Towards Contextualised Software Engineering Education: An African Perspective

Jens Fendler
Polytechnic of Namibia
Department of Software Engineering
jfendler@polytechnic.edu.na

Heike Winschiers-Theophilus
Polytechnic of Namibia
School of Information Technology
heikew@polytechnic.edu.na

ABSTRACT

The discipline of Software Engineering is continuously adapting to new challenges while gaining more and more insights. The age of globalisation has brought about a new movement of internationalisation and localisation. While practitioners fully embrace the efforts, educators only marginally consider the implications for the teaching and learning of Software Engineering. While the relevance of the software deployment context has been widely recognised, the intrinsic values of the development context are less evident. Besides western cultural indicators being omnipresent in software applications, they are deeply rooted in Software Engineering concepts and methods. Standards and models have been established in the absence of possible deviations from other – e.g. African – contexts. Educators and authors of common and internationally used textbooks present Software Engineering concepts and methods as universally valid. Thus software engineering graduates all over the world continue to be ill-equipped for specific software development contexts.

Moreover the necessity to localise Software Engineering education is illustrated by our vast amount of challenges, experiences and best-practices of teaching Software Engineering in a Sub-Saharan country. In this paper, we introduce a generic framework leading towards a Contextualised Software Engineering education (CSE^2).

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education

Keywords

Contextualisation, Software Engineering, education, curriculum, CSE^2, framework, culture, Namibia, Africa

1. INTRODUCTION

The discipline of Software Engineering (SE) continuously faces new challenges especially in the era of globalisation, which requires established standards and processes to be reviewed and adapted. Internationalisation and localisation efforts have gone through different phases in the past. Until today, SE processes and products have been driven by technologically advanced nations only. Most African countries merely participated as recipients and hardly influenced the direction of the ongoing evolution. Since the early 1980's western countries exported their own solutions, without the necessary adaptations or consultations regarding requirements or needs in the target context. As a result many of those projects either collapsed, or required substantial external resources for maintenance. Most systems developed or deployed in African countries are still far from being locally sustainable. Lessons learnt from the failures of early day technology transfer lead to a new movement of Internationalisation and Localisation, recognising the significance of the embedding context. However driven by western economic interests in emerging markets, the adaptations are often reduced to minimal product adaptations. In recent years a deeper understanding of the relation between software development and the target cultural context suggests an adaptation of development methods and models. This however still disregards the applications' intrinsic values emerging from the cultural context of origin and manifested in the underlying concepts of SE. Ongoing research into cultural factors in software development as conducted by Winschiers-Theophilus reveals the complexity of the interplay between the stakeholders, techniques and standards involved (Winschiers-Theophilus, 2009b). Thus only a contextual appropriation of the SE process will enable the development of acceptable and sustainable software products.

Comparing trends in Software Engineering practice and education shows similarities with major delays in the later; e.g. the very same countries, who previously received inadequate imported technology, now embrace SE education as defined by western-driven bodies, such as the ACM and IEEE (IEEE & ACM, 2004). Investigating internationally recognised academic SE curricula frameworks like IEEE & ACM (2004), it appears as if tertiary SE education can be easily standardised and distributed world-wide unaltered. "*Challenges and future trends*", as presented by Lethbridge et al. (2007) or a similar "*International View*" by Thompson (2008), in fact centre around questions in their immediate environments only. If at all, only minimal adaptations for other countries take place in terms of e.g. variations of course offerings or case study presentations. Neither the validity and adequacy of the SE concepts and methods within

ICSE '10, May 2-8 2010, Cape Town, South Africa

the context are questioned, nor the delivery in regard to students, lecturers, teaching methods and content.

In this paper we depict the importance of considering the cultural context within software development as well as its immediate consequence for SE education. We provide anecdotal illustrations of a fundamental cultural mismatch in SE education in Namibia implying the necessity of localisation. Promoting a contextual appropriation of processes, we conclude with the presentation of a generic framework facilitating the development of contextualised Software Engineering Education systems in other countries.

2. THE CULTURAL CONTEXT: AN OVERLOOKED MATTER IN SE EDUCATION

The relevance of the software deployment context is widely recognised and therefore anchored in SE practice and education. On the other hand, cultural factors, as part of the software development process, which naturally impact the final software product, have not been sufficiently formalised in theory and practice. Software Engineering methods, techniques, and concepts are continued to be taught as universals. While this may be appropriate for certain technical areas like hardware, architecture and network technologies, the opposite can been shown for Software Engineering.

2.1 The Cultural Determinant of Software Development

Software engineering consists of activities which are human centred such as planning, communication, negotiation, abstraction, modelling, evaluating, judgement, decision making, and collaboration. Underlying cultural-determined structures and processes such as values, perspectives, perceptions and communication patterns are fundamental elements of those activities. In a typical software development project, developers model a solution based on an abstraction of a perceived problem in a given reality (Floyd, 1997). The perception of a problem as well as the process of abstraction, differentiating relevant from irrelevant, depends on the individuals' perspective and cultural background. Lumpkin (1994) illustrates the diversity with an example of diagnosing illness: "*An acupuncturist may diagnose an ill person as having too much fire. An allopathic, biomedical physician may diagnose that same person as having high blood pressure. A Namibian diviner-herbalist may diagnose that person as having made a transgression against his/her ancestors. Meanwhile, a Western-trained psychologist might diagnose the person as having an anxiety disorder.* [...] *all perceive the patient's illness within different physical, social, psychological, and spiritual paradigms. Furthermore, in each case, treatment will be different, because each practitioner has diagnosed the patient according to a specific health care modality based on different cultural reality constructs.*" (Lumpkin, 1994)

In SE, western values and perspectives are embedded in practice and further propagated in education through the believed-to-be universally valid conceptualisation of Software quality criteria, established methods and metrics (Winschiers & Fendler, 2007). For example the quality criteria of usability is commonly understood to be user effectiveness, efficiency and satisfaction. The corresponding measurement is done with methods such as GOMS and Think Aloud Task solving, in terms of number and time of task completion. However previous research by the authors and colleagues has revealed that time metrics are not suitable for Southern African user groups as the conceptualisation of "usability" differs from the common definition (Winschiers & Fendler, 2007). The twofold bias of usability evaluations, one through the underlying definition of the usability criteria and two through the application of specific methods substantially contributes to the maintenance of western values in deployed software solutions (Winschiers-Theophilus, 2009a).

Thus in many African software development projects the western-trained software developer solves self-defined problems with imported methods and techniques, based on western standards and concepts, which have not been evaluated in the context. Attempts to adapt methods as part of recent localisation efforts lack a fundamental understanding of the influence of the stakeholders as well as the underlying SE paradigms. While e.g. agile or participatory methods – which are based on well-established western communication and interaction structures – are promising tools throughout all SE processes, they too need to be contextually evaluated and adjusted as necessary for successful local use.

2.2 Contextualised SE Education

Recognising that each SE project unfolds within a unique cultural context establishes a new set of requirements for SE education. "*Developing software across different cultural contexts is an art, for it being highly creative and sensitive, situational unique, and contextually self-defined, ideally leading to a synergism of the created artifact with its environment*" (Winschiers-Theophilus, 2009b). Therefore software engineers need to acquire, master, and further customise relevant skills and techniques. This can only be achieved through fundamental changes in SE training worldwide involving internationalisation as well as localisation of the education systems. SE training can no longer be teaching of "universals" but must include learning of software development process de-construction followed by a situational valid re-construction. Moreover local and international research and industry results must continuously flow into the educational system so as to ensure synchronicity between market needs and graduate skills.

Internationalisation of SE Graduates.

Today's SE graduates, independent of their personal backgrounds, must be equipped with additional skills and techniques to develop contextually adequate software. Thus the profile of SE graduates worldwide has to be complemented with skills and knowledge to account for global and localised software development and deployment.

To recognise the uniqueness of the development situation and be able to jointly re-define the development context, graduates should be familiar with ethnographic methods and cross-cultural skills in general. Ethnographic methods can enable software engineers to determine stakeholders' communication patterns and characteristics of the context. Equipped with cross-cultural skills, the developer can then facilitate culturally valid communication techniques required for the establishment of a common understanding of concepts and methods as the basis for a contextually defined development process.

Through collaboration between software engineers, psy-

chologists, intercultural trainers and educationalists extra-curricular cross-cultural training programmes for software engineers should be worked out promoting knowledge, behavioural strategies and affective aptness for intercultural encounters (Winschiers, 2001). Besides, a variety of methods should be incorporated in the courses. To ensure the ability to apply the theoretical knowledge, students should be involved in cross-cultural joint projects – preferably across geographically dispersed universities. A number of Universities have embarked into so called international studies where students spend phases of their studies in different cultural contexts. Such endeavours should be further encouraged and be integrated in SE education.

A further concern is the unavailability of local research results and industry feedback in the international literature and via the Internet, depriving international students from contextually relevant information. Therefore educators and authors worldwide should emphasis on the sharing of contextual development scenarios for the purpose of a truly international education.

Localisation of Delivery.

In specific educational systems, a successful knowledge transfer can only take place if the delivery consists of relevant content and suitable to the students in terms of teaching and learning methods. Locally relevant content must be informed by contextual research and SE projects. However educators and authors of common and internationally used textbooks still present Software Engineering concepts and methods as universals rather than contextually adaptable. For example, widely used software engineering textbooks suggest "to use prototyping as a requirement engineering method if the requirements are not clearly understood". Yet, this method is not necessarily contributing to a clarification of requirements in all cultural contexts. Imagine the Namibian setting, a historically based, oppressive colonial system, with an authoritarian and hierarchical social order contributing a great deal to a submissive uncritical attitude supplemented by a culture of respect accepting what is offered. Presented with a prototype, to clarify requirements, Namibian stakeholders acknowledge the prototype as designed by the developer team. An other striking example is the often suggested use of questionnaires to collect valid opinions informing further decisions in the development process. However once more it can be shown that in the Namibian setting the use of this method results in invalid data as the majority of participants is guided by expected answers rather than truthful answers. This behaviour is deeply rooted in a culture of conflict avoidance oriented towards listener satisfaction.

Having proved the inadequacy of certain methods in a given context, the authors have led numerous class discussions, problematising the validity of the content of used textbooks. While students fully comprehended the argument in class, in examinations or real-life situations they will nevertheless still opt for the invalid methods and concepts "because it is written in the textbook". A major paradox on the way forward with regard to method appropriation by a currently uncritical population has been described by Bidwell & Winschiers-Theophilus (2010). Meanwhile the necessity to upgrade widely used textbooks with different contextual scenarios and appropriate methods and concepts should become a priority. At the same time, the need for contextually adequate teaching methods should be explored to ensure that students are able to take correct decisions in specific situations. In the next section we present contextual mismatches, based on more than a decade of teaching experience in Namibia, to support our argument for the need of a localisation of SE education.

3. CHALLENGES IN THE NAMIBIAN SE EDUCATION

The overall importance of an adaptation of ICT and educational processes to the environment is not new. Latu & Young (2004), in the Pacific Islands, rightfully state that "*despite being scientifically developed and proved to work efficiently ICT* [education] *must be adapted to the environment in which it will be employed for the benefits of those that use it*". Though limited in terms of cultural diversity, further research by Swigger et al. (2009) also exemplifies significant effects of culture on student performance in global SE group projects.

In the following paragraphs we are sharing selected anecdotes, demonstrating contextual mismatches between the western-style teaching concepts and methods, and SE education in Namibia. Although our experiences are not intended to draw a *complete* picture of the Namibian context, they are useful in pointing out important aspects, and give raise to questions about the appropriateness of "globally valid" SE teaching methods.

Learning with Fun.

Fun activities like gaming, game development or various kinds of "plays" are often propagated as modern and valuable educational methods (see e.g. Claypool & Claypool, 2005; Mayo, 2007).

Yet having tried such approaches in Namibian SE education has proved largely fruitless: students don't seem to enjoy given tasks regardless of their "packaging". Neither competition nor considered-fun activities resulted in higher marks or increased interest in the subject area (Fendler & Meyer, 2010). Although many students enjoy individual aspects of their participation in e.g. game-based *events*, it appears these are mostly seen as highly welcome breaks from education as such.

Our assumption is that this misperception largely results from the lack of a Namibian hobby-culture. While many children in the western world are already exposed to extra-curricular activities under a "hobby" theme, this is not part of the Namibian cultures. Furthermore, competition hardly takes place in the traditional Namibian cultures, and interest-based activities without immediate benefit for the community are still the exception rather than the rule. This is especially true for low-income groups (which form a large majority of the Namibian population). In fact, local students confirmed that they would expect their studies to be hard work, rather than enjoyable activities.

Courses, programmes and degrees are therefore considered of low value if students did not have to struggle their way through. The successful incorporation of fun activities into Southern African curricula is therefore not neccessarily transferable across educational systems from different cultural contexts.

Perception of Time.

Most visitors to African countries will easily notice a fundamental socio-cultural difference: local people's perception of time. What is often referred to as "African Time" implies a very fuzzy concept not easily dealt with by foreigners. Typical scenarios are students not showing up "in time" for classes, colleagues not keeping to deadlines, or friends being late for appointments. Such incidences are intuitively often considered rude or unprofessional by people with a western background. However, such interpretations are usually inappropriate as they are based on the assumption of conscious and intentional actions. Time, in many African cultures, is simply not as an important and clearly-defined concept as it is e.g. in the western world. But although this knowledge may assist in the *understanding* of students' actions, it nevertheless creates a wide range of cross-cultural problems in day-to-day activities whenever western methods are being at least partially employed. Time management, and consequently project management as it is usually understood, is virtually absent in student projects. Assignments are often not completed in time, and the vast majority of students start given tasks during the last one or two days only.

Collaborative Project Work.

Group projects aim to equip students with life skills for their prospective working environments, where many projects are team-based and require a high degree of collaboration among peers. Thus many courses in tertiary education are heavily relying on group work. It is expected that students working jointly on the same project learn better and deliver "better" solutions through means of extensive communication, efficient distribution of sub-tasks, and complementing different competencies.

Most group projects in our SE courses, however, revealed no significant improvement regarding the outcomes as compared to individual work. Our experiences rather show an increasing amount of low quality work, with essential parts often completely missing, or not being properly integrated into the groups' final submissions. Observing groups during their work shows a distribution of *time*, rather than a distribution of *work*: all group members gather around a single PC, watching *one* individual at a time completing a part of the work. These parts, however, were hardly assigned to that particular student beforehand, neither does the work result from an agreed project plan or upon the distribution by one designated team leader. Seemingly random group members simply work for some time, with the project being passed on to the next person afterwards. Such distribution could be misinterpreted as an instance of "Peer Programming" (for programming projects): adding redundancy and decreasing the overall failure rate by ensuring that every group member has full knowledge over all aspects of the project at any given time. Yet this is not the case as this "peer work" breaks apart the moment students leave their class. Sometimes individual (usually exceptionally good) students complete most of the work by themselves, while others simply participate from the results. Sometimes limited planned distribution takes place, but students don't seem to feel committed to sticking to the resulting scope and time constraints. Thus merging of sub-tasks is almost impossible due to a lack of coordination and quality control.

Investigating the reasons for this group work behaviour, we found a fundamental cultural mismatch between western-style – individualistic – specialisation cultures, and African community cultures emphasising on e.g. redundancy and multi-skilled individuals. Strict hierarchies within many local cultures, which e.g. do not allow youngsters to speak up against elders, lead to a resistance against taking orders from people which are on the *same* (or even lower) level in this hierarchy. As this is also the case for students of similar age, within the same course or within the same year of studies, none of these students has the authority to distribute work. Neither will other students easily accept their peers' decision or suggestions. This aspect even remains when "authority" is explicitly given to one student by a lecturer, for example by assigning a project leader role to one student per group. While this approach sometimes results in short-time improved team collaboration, it usually also vapours outside of the class environment or during long-term projects.

Constructive Criticism.

Furthermore we believe that another reason for an inappropriate allocation of sub-tasks is the students' inability or unwillingness to express criticism. Having never learnt to question decisions or orders given to them (by people "higher" in their hierarchy), many students also never learnt to question their own abilities or to judge them realistically. Students can therefore hardly express their preference, capabilities or in-capabilities regarding their potential performance in a project's tasks. For some Namibian students this goes as far as them being effectively unable to tell whether they understood a particular topic or not. If in doubt, they will rather fall back to a passive pattern in which assuming to having understood the concepts appears like a natural mechanism. Yet, Namibian students with such partial and significantly incomplete knowledge will hardly ever question – and thus can hardly influence – the teaching style or methodology used by a lecturer. Furthermore, *absolute* subject knowledge by the lecturer is, for students, an immutable atomic entity that might only – if at all – be doubted by the most senior and experienced students (often from slightly different and western-influenced backgrounds). Such behaviour is often apparent in a variety of decision-making scenarios, ranging from prototype evaluation or software design considerations in the first semesters to completely inappropriate methodology choices in their final SE research projects.

Educational Background.

Another context-related dependency is the education background (e.g. secondary schools) of students. While international SE curricula usually seem to be designed with the assumption of western education systems in mind, trying to implement these programmes without a thorough investigation into the presence of necessary pre-requisites can be disastrous. As shown by Nash (2009) in a South African scenario, "[...] *it is important that adequate training and support is provided to ensure that previously disadvantaged students are not further disadvantaged by the use of technology in tertiary institutions*".

We can confirm problems similar to those identified by Nash (2009) for Namibia. Partly depending on the particular high school students attended, their knowledge in subjects essential to a good start in every IT-related curriculum are usually extremely weak. This in turn makes it necessary at tertiary level to offer subjects like English, (basic) Mathematics, or computer user skills to the students during

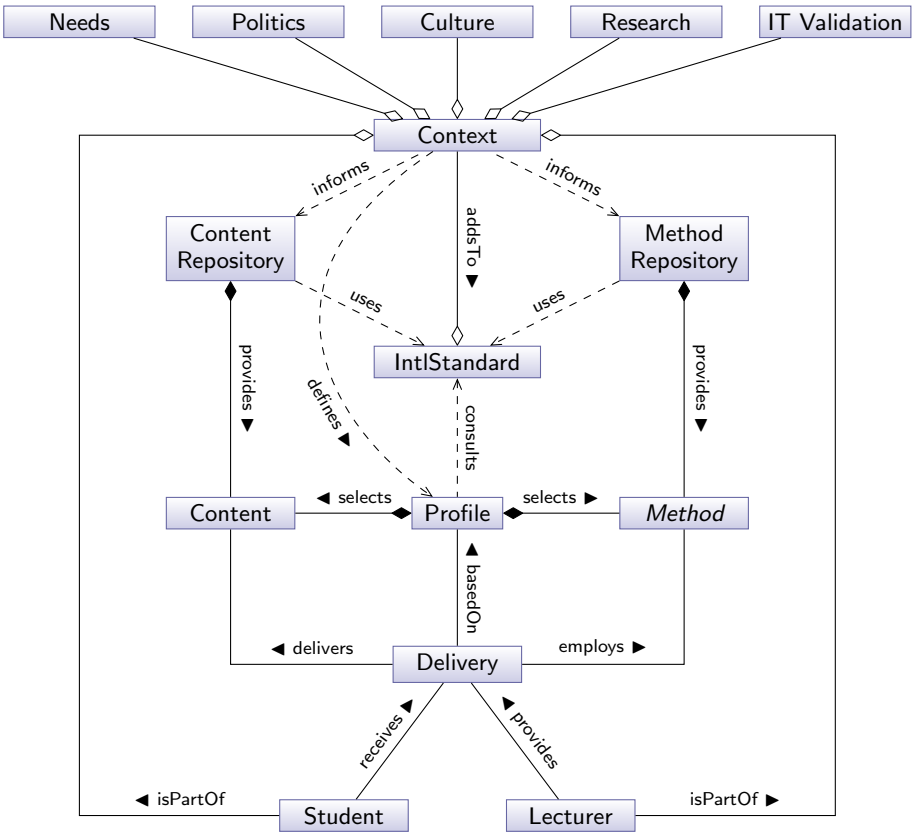

Figure 1: Contextualised Software Engineering Education (CSE^2) Framework (in UML notation)

their first semesters. Especially English, although being the national language of Namibia, is not well read or written by the majority of students. This in turn makes it exceptionally difficult for them to follow lectures, or to read up on particular subjects in literature.

The selected challenges presented here illuminate the need for fundamental revision of local SE education, while ensuring international comparability.

4. GENERIC FRAMEWORK FOR CSE^2

Based on our research in SE practice and education we challenge not only the notion of "universally transferable" education, but similarly the direct transferability of most complex IT-related processes. Rather than exporting standardised SE curricula or trying to identify singled-out factors for successful education (in highly comparable contexts), we propose the application of a generalised, context-aware process. Additionally, as with the development of software products, the educational process itself must be subject to constant review and evaluation cycles. Yet, emphasis should be put on addressing local needs, rather than satisfying an internationally agreed status-quo in SE education. Only then will it be possible to educate students specifically for local industries, whose needs often greatly differ from western requirements. Non-western training institutions should no longer consider the acquisition of SE skills as a passive fact-learning process but as an active co-construction of local knowledge by merging their cultural context with information technology. Thus besides teaching a variety of established methods and techniques a critical contextual evaluation and possibilities of localisation have to be discussed with the students. This has to be supported by local research of appropriation of methods, which in turn should be internationally published.

Addressing the discrepancies as described previously, we propose a far-reaching framework for Contextualised Software Engineering Education (CSE^2). It takes fundamental contextual differences such as political motives and agendas, local research results, cultural values, and perceptions towards technologies into account and integrates them into the SE education system throughout all levels: from curriculum planning and development, teaching methods and delivery modes, to outcome evaluation and feedback.

4.1 Overview

The CSE^2 framework as shown in figure 1 provides a generic tool to design, implement and evaluate Software Engineering education. As shown above, education takes place in a diverse context, of which relevant variables need to be considered. Our framework is thus based on the fact that e.g. Software Engineering graduates will require skills specific to the local market, as this will be the dominating environment for their future work. Furthermore, the framework includes the selection of teaching methods and evaluation, which also need to be aligned to the contextual environment. Figure 2 illustrates the curriculum development process according to the CSE^2 framework.

4.2 Context Identification

At the very beginning of every SE curriculum development, a through analysis of contextual factors is required. Figure 1 shows e.g. market needs, political agendas and motivations, cultural values, local research activities and results, and outcomes of IT validation studies. These entities have been selected based on our experiences in the Namibian context, where they have shown to be significant contextual factors. Yet it must be clear that these are only examples – neither exclusive, nor exhaustive – and need to be re-evaluated for different environments and changing society demands. While contradictions among aims of different factors are likely to occur, a weighting scheme might be useful in order to determine the most relevant contextual aspects. For example, national agendas or political programmes might favour software development high over business computing needs as expressed by industry. Programme developers should thus not only *identify* contextual factors, but extensively and transparently *discuss* those identified with all relevant contextual stakeholders.

4.3 Method and Content Repositories

Both, locally relevant content as well as locally appropriate methods are selected from respective contextualised repositories. These repositories feed from

1. the local context,including application validations (in feedback cycles), and
2. recognised international standards, such as e.g. low-level mathematical concepts, computational theories, algorithms and data structures, or standardised architectures.

While internationally standardised (or considered to be state-of-the-art) curricula clearly need to be included in local developments, their role should be an informative within a wider consultation process, rather than a primary driver of the content selection. Equally important is the definition of locally applicable teaching and learning methods. It will hardly be beneficial for students if methods which were e.g. successfully applied for a given topic in China are being integrated into non-Chinese curricula without any reflection on their local appropriateness.

It should be noted that what we refer to as *Content Repositories* and *Method Repositories* are not to be confused with *the* content being taught, or *the* methods being used within any specific educational programme. The repositories are rather to be seen as containers from which contextually suitable content and methods are selectively chosen on a per-programme basis.

4.4 Education Profile Development

In close co-operation with all relevant stakeholders – e.g. government institutions, parastatals, local key industries or alumni associations) – expected programme and course outcomes are specifically defined in Education Profiles. During the profile definitions a common understanding of, and commitment to, the overall framework and its inherent dependencies must be established among all participants. While one purpose of the profiles is to provide specific input to e.g. final programme and course outlines, their second main aim is to provide a formal instrument for later benchmarking.

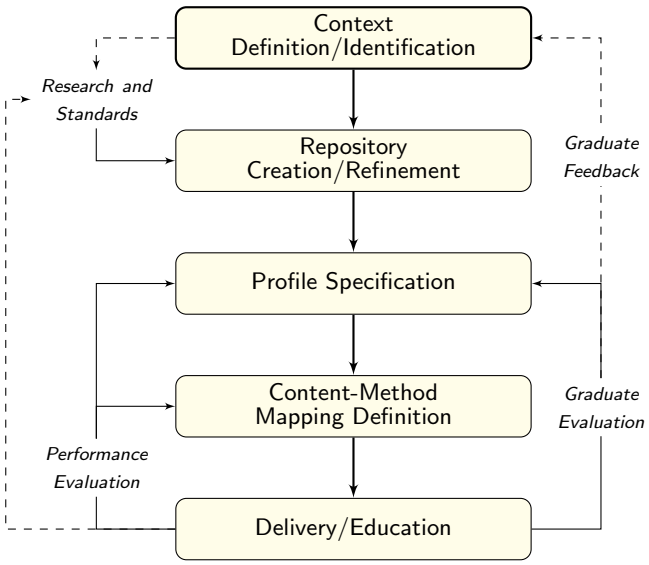

Figure 2: Curriculum Development Process within the CSE^2 framework (simplified)

While programmes are running, the profiles are used to evaluate student performance ("downward" evaluation), as well as the appropriateness of the programme itself within the local context ("upward" evaluation). Results from a downward evaluation are used to refine content-method mappings (see below), and – depending of the outcomes – to trigger and inform an upward evaluation, thereby looking into the appropriateness of the profile itself, aiming for sustainability within the context.

4.5 Mapping Content and Methods

Once the education profiles have been developed, they are used to derive suitable Content-to-Method mappings which will guide the delivery process. For every topic from the programme content (as per profile), one or more teaching and learning methods need to be chosen from the local method repository. The selection of mappings is mainly informed by three aspects: First, which of the available methods have previously been used to deliver similar content (and with which success)? Second, which of the available methods might further be suitable? And third: which out-of-context methods have been proposed (or used) for comparable subjects?

While such mappings are implicitly used in the majority of academic programmes world-wide, emphasis should be put on the necessity to select methods not only based on the content to be delivered, but specifically to take the education context into account. Therefore preference should be given to methods from the local repositories. External (or newly developed) methods can be newly integrated into the repository at this stage. However, for every alteration of the repository, relevant stakeholders need to be consulted once more, supporting method adjustments as might seem necessary. Furthermore, the delivery process with "new" methods needs to be monitored carefully and should provide significant amounts of data regarding contextual long-term applicability.

4.6 Evaluation and Feedback

During the content delivery, lecturers need to ensure not only the strict (i.e. comparable and, if possible, measurable) application of methods as per the given mappings, but also to constantly reflect on both, the content's and the the mappings' perceived appropriateness. Results of such parallel reflections are then used to provide input for feedback cycles into the repositories, as well as into the next iteration of mapping selections.

A second important level of feedback information arises from graduates. After completion of a course or programme, students are expected to apply and further broaden their knowledge within the work places. In doing so, they effectively manipulate the local context through personalised, informal feedback. However, before this alteration takes place, further evaluations of the students' performance according to the profile used, as well as an evaluation of the delivery quality needs to take place. This measurement will allow educators to adjust and improve not only the Content-Method mapping used during delivery, but also the underlying profile, should this become necessary.

4.7 Incorporating Local Research

We consider local research as one of the most important contextual factors of our framework. This is due to two facts: First, research – although conducted locally – is usually evaluated internationally (e.g. though publications and conference participation) and thus adheres to international standards. Consequently researchers can reflect on their work based on an international scale and compare results obtained locally with those from distant peers in often highly disjunctive environments. In the case of SE content or teaching and learning methods being the subject of local research, such projects allow for informed adjustments of previously chosen content-method mappings. Although international reflection of research projects can not provide exclusive means of local appropriation measurements, it does contribute great deals of valueble input to be considered by the curricula-developing stakeholders.

Second, we consider research projects as excellent vehicles of communication across different contexts. Such semi-formalised, professional information exchange is essential in the continuous revision of competitive curricula. Apart from contributing to international SE standards, it provides content and method templates for the local content and method repositories.

5. A NAMIBIAN INSTANTIATION

In this section we are demonstrating the application of the framework to the Namibian SE education. Namibia is characterised by a relatively high-developed ICT infrastructure for a population of only 2 million inhabitants. Agriculture, mining, and tourism are its main economical drivers, while a producing industry is virtually not existent. Tertiary education is mostly provided by two public universities, accepting only 400 IT students per year combined.

Following the proposed process, we start with a contextual definition. Emphasis is put on what are considered the most relevant aspects not explicitly addressed in standard curricula developments. Incorporating international standards, we then derive a high-level profile, mapping content to applicable methods and tools. The instantiation of the CSE^2 framework is then concluded with a description of mid- and long-term evaluation and feedback mechanisms.

5.1 Context Identification

Needs and Capabilities.

Namibia is generally short on ICT skills and competences. As nearly every unit in government, parastatals or private industries is in need of IT experts not only from the SE field, most IT students are being employed full-time long before they have completed their studies. Consequently the average time to graduate is significantly increased due to job-based committments. The Namibian business landscape – although dominated by few large government agencies and private enterprises – features highly diverse needs in terms of technology and customisations. As little producing industry exists the service industry – consisting of mostly Small and Medium Enterprises – accounts for the majority of SE product requirements. Although "off-the-shelve" solutions are mostly employed, such standardised products often come with little or no customisation options or specialised user training. The few locally developed software is usually conducted by South African companies or international consultants, typically unaware or unwilling to provide truly "Namibianised" versions or training packages.

Politics.

Namibian politics is currently driven by its "Vision 2030" (National Planning Commission, 2004), a long-term and high-level planning document aiming for "*P*rosperity, Interpersonal Harmony, Peace, and Political Stability". One of the goals stipulated therein is the development of a knowledge-based society, which is expected to lead to large-scale development and world-wide export of tailor-made software by the year 2030 National Planning Commission (2004). Yet this document, which effectively serves as a national benchmark for all public and parastatal institutions, hardly takes specific cultural values into account. It rather aims for universal standards (sometimes even exceeding them), but falls short in terms of guidelines on *how* to achieve these given the unique Namibian setting. Implementation details are thus largely left to individual business plans and interpretation by, sometimes foreign, decision makers or consultants.

As for tertiary education in the field of Software Engineering, the small student intake in combination with a readily absorbing IT market in the capital lead to a compromise in the quality of our education system's output. Industry and government institutions expect a constantly growing supply of high-calibre graduates which can hardly be met. As a result, programme passing criteria are relatively low, and students can hardly fail a programme finally, given sufficient financial support. At the same time, however, a significant number of applicants is turned down due to shortage of qualified staff and resources.

Culture.

The Namibian culture is heavily based on oral communication. Story telling serves as the primary mode of informal education, and conveys not only best practices, do's and don'ts, but also the key values of the Namibian peoples. Abstraction is hardly present in day-to-day life, and "learning by doing" clearly supersedes more theoretical approaches. Competitive aspects (which are often an instance of indi-

vidualism) are hardly witnessed. We believe this is due to the fact that such competition would have endangered the well-being of a group facing scarce resources and relying on highly redundant communities. Thus collective approaches with similar (and thus often similarly low) skills have outperformed attempts of specialisation and exceptional performance in the past.

Local Research.

No SE research institutes exist in Namibia. The very little progress that is being made in this regard is hosted in few departments within the tertiary education institutions, and is often lead by foreigners. Research projects in the wider area of IT are usually bound to an adaptive "Namibianisation" of existing technology, while in parallel industry is effectively setting the status-quo by employing western-developed solutions. So far, the impact of indigenous Namibian SE research is thus highly limited.

As students are therefore hardly exposed to research outside of their own Bachelor or Master theses, we believe it is of great importance to promote and incorporate SE research in the majority of our courses. Direct and indirect feedback from students can then help to refine the methods applied and to further expand the students' research capabilities not only in a university environment, but possibly already at high-school level.

IT Validation.

As there is a extremely low number of software engineering professionals in the country, a true evaluation of the validity of IT in the local context is a major challange. Care must be taken as not to create a too strong industry-bias in our curricula, e.g. through industry advisory boards. While obviously graduates must be well-equipped for the local market, Namibia will hardly reach its goals for 2030 by exclusively relying on the generation of short-term profits (i.e. through an exclusive use of western-designed software). To promote custom SE projects, and to create successful, locally operating, software development companies, a critical mass of highly educated and specialised SE professionals is essential in order to provide local solutions with a high degree of local appropriation and thus added value.

5.2 Profile Definition

In recognition of the specificity of the Namibian context, the development of certain skills should be particularly emphasised. Considering the low number of local Software Development companies and staff, the graduates should be cross-functional and familiar with different technical fields such as databases, Internet programming and mobile technology. Most graduates are employed at middle-management level, with decision powers regarding tenders, requirements specifications and software evaluations. Thus students should have communication and negotiation skills, as well as contextual awareness. Emphasis in the curriculum should further be on re-factoring, re-engineering, customisation, evaluation, quality assurance and project management. Suitable teaching methods in the Namibian context include semi-structured group projects, high-frequency written and oral presentation of intermediate results, class discussions, peer evaluations, and most of all real-life projects supervised by and jointly coordinated with industry partners and academics.

5.3 Evaluation and Feedback

Evaluation of the student skills can be done with traditional means such as examinations, project evaluations and presentation evaluations by peers, lecturers and external partners.

One important feedback mechanism are the industry partners, who employ graduates and evaluate their skills at the workplace. Identified gaps can be channeled back through advisory board meetings and informal channels. Alumnis are also currently involved in the review of curricula having undertaken studies at the University and collected working experience they are now able to give valuable feedback. Students continuing studies at other Universities oversees also frequently send feedback in terms of their knowledge gaps.

6. CONCLUSION

Methods, processes and concepts in the discipline of Software Engineering are neither constant nor universal. In order to deliver successful and sustainable systems in different environments, contextual dependencies have to be identified and taken into account. Graduates from Software Engineering programmes need to acquire skills that allow them to work in any environment – regardless of contextual constraints and boundaries, or culture-based perceptions and assumptions. Cross-cultural values have to be negotiated and adapted within global SE projects. Educators, in turn, must ensure that the content and delivery methods suit not only the teaching and learning environment, but also the wider society.

Addressing the above mentioned points, we have developed and proposed a Contextualised Software Engineering Education framework. Curricula, teaching methods and evaluation criteria developed in accordance with the CSE^2 framework are locally adequate while still embracing international trends. They are expected to produce a significant increase in terms of well-equipped SE graduates, as well as sustainable and acceptable software products for international *and local* markets.

It is our hope that the CSE^2 framework will contribute to fruitful discussions and further research in the international community, to promote and actively support true contextualisation of not only software products and engineering, but also of Software Engineering education.

7. ACKNOWLEDGEMENTS

We would like to thank all Software Engineering students and alumni at the Polytechnic of Namibia, as well as our colleagues in the School of Information Technology.

So far, we often only perceive the mismatch in our attempts to teach Software Engineering, and can only try to imagine the difficulties for students trying to grasp novel concepts with often inappropriate methods.

However, we hope to be on the right track in our attempts to understand the underlying differences, and to come up with working and sustainable solutions helping not only students in Namibia but in other countries as well to become IT professionals in *their own* ways.

References

Bidwell, N. J. & Winschiers-Theophilus, H. (2010). Beyond the Benjamins: Towards an African Interaction Design. *ACM Interactions*, *17*(1), 32–35.

Claypool, K. & Claypool, M. (2005). Teaching Software Engineering Through Game Design. In *Proceedings of ITiCSE*, (pp. 123–127)., Monte De Caparica, Portugal.

Fendler, J. & Meyer, M. (2010). The PrimeGame: Combining Skills in Undergraduate Computer Science Programmes. In *Proceedings of the Int. Technology, Education and Development Conference 2010*, Valencia, Spain. To be published.

Floyd, C. (1997). Autooperationale Form und situiertes Handeln. In *In Cognito Humana XVII*, (pp. 237–252)., Leipzig. Deutscher Kongress für Philosophie, Akademie Verlag.

IEEE & ACM (2004). Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programmes in Software Engineering. Published by The Joint Task Force on Computing Curricula. IEEE Computer Society and Association for Computing Machinery. `http://sites.computer.org/ccse/SE2004Volume.pdf`.

Latu, S. & Young, A. (2004). Teaching ICT to Pacific Island Background Students. In Lister, R. & Young, A. (Eds.), *Sixth Australasian Computing Education Conference*, volume 30 of *Conferences in Research and Practice in Information Technology*, (pp. 169–175)., Dunedin. Australian Computer Society.

Lethbridge, T. C., Diaz-Herrera, J., LeBlanc Jr., R. J., & Thompson, J. B. (2007). Improving software practice through education: Challenges and future trends. In *Future of Software Engineering '07*. IEEE Computer Society.

Lumpkin, T. W. (1994). Traditional Healers and community use of traditional medicine in Namibia. Namibian Ministry of Health and Social Services and UNICEF.

Mayo, M. J. (2007). Games for Science and Engineering Education. *Communication of the ACM*, *50*(7), 31–35.

Nash, J. (2009). Computer Skills of First-Year Students at a South African University. In *Proceedings of SACLA '09*, (pp. 88–92). ACM.

National Planning Commission (2004). *Vision 2030. Prosperity, Harmony, Peace and Political Stability*. Windhoek, Namibia: Namibian National Planning Commission Secretariat. Last accessed January 26, 2009 from `http://www.npc.gov.na/vision/vision_2030bgd.htm`.

Swigger, K., Alpaslan, F. N., Lopez, V., Brazile, R., Dafoulas, G., & Serce, F. C. (2009). Structural Factors that Affect Global Software Developmnt Learning Team Performance. In *Proceedings of SIGMIS-CPR '09*, (pp. 187–195)., Limerick, Ireland. ACM.

Thompson, J. B. (2008). Software Engineering Practice and Education. An International View. In *Proceedings of SEESE '08*, (pp. 95–102)., Leipzip, Germany. ACM.

Winschiers, H. (2001). *Dialogical System Design Across Cultural Boundaries*. PhD thesis, University Hamburg.

Winschiers, H. & Fendler, J. (2007). Assumptions Considered Harmful – The Need to Redefine Usability. In Aykin, N. (Ed.), *Usability and Internationalization. LNCS 4559*, (pp. 452–461). Springer Verlag Berlin Heidelberg.

Winschiers-Theophilus, H. (2009a). *In Whitworth, B. (Ed.): Handbook of Research on Socio-Technical Design and Social Networking Systems*, chapter Cultural Appropriation of Software Design and Evaluation. IGI Global.

Winschiers-Theophilus, H. (2009b). The Art of Cross-Cultural Design for Usability. In *Proceedings of the 13th Intl. Conference on Human Computer Interaction*, San Diego, USA.