundational concepts of SE, software design, software construction, and software testing.
- It should help them perform better in their regular tasks such as design, construction, unit testing, refactoring (which is important in the context of agile development as well as for effective maintenance), and writing technical documentation.
- It should introduce them to language-independent best practices in programming. For example, since CT DC IN mostly deals with object-oriented software, these best practices should include the application of proper object-oriented principles such as abstraction and encapsulation. They should also be able to apply these concepts in developing or maintaining software in a programming language relevant for their project. Currently, C++, Java and C# are the most widely used languages, in addition to other languages, within our organization.
- It should also introduce them to the concepts behind software development tools such as static and dynamic code analyzers, SCM tools, and testing tools.

The final step in our approach was to decide upon the curriculum which would meet the above objectives. Towards this end, we surveyed the curriculum adopted and recommended by various organizations and universities, and finally arrived at the following alternatives:

- Following a SE curriculum prescribed at a reputed university
- Following a SE curriculum presented in a well-known SE book
- Asking in-house experts within the organization to design an organization-specific curriculum from scratch
- Following the SWEBOK-based CSDA curriculum

There were several discussions on which of the above was the best option for CT DC IN. Since the SWEBOK provides a complete guide to the various topics in the area of SE and has been created and validated by a group of SE experts recognized by the community, CT DC IN decided to use the CSDA curriculum (which is based on the SWEBOK) as the basis for the FOCUS program. It should be noted that, as per IEEE, the CSDA curriculum is meant for fresh engineers as well as for engineers with some work experience. Our initial experience with CSDA curriculum showed that understanding and following the concepts in the curriculum is better facilitated if the participants have exposure to real-world projects. For example, engineers can recognize the importance and understand the concepts in Software Maintenance KA when they have work experience relating to maintenance of large and complex real-world software. So, we felt that while the CSDA curriculum may seem a bit difficult for fresh engineers to relate to, it would actually be a good fit for our target group of engineers.

However, in order to meet the objectives of CT DC IN (covered earlier in this section), the depth and duration of topics recommended by CSDA needed to be altered and several new topics such as technical writing and best practices in programming needed to be included to meet the CT DC IN objectives. This resulted in a new curriculum which we call FOCUS. Table I shows the topics covered in FOCUS, and how a FOCUS training session is organized.

B. Evaluation method

Once the FOCUS curriculum was finalized, we needed to address how the learning of the participants could be evaluated post-training. We considered several alternatives including an objective type test which is followed by IEEE for CSDA and CSDP certifications, evaluating the participants based on project-specific case studies, assignments, presentations etc. Most of these assessment alternatives, except the

Table I
FOCUS TRAINING: TOPICS COVERED AND DURATION

Day	Topics covered (duration in hours)
Day 1	Software requirements (4 hrs), configuration management (3 hrs)
Day 2	Computing, mathematical and engineering foundations (7 hrs)
Day 3	Software design (7 hrs)
Day 4	Software construction and refactoring (7 hrs)
Day 5	Software testing (7 hrs)
Day 6	Language independent programming best practices, concepts of software development tools (7 hrs)
Day 7, 8	Language specific programming best practices in C++/C#/Java (14 hrs)
Day 9	SE process (2 hrs), software quality (2 hrs), tools and methods (3 hrs)
Day 10	Software maintenance (2 hrs), SE management (2 hrs), technical writing (1 hr), software project dynamics (1 hr), SE professional practices (1 hr)

Table II
WEIGHT OF TOPICS (BASED ON PERCENTAGE OF TEST QUESTIONS): CSDA VS. FOCUS

Knowledge Area (KA)	CSDA % weight	FOCUS % weight
Software Requirements	7%	5%
Software Design	8%	10%
Software Construction	10%	15%
Software Testing	7%	10%
Software Maintenance	7%	7%
Software Configuration Management	3%	3%
Software Engineering Management	3%	3%
Software Engineering Process	4%	3%
Software Engineering Methods	5%	3%
Software Quality	6%	5%
Software Engineering Professional Practice	7%	3%
Software Engineering Economics	3%	3%
Computing Foundations	10%	20%
Mathematical Foundations	10%	5%
Engineering Foundations	10%	5%

objective type test, involve an element of subjectivity from evaluators. Further, the effort required for conducting an objective type test can be reduced significantly if the process of setting the question paper and evaluating answers can be automated. For these reasons, we decided to adopt a CSDA-type assessment test for the FOCUS program. However, since there are many new topics covered in our curriculum (see Table I), and the depth of topics is different from CSDA, we decided on our own internal test for FOCUS. Additionally, the participants have an option to appear for the CSDA exam with the approval of their managers. Table II compares the weight of topics in CSDA as on Dec'08 (when CSDA was introduced) with the weight that we currently follow in the FOCUS curriculum.

C. Training delivery

Our final task was to address the delivery of the FOCUS training. An initial option we explored was web-based content delivery. However, since many of the SE topics require considerable discussion, it was decided to follow a instructor-led classroom training approach for FOCUS. This would also enable participants' questions to be answered immediately in the context of the concepts being discussed. Next, we explored whether we could invite professional trainers from outside the organization to conduct these trainings. The main concern here was whether the topics could be explained using examples of real-world projects from our organization. This was really important since the target group of engineers was being trained to shoulder more responsibilities in the future. Therefore, it was decided that the instructors would be internal to the organization. This would also enable the instructors to guide the FOCUS trainees to correlate SE concepts to the processes and methodologies already being followed within the organization.

The preparation for FOCUS training started six months before the actual roll-out in March 2009. As mentioned above, it was decided that the FOCUS trainers would be internal to the CT DC IN organization. Towards this end, potential trainers were nominated by different business units. These trainers underwent training and received certifications for CSDA and CSDP. They also participated in a "Train The Trainer" (TTT) session conducted by IEEE certified trainers. Following this, in September 2009, four of the trainers were certified by IEEE as "Software Engineering Certified Instructors" (SECI). This is a certification for providing training in CSDA/CSDP. Subsequently, in December 2010, these SECI's have also contributed to the course material for the "SWEBOK Certificate Program" (SCP) [11].

The trainers were allotted topics in the FOCUS curriculum based on their background, experience, and interest. Trainers were responsible for creating the course content for their topics using the CSDA course content as the reference. In spite of being constrained by the limited duration and the extensive course content that needed to be covered for a topic, trainers added examples of real-world CT DC IN projects and discussion-provoking questions wherever applicable to facilitate better understanding of the concepts. The created training material was subjected to a stringent review process by all the trainers to ensure quality. Trainers also helped create a high-quality question bank which was used for the FOCUS objective type test.

D. Roll-out

As of October 2011, 22 FOCUS sessions have been successfully conducted and more than 500 participants have been trained so far. Each FOCUS batch consists of 22-25 participants. After every FOCUS session, feedback on the course content and the trainers is collected from the participants. This anonymous feedback is subsequently aggregated

Table III
FEEDBACK FROM PARTICIPANTS ON FOCUS TRAINING

Q.No	Question (note in parenthesis)	Mean (Overall)	Mean (CS)	Mean (Non-CS)
Q1.	How much did the training help you in your regular work [0 - 10] (0 - no change; 10 - significant change)	5.3	5.2	5.6
Q2.	Did the training bring out a change in approach towards your work [0 - 10] (0 - no change; 10 - significant change)	5.5	5.4	5.7
Q3.	Do you think this training will contribute to your overall development in the long run [0 - 10] (0 - no change; 10 - significant change)	6.4	6.2	7.1
Q4.	Rate your overall SE knowledge before the training [0 - 10] (0 - basic technical knowledge; 10 - knowledge matches the level of "expert" in technical area)	5.1	5.3	4.3
Q5.	Rate your overall SE knowledge after the training [0 - 10] (0 - basic technical knowledge; 10 - knowledge matches the level of "expert" in technical area)	6.8	7.1	6.4
Q6.	Is the duration of FOCUS adequate [0 - 10] (0 - too less; 5 - adequate; 10 too much)	5.4	5.6	5

and shared with the trainers in order to help them improve the lecture material and their delivery for future FOCUS sessions. Once a FOCUS session concludes, the participants of that batch have to undergo the FOCUS test.

IV. SURVEY RESULTS

In an organizational context, it is difficult to conduct studies to quantify the benefits of adopting a new training program. Further, since the FOCUS training mainly consists of SE theory and concepts, the benefits of adopting the program may be visible only in the long run. Hence, we conducted a survey to evaluate the perceived effectiveness and usefulness of the training, and also elicit ways to improve the training.

We sent separate survey questionnaires to FOCUS participants, their managers and FOCUS trainers. While there are a few questions that are common across the three questionnaires, each questionnaire also has some questions targeted towards the specific survey group. For example, of the surveyed groups, only the FOCUS trainers have a clear idea about the CSDA/CSDP curriculum, and the differences between CSDA and the FOCUS training (in terms of weight of the topics, tailoring introduced in FOCUS etc.). So, the trainers' questionnaire had specific questions on these topics (see questions Q7, Q5, and Q6 in Table V). Since responding to this survey was optional and not mandated by the organization, we kept the survey questionnaires short (limiting ourselves to 10 questions) to encourage the survey participants to respond with their feedback.

The survey questionnaires consisted of several rating-based questions (where the participant had to give a rating between 0 and 10 for a question) and a few subjective questions. In the rest of this section, we present the survey results. For each survey group, we first present the results from the rating-based questions, followed by a summary of the feedback from the subjective questions.

A. Survey results - participants' feedback

We sent the survey questionnaire to 50 selected participants. The sampling was distributed uniformly across performance-levels in FOCUS and covered all business units. Out of 50 requests, 32 participants responded. Table III presents the mean values of their responses (since there were 32 responses, the resulting table is quite large, and hence the individual responses are not provided in the table). In this table, the columns "Mean (overall)", "Mean (CS)", "Mean (Non-CS)", refer to the averages from all participants, from participants with computer science (CS) background, and from participants from background other than CS, respectively. (We would have preferred segregating the participants based on having SE as the academic background; however, SE degrees are not common in India compared to conventional CS degrees.) As mentioned earlier, within our organization, a considerable percentage of engineers come from disciplines other than CS. Therefore, we explicitly wanted to track the effectiveness of the FOCUS program for participants from CS as well as non-CS backgrounds. Of the 32 respondents, 9 were from CS background and the remaining 23 were from a non-CS background.

Some noteworthy points based on the table III are:

- In general, participants (both CS and non-CS background) felt that the FOCUS training is useful (based on Q1, Q2, and Q3). The difference in mean values for participants with non-CS and CS backgrounds for these questions is understandable; clearly, a participant from non-CS background would be introduced to many new concepts and would feel a greater impact due to the training as compared to a participant with CS background.
- An interesting observation (from Q3) is that non-CS participants felt that the training will contribute more to their overall development in the long run than the CS participants. One possible reason is that the FOCUS training makes the participants with non-CS background feel technically more equipped to work in a software development environment.
- Understandably, participants from non-CS background thought that their SE knowledge is less compared to participants with CS background (from Q4).

Table IV
FEEDBACK FROM MANAGERS ON FOCUS TRAINING (M STANDS FOR MANAGER IN M1-M9)

Q.No	Question (note in parenthesis)	M1	M2	M3	M4	M5	M6	M7	M8	M9	*Wt. Mean*
Q1.	Did the training bring out an immediate change in approach towards work [0 - 10] (0 - no change; 10 - significant change)	-	6	6	5	4	6	7	5	5	*5.3*
Q2.	Do you think FOCUS will contribute to the overall development of the team member in the long run [0 - 10] (0 - no change; 10 - significant change)	4	7	8	8	7	7	9	8	7	*7.6*
Q3.	Rate the overall SE knowledge (avg) of trainees before the training [0 - 10] (0 - basic technical knowledge; 10 - knowledge matches the level of "Expert" in technical area)	4	5	6	5	3	5	4	6	5	*4.6*
Q4.	Rate the overall SE knowledge (avg) of trainees after the training [0 - 10] (0 - basic technical knowledge; 10 - knowledge matches the level of "Expert" in technical area)	6	6	8	5	5	7	7	8	7	*6.2*
Q5.	Increase in value addition (overall) to the project by the trainee [0 - 10] (0 - no improvement; 10 - has become a key player of the project team as a result of the increased technical skill and contribution)	-	5	7	2	3	5	7	7	6	*4.8*
Q6.	Is the duration of FOCUS adequate [0 - 10] (0 - too less; 5 - adequate; 10 - too much)	5	5	5	-	8	0	8	-	7	*6.7*
Q7.	Do you think conducting a formal test (with pass/fail result) is a good way to judge effectiveness of the training [y/n] (y - yes it is sufficient, n - no it is not sufficient))	Y	Y	Y	Y	Y	Y	Y	-	N	-

- It is interesting to note (based on Q5) that even participants with CS background experienced a 34% increase in their overall SE knowledge. This is a bit surprising since we expected this group of participants to be already familiar with most SE concepts. This clearly reveals that the adapted SWEBOK-based curriculum is relevant and useful even for participants with CS background (and with 2 to 4 years of experience in the software industry).

We list below the subjective questions and a summary of the responses received from participants:

- **Which are the areas/topics where maximum improvement is observed?** The participants mentioned a wide-range of topics the most common being software requirements, design and design patterns, code quality, and testing.
- **Which areas/topics should be considered as core technical topics that will be helpful for your work?** A couple respondents mentioned better language skills and knowledge about SDLC; most others mentioned design and design patterns. Perhaps, it indicates that most software engineers in the target group are interested in honing their design skills.
- **What are your suggestions about areas of improvement in the program?** Most suggestions hinted at increasing the emphasis on the practical application of the covered SE concepts in the form of real-world case-studies, group projects, etc. Some participants also desired longer sessions on requirements and design.

B. Survey results - managers' feedback

We sent the survey questionnaire to all the 18 Line of Business heads (henceforth referred to as managers) in CT DC IN. In response, we received feedback from 9 managers. The detailed results are provided in Table IV. The value in the weighted mean (Wt. Mean) column is computed by weighting the rating given by a Line of Business (henceforth, LoB) head with the number of FOCUS participants belonging to that LoB. This ensures that the opinion of a LoB head whose group had a higher number of FOCUS participants is given more weight than that of a LoB heads with a lesser number of FOCUS participants.

Some noteworthy points based on the table IV are:

- The LoB heads in general agreed that the training has helped bring immediate positive change in the participants' approach towards work (based on Q1). Further, they anticipated a substantial impact in the long run (based on Q2). The latter is indeed significant since it shows that the FOCUS curriculum and training are considered effective by the managers to meet their original requirements.
- All the LoB heads reported a reasonable increase in the overall SE knowledge of the participants (based on Q3 and Q4).
- There is reasonable value addition to the project by the participants: 4.8 in the scale of 0 to 10 (based on Q5).
- There is no clear agreement on the duration of the course though three respondents felt that the duration is adequate (based on Q6).
- Most LoB heads thought that conducting (an objective type) formal test with pass/fail result is a good way to judge effectiveness of the training (based on Q7).

We list below the subjective questions and a summary of the responses received from the LoB heads:

- **Which is the area where maximum improvement**

Table V
FEEDBACK FROM FOCUS TRAINERS (T STANDS FOR TRAINER IN T1-T9)

Q.No	Question (note in parenthesis)	T1	T2	T3	T4	T5	T6	T7	T8	T9	*Mean*
Q1.	No. of years of experience you have in the software industry	20	9	21	18	9	20	4	15	14	*14.4*
Q2.	Rate the trainees' knowledge (avg) on your specific topic before the training [0 - 10] (0 - basic technical knowledge; 10 - knowledge matches the level of "Expert" in technical area)	4.5	4	4	3	4	2	2	3	6	*3.6*
Q3.	Rate the trainees' knowledge (avg) on your specific topic immediately after the training [0 - 10] (0 - basic technical knowledge; 10 - knowledge matches the level of "Expert" in technical area) [give an approximate idea with respect to (2) to what level would your training session have increased the trainees knowledge in your specific topic]	6.5	6	7	-	5	5	4	5	6	*5.5*
Q4.	Do you think FOCUS will contribute to the overall development of the trainee in the long run [0 - 10] (0 - no change; 10 - significant change)	6	7	6	7	6	5	5	5	5	*5.8*
Q5.	Rate the usefulness of the CSDA curriculum if it were to be introduced as it is to the trainees [0 - 10] (0 - not useful; 10 - significantly useful)	5	5	5	5.5	5	5	3	6	7	*5.2*
Q6.	Rate the usefulness of the current adapted CSDA curriculum on the FOCUS participants (keeping the organizational requirements in mind) [0 - 10] (0 - not useful; 10 - significantly useful)	8	7	6	4.5	6	5	7	7	7	*6.4*
Q7.	Based on your CSDP experience, rate the usefulness of introducing CSDP curriculum for software engineers with >5 years experience in the future in the organization [0 - 10] (0 - not useful; 10 - significantly useful)	6	5	4	5.5	8	6	7	7	5	*5.9*
Q8.	Is the duration of FOCUS adequate [0 - 10] (0 - too less; 5 - adequate; 10 - too much)?	4	4	5	6	3	5	4	3	8	*4.7*
Q9.	Do you think conducting a formal test (with pass/fail result) is a good way to judge effectiveness of the training [y/n] (y - yes it is sufficient, n - no it is not sufficient)?	N	Y	N	Y	N	N	N	Y	Y	-

is seen after the FOCUS training? Most managers mentioned that the improvement was observed in the form of better quality of design, code and (unit) tests. Another manager added that FOCUS has helped improve the participants' understanding of SDLC and that specifically participants from non-CS background benefited the most due to FOCUS.

- **What would you suggest to improve the FOCUS training?** There were several different suggestions to improve the training. However, three managers stressed on one common point that the participants should be asked to practically apply the SE concepts being covered to case-studies or use them in hands-on exercises. One manager suggested that all participants who clear FOCUS should be asked to get a CSDA certification; this would help determine the standard and quality of the FOCUS curriculum and training. Finally, one manager seemed to be impressed with the value-add of the FOCUS training and wanted even experienced engineers to undergo that training.

C. Survey results - trainers' feedback

The survey questionnaire was sent to all 9 CSDP certified trainers (of them four were SECI certified trainers) and all of them responded. Table V presents the mean values of their responses.

Some noteworthy points based on the Table V are:

- All the trainers felt that the FOCUS training will benefit the participants in the long run (based on Q4).
- In general, the trainers felt that the the current adapted CSDA curriculum is more useful to the participants than if the CSDA curriculum were to be introduced as is (based on Q5 and Q6).
- Most trainers felt that introducing CSDP curriculum for software engineers with >5 years experience in the future in the organization would be beneficial (based on Q7).
- Most trainers felt that the duration of the FOCUS training can be increased (based on Q8).
- An interesting finding from the trainers is that many of them do not consider that conducting a formal test (with pass/fail result) is a good way to judge the effectiveness of the FOCUS training (based on Q9).

We list below the subjective questions and a summary of the responses received from the FOCUS trainers:

- **What do you think can be done to improve the current FOCUS curriculum?** Most trainers were concerned about how the participants could apply the theoretical SE concepts they learned during the training into practice. Specifically, three trainers suggested evaluation based on the participants' presentations on how the SE concepts they have learned have been applied or relate to their projects. Some trainers also suggested the use of more case studies for better comprehension of concepts.
- **What topics in the current FOCUS curriculum do**

you think should have more focus? Most trainers mentioned that more focus should be given to design, construction, refactoring, and testing topics. Further, two trainers mentioned that computing foundations should receive more attention to help participants who have a non-CS background.

D. Survey results - summary

In this section, we summarize the feedback received from the three survey groups:

- All three survey groups agree that the training helps bring out both an immediate change towards work as well as increased benefits in the long run. It is interesting to note that of the three survey groups, it is mainly the managers who believe that the FOCUS training will bring more benefit in the long run (Q2 in IV).
- While managers consider it acceptable to have an objective type test, many trainers consider this to be insufficient to judge the effectiveness of the training. Since the trainers are familiar with the technical topics as well as have experience with objective type tests (since they have CSDA and CSDP certifications), their inputs on the value of the exam is important.
- On improving the FOCUS training, all three survey groups clearly agree that more stress should be on applying the SE concepts in practice using case-studies, hands-on-projects, presentations etc.
- Design, design patterns, refactoring, testing, and computing foundations topics require more focus.

V. Discussion

Based on our experience, and from the feedback we have received (Sec. IV), we provide a short summary of our experience, lessons learned, and the future directions.

A. Effectiveness of FOCUS

The FOCUS training program was designed with the organizational requirements in mind, as outlined in Sec. III-A. It is clear from the survey feedback (Sec. IV) that FOCUS has reasonably met the organizational requirements. Further, CT DC IN has been receiving increasingly positive feedback from its customers since the last two years. While the CT DC IN management attributes some of this improvement to the FOCUS trainings, the exact contribution of FOCUS trainings towards this improvement cannot be quantified.

Based on the positive feedback received for the training, we have recently introduced a training program for fresh engineers (0-2 years of experience) also. As mentioned earlier in Sec. III, we found CSDA curriculum to be mainly suitable for engineers with 2-4 years of experience: fresh engineers lack real-world project experience, they would find it difficult to make best use of the CSDA curriculum. For this reason, we have introduced a shortened version of the FOCUS training program for freshers, which is named as "FOCUS-Starter".

Table VI
FOCUS-Starter training: topics covered and duration

Topics covered	Duration
Software requirements, software design, software construction, software testing	2 days
Software configuration management, software maintenance, software quality, tools and methods	1 day
Programming best practices, concepts of software development tools, introduction to development technologies	3 days

B. Training for freshers (FOCUS-Starter)

The FOCUS-Starter training program has been designed with the following considerations:

- The training program should provide an overview of important and relevant SE topics, neither covering too many topics, nor covering any specific topic in depth.
- The training should introduce participants to language-independent best practices in programming(such as concepts of defensive programming, ensuring exception safety, and following coding standards) early in their career, so as to enable them to develop quality software.
- The training should also introduce concepts behind software development tools such as static/dynamic analyzers, SCM tools, and testing tools.
- The training should also provide an overview of development technologies such as .NET, COM, and XML.

The topics and duration for this instructor-led training program is given in table VI. First three days are dedicated to SE topics in which we cover 7 KAs (of total 15 KAs in CSDA curriculum) that are important and relevant for freshers. In the following three days, we cover programming best practices, development tools and technologies. The training is followed by an assessment test, with the same level of difficulty as CSDA and FOCUS (but with lesser number of questions). So far, four FOCUS-Starter training sessions have been conducted and around 100 fresh engineers have been trained under this program.

C. Future directions for FOCUS

Based on the survey feedback, we have identified a set of changes we believe will help tune the FOCUS curriculum and trainings to make them more effective to the CT DC IN context. Some key enhancements are listed below:

- Reorganization of the FOCUS training into two parts: one on topics that are common for engineers in all roles and the other on topics that are more relevant to developers.
- Further increase in the weight of Computing Foundations, Design, Construction and Testing topics. This includes more discussion on data structures and algorithms, refactoring, design patterns, and unit testing.

- Addition of more case studies and real-world examples to the lecture material.
- Using case-study based evaluation in addition to objective type test.

VI. Lessons learned

The lessons learnt from our FOCUS experience can be broadly categorized into those that apply to the curriculum and those that apply to the training. In this section, we list some suggestions related to the curriculum and trainings that we believe can serve as useful pointers to organizations that are planning to adopt a SWEBOK-based curriculum.

Our experience with FOCUS shows that there are significant benefits in adopting a SWEBOK-based curriculum. However, the following points should be considered before adopting it in an industrial context:

- The CSDA curriculum, if adopted as it is, may bring more benefits in the long run but its immediate impact may be missing. An organization that wants to see immediate impact may want to consider adding topics to the curriculum that can help generate an immediate impact. For example, in the case of FOCUS, topics related to best practices in programming resulted in immediately perceivable benefits (i.e. better quality design and code).
- The CSDA curriculum covers a wide-range of topics in SE, and covers topics such as SE management and SE economics. However, depending upon the context, there may be a more pressing need to train them in important topics such as refactoring and unit testing that get only a passing mention in the CSDA curriculum. Further, relevant topics such as agile development and design patterns are not given adequate importance. While it would be useful if SWEBOK, as well as the curricula based on SWEBOK, address these shortcomings in future, an organization planning to immediately use a SWEBOK-based curriculum can consider adding these topics to the curriculum.
- The CSDA curriculum assigns a certain weight to topics. Depending upon the need, an organization may need to tailor this (perhaps more than once) to fit their purpose. For example, based on the organizational requirements and the background of the target group of engineers, we initially adopted a certain weight of topics for FOCUS as shown in Table II. Feedback from the FOCUS survey, however, shows that this tailoring may need to be fine-tuned in the future.
- The CSDA curriculum covers a lot of useful SE concepts. Results from our FOCUS survey show that further guidance may be required for applying these concepts into practice. An organization may, therefore, need to enhance the curriculum with relevant pointers to guide the practical application of these concepts.

The following points should be considered while rolling out trainings for SWEBOK-based curricula:

- In order to facilitate a better understanding of SE concepts during a training based on the CSDA curriculum in an industrial context, an organization may need to supplement the curriculum with case studies and real-world examples. For instance, though we have added several case studies and real-world stories in the FOCUS lecture material, FOCUS survey results clearly reveal that more such examples are needed.
- In our experience, for a SE training to be effective, an organization must ensure that all the trainers have a common and consistent understanding of the entire curriculum. In the case of FOCUS, this was easily enabled by having CSDP-certified trainers.
- An organization must consider several factors such as the duration of the training, the number of participants in each training, and the evaluation process that can influence the effectiveness of the training program. For instance, feedback from FOCUS trainers indicates that other forms of evaluation are needed in addition to the originally-chosen objective type test.

VII. Limitations

The paper has some limitations. First, it is quite difficult to evaluate the effectiveness and usefulness of a training program such as FOCUS, particularly in an organizational context such as ours. We have tried to address this limitation through our survey of the FOCUS participants, their managers, and FOCUS trainers which has given helpful feedback on the *perceived* usefulness of the training program.

Second, in order to draw reasonable comparisons between a SWEBOK-based curriculum and other curricula (such as from a well-known SE book or from the curriculum followed at a reputed university), an extensive set-up would be required. For instance, it would be ideal for comparative reasons to have the same set of trainers train the same type of participants in multiple curricula. However, in an organizational context, this is extremely difficult to achieve due to economic considerations. The same constraints hold true for doing a comparison between the CSDA curriculum and the FOCUS curriculum (which is an adaptation of CSDA). We have tried to address this limitation to a certain extent by explicitly seeking the opinion of the CSDP-certified trainers on the CSDA versus FOCUS curricula in the survey.

Third, though more than 500 participants across 18 LoBs have undergone FOCUS training, in this paper, we have presented feedback from only 32 participants and from only 9 LoB heads. While this data is certainly limited, partly due to organizational policies which prescribe such surveys to be voluntary in nature, we believe that this initial survey has given us a reasonable glimpse into the effectiveness of FOCUS. Planning is currently underway to introduce an online mechanism to regularly collect feedback from all

three survey groups. Suitable incentives are also planned to encourage participation in the surveys.

VIII. Conclusion

This paper describes our experience with FOCUS (FOundation CUrriculum for Software engineers) which is an adaption of a SWEBOK-based curriculum. FOCUS has been used for the last 3 years in CT DC IN to train more than 500 engineers with 2-4 years of work experience. To evaluate the effectiveness of FOCUS, we have used a survey to gather feedback from FOCUS participants, their managers, and the trainers. The feedback clearly indicates that FOCUS has largely met the organizational requirements. The survey results also point towards certain areas of improvement which are being used to tune the FOCUS program.

A study of existing literature shows that while SWEBOK-based curricula has been successfully adopted in many academic institutions, their adoption has been limited in the industrial context. Based on our successful experience with training a large number of engineers using a SWEBOK-based curriculum, we feel that there is potential for a much wider adoption of such a curriculum in the industry. We hope that our experience and the insights we have gained will serve to provide useful inputs and directions to others who are planning to introduce SWEBOK based training programs in their organizations.

Acknowledgment

We would like to thank all the survey respondents for providing their feedback. We would also like to thank Ravinder Singh for her help in conducting the survey.

References

[1] ISO/IEC TR 19759:2005, Software Engineering - Guide to the Software Engineering Body of Knowledge. *International Organization for Standardization*, Geneva, 2005.

[2] ISO/IEC 24773:2008 Software Engineering – Certification of Software Engineering Professionals – Comparison Framework. *International Organization for Standardization*, Geneva, 2008.

[3] A. Abran, J. Moore, P. Bourque, R. Dupuis, and L. Tripp. Guide to the Software Engineering Body of Knowledge. *IEEE Computer Society*, 2004.

[4] M. Ardis, P. Bourque, T. Hilburn, K. Lasfer, S. Lucero, J. McDonald, A. Pyster, and M. Shaw. Advancing Software Engineering Professional Education. *IEEE Software*, 28:58–63, July 2011.

[5] F. P. Brooks, Jr. *The Mythical Man-Month (Anniversary ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.

[6] S. G. Eick, T. L. Graves, A. F. Karr, J. Marron, and A. Mockus. Does Code Decay? Assessing the Evidence from Change Management Data. *IEEE Transactions on Software Engineering, Vol. 27, Issue 1, pp. 1-12*, 2001.

[7] D. J. Frailey and J. Mason. Using SWEBOK for Education Programs in Industry and Academia. In *Proceedings of the 15th Conference on Software Engineering Education and Training*, Washington, DC, USA, 2002.

[8] IEEE Computer Society. Certification and Training for Software Professionals. *http://www.computer.org/portal/web/certification/ (last accessed 05-Mar-2012)*.

[9] IEEE Computer Society. CS Software Certifications Meet New ISO Standard. *http://www.computer.org/portal/web/pressroom/2008/isoiec (last accessed 05-Mar-2012)*.

[10] IEEE Computer Society. Publications Discussing the SWEBOK Guide - Refereed Conference Papers. *http://www.computer.org/portal/web/swebok/confpapers (last accessed 05-Mar-2012)*.

[11] IEEE Computer Society. SWEBOK Certificate Program. *http://www.computer.org/portal/web/certification/scp (last accessed 05-Mar-2012)*.

[12] M. Lehman and J. Ramil. Rules and Tools for Software Evolution Planning and Management. *Annals of Software Engineering, Special Issue on Software Management, Vol. 11, Issue 1, pp. 15-44*, 2001.

[13] T. C. Lethbridge, R. J. LeBlanc Jr., A. E. K. Sobel, T. B. Hilburn, and J. L. Daz-Herrera. SE2004: Recommendations for Undergraduate Software Engineering Curricula. *IEEE Software*, 23:19–25, 2006.

[14] J. N. Morgan. Why the software industry needs a good ghostbuster. *Communications of the ACM*, 48:129–133, August 2005.

[15] R. Narayanan and S. Neethi. Building Software Engineering Professionals: TCS Experience. In *14th Conference on Software Engineering Education and Training*, pages 162–171, 2001.

[16] F. Paulisch and P. Zimmerer. A Role-based Qualification and Certification Program for Software Architects: An Experience Report from Siemens. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2*, ICSE '10, pages 21–27, New York, NY, USA, 2010.

[17] G. Pour, M. L. Griss, and M. Lutz. The Push to Make Software Engineering Respectable. *IEEE Computer*, 33:35–43, May 2000.

[18] A. Pyster. Graduate Software Engineering 2009 (GSwE2009): Curriculum Guidelines for Graduate Degree Programs in Software Engineering, Integrated Software & Systems Engineering Curriculum Project. *Stevens Institute*, September 30, 2009.

[19] J. B. Thompson and C. J. Hardy. Use and Evaluation of SWEBOK by Postgraduate Students. In *Proceedings of the 15th Conference on Software Engineering Education and Training*, Washington, DC, USA, 2002.

[20] J. van Gurp and J. Bosch. Design Erosion: Problems and Causes. *Journal of Systems and Software*, 61:105–119, March 2002.