A Practice-based Systems Engineering Programme

Duarte Gonçalves
+27 12 841 3963
dgoncalv@csir.co.za
Council for Scientific and Industrial Research (CSIR)
P.O Box 395
Pretoria, South Africa
0001

Abstract. South Africa, and indeed internationally, has been experiencing a shortage of systems engineers. On the other hand we seem to have only introductory systems engineering courses at local universities. Systems engineers have developed by means of experience on the job. This is a long route fraught with many challenges.

We look to develop a strategic solution to the problem in this paper. We start by considering a number of reasons why systems engineering is difficult to learn. A framework for defining the required system engineering competencies is introduced. A practice-based approach is presented as part of the solution, including the roles of universities, students and industry within this approach. Finally we elaborate on a proposed curriculum for a practice-based SE educational programme.

The shortage of systems engineers requires strategic action. In order to accelerate the development of high competency systems engineers we will need to adopt new (old) approaches. What is being proposed will require considerable effort, but is expected to yield good results and will contribute to developing the next generation of systems engineers.

Introduction

Systems engineering is a critical capability for engineering large, complex projects in the South African industry and specifically in the Defence and Aerospace industries. However, systems engineers with the appropriate levels of competence are in short supply, not only in South Africa, but internationally. One of the reasons for this is that systems engineering is difficult to teach at universities using traditional models. Systems engineering requires both explicit knowledge (which can be taught at a university) and tacit knowledge (like skills, judgement, which are learnt through doing). Another issue is that unless students have experienced some of the problems that can occur on large development projects, students will not learn in a meaningful way. So, for the large part, systems engineers have developed through experience and this can be a lengthy process.

The systems engineering world as seen through the International Council on Systems Engineering (INCOSE) is moving from document-centric approaches to model-based systems engineering (MBSE). Many international projects are using MBSE and tools are widely available today. MBSE is seen as a growth area in INCOSE's vision for 2020. So, in addition to the learning issues, emerging developments in architecture frameworks in combination with MBSE requires a curriculum that prepares a new bread of systems engineers.

In order to address the learning issues, we propose developing a practice-based systems engineering (PBSE) programme as a joint collaboration between universities and industry: Universities present the theoretical background and participating industry organisations provide opportunities to apply material on real projects and work under an experienced systems engineer (coach).

What we are proposing has been done elsewhere in the world. An ex-South African, Alistair Campbell, has been involved in setting up a similar programme at the University of South Australia in collaboration with the Australian industry (Campbell and Cropley, 2009). In Europe, EADS is a lead user of a practice-based training platform that not only uses a practice-based approach, but also seeks to develop collective skills (Fournier et al. 2010). There are also parallels with the field of medical education where a number of models, such as the SPICES model, have been developed which are also practice-based (Gonçalves, 2008).

Since the model being proposed represents a change from traditional teaching models, a pilot programme may be required to understand the implications for universities, industry and students. Although the CSIR has initiated the programme, the intention is to expand it to the broader industry.

This paper presents the plan for a PBSE programme. To this end, the next section describes practice-based SE (PBSE) education in more detail starting with why SE is difficult to learn. This is followed by the requirements and considerations for a PBSE course. A SE curriculum is proposed, aimed at SE for developing systems in the early phases of the systems life-cycle.

A Practice-based Approach to SE Education

We discuss SE education in the context of achieving engineering work objectives, i.e. delivering successful, working systems. Learning and development are seen as secondary goals that are to be achieved on the way to work objectives. Bobbit, in his seminal work produced in 1918 (Bobbit, 1918), argues for “work-activities as the only possible normal method of preparing for the work of the world”. It is unlikely that employers would want to take on the job of ‘preparing’ every employee in general, SE has some significant differences (discussed blow) which requires a ‘new’ approach. Bobbit elaborates further saying that the student “…examines every fact and principle in relation to his practical problem, and not merely as a field of intellectual sight-seeing”.

Requirements relating to a new programme concerned with the development of SE competence, raised in (Goncalves, 2008), are:

1. A new SE programme must transfer both explicit and tacit knowledge components of knowledge relating to SE. Transferring tacit knowledge is more difficult than transferring explicit knowledge, requiring approaches that depart from traditional engineering education.
2. The programme must provide a learning context in which SE will be applied. The industry, the types of products being developed and level of these products in the systems hierarchy and the life-cycle phase *of real projects* define this context. The context is essential for *situated learning*.
3. Three different levels of learning: *individual*, *group* (or team) and *intra/inter-organisational*. The last two levels require participation (social interaction which includes ‘seeing’ and *doing*). Many current programmes in South Africa are focused on the individual level.
4. Students wishing to participate on the programme must have sufficient ‘absorptive capacity’. This is prior knowledge that allows SE knowledge and skills to be properly assimilated. As a guideline, students should have at least 3 years of engineering experience.
5. A new programme needs to be student centred because *it does not matter what is taught, only what the student learns*.

A practice-based SE course, in conjunction with screening of students, is believed to address these requirements. It is characterised by at least three things: situated learning,

learning that can take place at the three different levels: individual, group (or team) and organization, and tacit knowledge can be transferred. Explicit knowledge would be transferred by means of lecture modules in conjunction with group exercise. Thus the foundation is built on SE *principles* (not just 'experience'). The fourth, one could argue, is a potentially higher level of emotional involvement, which determines the level of programme success.

Why do we need this? In order to answer this question, a five levels model of SE education is proposed in Table 1. At the highest level, 'why', relates to purpose and typically the domain of leadership, 'when' is judgement regarding tailoring (to some extent the purpose can also be important in this regard), 'what' being the process or knowledge for achieving the purpose and 'how' is linked to skills. One of the issues with SE Standards is that they deliberately avoid the 'how' level exactly because it is dependant on the industry and product under development and the life-cycle phase. The 'who' describes the requisite characteristics of the person or team, an aspect receiving attention locally and internationally (Gonçalves and Britz, 2008).

Table 1: The WWWHW model for SE learning

Why	Purpose
When	Judgement
What	Process (knowledge)
How	Skills
Who	Personal Characteristics

WWWHW model is a framework that assists in identifying the various SE aspects to be learnt. At least three levels of consideration are proposed (Table 2): Systems engineering (global level), process level, and an analysis level. The number of levels is determined by the complexity of the project and system requirements. If we consider the process level, then it is the 'what' of the SE level, with some of the 'how' being considered at the analysis level (a similar model is used by Vincente). Courses may not focus on all the knowledge levels and also not address sufficient levels of consideration. There are some that deal with the 'how', but few are able to cover all levels.

Table 2: Adding the level of consideration to the WWWHW model

Knowledge Level	**Level of consideration**		
	Global	**Process**	**Analysis**
	Systems engineering	Requirements Analysis	Behaviour Analysis
Why			
When			
What	Analyse requirements		
How		Behaviour Analysis	
Who			

There are a number of challenges in implementing a PBSE programme. Firstly, engineers are away from work while they are doing theory. We need to do this in a way that minimises impact on work. Secondly, we need to ensure systematic practice. In other words, we need to have either a variety of projects or a single large project where the student can practice over the required variety of SE competencies. This may not always be available at one company. The other concern is that security and company confidentiality may hamper the programme. Also, the delivery of theory needs to be synchronised to practice in order to close the theory-practice loop. This may make scheduling challenging.

One proposal to address some of these issues is to partition the course into small modules that covers theory and practice-based learning in a specific module. There could be a basic module that covers SE introduction and a number of other modules, possibly along process lines. The advantage of this approach is that we can deliver, for example a requirements analysis module, when it is needed on a project. The student is only away for the duration of one module. The practice-based part of the module can be linked to the project that required the competency. We are not attempting to cover an entire SE course all at once. The assumption here is that there are large projects or sufficient smaller projects to get the practical exposure. Once sufficient modules have been completed, the course is considered to be complete. If the courses are offered largely on a student-centred model (Gonçalves, 2008) then we need to be more flexible on when we present the modules. In the next section we look at the stakeholders of a PBSE programme.

Stakeholders of the PBSE Programme

There are four main categories of direct stakeholders of the PBSE programme (discussed below): the universities, industry, students, some of whom would be at a university or in industry, and the INCOSE SA Chapter.

DPSS, a unit of the CSIR, is taking the lead on developing the programme. While the CSIR as a science council is not part of industry, we will include it under industry as an employer of systems engineers for the purposes of this discussion. It is intended that the PBSE programme be extended to the broader industry to mitigate the shortage of systems engineers in South Africa as a national initiative. Based on interactions with a number of industry organisations there is broader interest beyond just DPSS. However, this interest would need to be developed further.

At least three categories of students could be considered: undergraduate students, post graduate study immediately after the first degree and study after an initial period of approximately three years of industry experience. Because of 'absorptive capacity' considerations (discussed in Gonçalves, 2008), the PBSE programme will not consider undergraduate students. Characteristics of students that should be taken into account are described in Table 3.

Table 3: Characteristics of students considered for PBSE

View	**Study straight after B degree** **Age < 25 years**	**Study as a working student** **Work experience >3years**
Financial	Low income	Steady junior eng salary
Experience	Little to no work experience Fresh experience as a student	Organisational, technology and project experiences
Project access	Access may be limited or somewhat artificial	Ongoing access to projects
Teams	Individual responsibility	Team delivers
Coach	Access may be limited	Ongoing access to coach, although may not always be available.
Theory	Good access to the extent that the SE skills are available at a university.	Limited by pressure of work. May not get a good framework.
Time	Attending classes 2-3h/weekday	Working typically 8h/weekday

View	Study straight after B degree Age < 25 years	Study as a working student Work experience >3years
Motivation source	Self-motivated	Self-motivated, work related motivations such as bonuses, building a career.

Indirect stakeholders are Armscor, the South African National Defence Force and other clients who require the skills but may impose security requirements. Another indirect stakeholder would be the accreditation institutions.

The INCOSE, South African Chapter represents the interests of systems engineers in South Africa and by association also those of their employers with a number of these having a need for developing SE competencies.

Requirements for a PBSE Programme

In this section, the need for systems engineers is defined in terms of SE competencies and the ability to deal with problems and solutions, objectives and goals. Some requirements that enable learning and organisation specific requirements are identified. Roles and responsibilities are proposed. Matters relating to certification and accreditation, projects, supervision and assessment of students, intellectual property and security are briefly discussed. The section concludes with a concept for the PBSE programme roles and responsibilities.

Need

Two frameworks are used to define the need in terms of SE competencies and the ability to deal with problems and solutions:

1. for competencies, INCOSE UK Systems Engineering Competencies Framework (INCOSE UK, 2006)
2. for the ability to deal with problems and solutions, Kasser et al. five type of SEs (2009).

The industry need is to develop high-competence systems engineers. Competence requires knowledge, skill and the psychological characteristics. We propose using the INCOSE UK Systems Engineering Competencies Framework (INCOSE UK, 2006) which defines 21 competencies (Table 4) and 4 levels of competence: awareness (A), supervised practitioner (SP), practitioner (P) and expert (E). High-competence is defined as practitioner or expert level. The priority for developing each competency is either high (H), medium (M) or low (L). The DPSS priorities are requirements analysis (more broadly than just requirements management) and architecture. It is likely that these two competencies would be a priority across industries, but expect priorities of other competencies to vary across industries. The DPSS priorities for other competencies are preliminary. Modules do not need to be structured along the lines of competencies. In fact, we may want to structure modules along broad process lines or life cycle while avoiding industry specific terminology. The basic concepts should be applicable across industries, but this would need to be validated (a research project relating to this topic is in the pipeline). Students would need to pick a set of 4-6 competencies that they would focus on based on the industry organisation.

Table 4 SE Competency Requirements for DPSS

Category	Competency	Competence Level	Priority
Systems Thinking	System Concepts	P	H
	Super System Capability Issues	P	H
	Enterprise & Technology Environment	P	M

Category	Competency	Competence Level	Priority
Holistic Lifecycle View	Determining and Managing Stakeholder Requirements	P	H
	Systems Design – Architectural Design	P	H
	Systems Design – Concept Generation	P	M
	Systems Design – Design for…	SP	M
	Systems Design – Functional Analysis	P	H
	Systems Design – Interface Management	P	H
	Systems Design – Maintain Design Integrity	P	M
	Systems Design – Modelling & Simulation	P	M
	Systems Design – Select Preferred Solution	P	H
	System Design – System Robustness	SP	L
	System Integration & Verification	P	M
	Validation	P	M
	Transition To Operation	SP	L
Systems Engineering Management	Concurrent Engineering	SP	L
	Enterprise Integration	SP	M
	Integration of Specialities	SP	M
	Lifecycle Process Definition	SP	M
	Planning, Monitoring & Controlling	P	H

However, this competency framework has some issues. It is focused on developing systems where the requirements are largely developed. At DPSS, we need some of these skills, but also the ability to define the problem. The INCOSE UK Framework is limited in the area of requirements analysis. It refers to functional analysis, but this is actually functional design. Functional analysis (uses the same notation, but with different rules), which is done as part of requirements analysis, is not considered. Problem definition will not be fully covered initially but is a critical skill for systems engineers working in the early system lifecycle phases. Five types of SEs can be defined based on their ability to deal with problems and solutions (Kasser et al., 2009):

Table 5 Five Types of Systems Engineers

Type I	This type is an "apprentice" who can be told "how" to implement the solution and can then implement it.
Type II	This type is the most common type of systems engineer. Type II's have the ability to use the systems engineering process to figure out how to implement a physical solution once told what conceptual solution to implement. Most systems engineers fall into this category.
Type III	Once given a statement of the problem, this type has the necessary know-how to conceptualize the solution and to plan the implementation of the solution.
Type IV	This type has the ability to examine the situation and define the problem.
Type V	This type combines the abilities of the Types III and IV, namely has the ability to examine the situation, define the problem, conceptualise the solution and plan the implementation of the physical solution.

Type I is a transitory educational level. The main DPSS requirement is for SEs at levels III to V, with the main focus on type IV. This is consistent with the fact that DPSS engineers small numbers of products, but its main focus is on feasibility part of the system life-cycle. This will shape the curriculum.

Both the CSIR and the universities are required to produce research outputs. There are three categories where research outputs could be produced:

1. Development and evaluation of a PBSE programme
2. Systems engineering, and
3. Project related research.

SE research would constitute new methods, etc. While it is not a requirement of this programme to produce such outputs, should they be produced, they can be published. It may be possible to publish project related research, but this would need to be discussed between the University and the industry organisation on a case-by-case basis subject to intellectual property and security considerations.

Organisation Specific Requirements

It is likely that each industry organisation will have specific requirements. For example, DPSS may screen candidates in terms of psychological characteristics. Transformation goals are also important for DPSS as a government funded organisation. It is foreseen that universities might also want to utilise the SE modules as part of other programmes, where DPSS is for example looking more towards certificate courses.

Roles and Responsibilities

The roles and responsibilities presented here are based on competence in the various areas and match with the organisation's mission. Universities would be best suited to presenting the SE principles (if lecturers with the skills are available), evaluating the students and accrediting the course. The universities have long-standing experience in certification and accreditation of students and courses respectively. We are not currently looking for SE researchers (although this is a longer term consideration), rather systems engineers with theoretical grounding and skills. While a Masters degree could be used as a mechanism, a certificate course might be preferred. The best course of action appears to be university specific and on this issue we will follow the university's preference. Two certificates may be required (it would be necessary to clearly distinguish these):

- Although we are trying to establish a practice-based approach to SE education, this might not be practical for some small companies or companies where there is no culture of systems engineering. In these cases, the theory only could be offered as a certificate.
- A certificate for the full practice-based approach, including the theory.

Whatever mechanism is chosen, the programme would need to be accredited (with for example, the South African Qualification Authority) following the pilot programme in order to increase industry acceptance and ensure quality.

The industry organisation would provide *real* projects, a SE coach to guide the student and a suitable and stable environment for learning. SE work will be done in the context of a live project with defined deliverables. Students should not be leading such projects if there is project risk and should work with a coach from the industry organisation. The student develops SE competencies by applying SE principles on real projects within a team and organisational context. Working as part of a team leads to development of social skills required for SE. The university supervisor, the coach and the student's immediate supervisor or project manager perform the assessment of the student with defined roles led by the university supervisor. To ensure a consistent standard across industry, we would need to define module level objectives to be achieved in practice. Students would be assessed on these as proposed in the following section.

Development of the PBSE programme pilot programme might be done jointly by universities and DPSS. An alternative being explored is to re-use suitable material from other courses. Each module (discussed in the following section) is presented by a competent lecturer, whether from industry or a university.

Proposed PBSE Pilot Curriculum

This section provides an outline of the PBSE modules and how these map to the required SE competencies.

Outline of SE Modules for the PBSE Programme

An overview of the PBSE modules envisioned is presented in Figure 1. The pilot will focus on the SE core (mandatory first module), requirements analysis (RA) and architecture module. An SE management module outline has been developed although it is not currently intended to be used as part of the pilot. Modelling and simulation, integration, verification and validation and specialities will follow once the practical issues of a practice-based approach have been resolved. Building the foundation for MBSE will be a central theme all the modules. Many international projects are using MBSE and tools are widely available today. We will delve into those modules that are directly relevant to the pilot bearing in mind the why, when, what and how parts of the framework introduced in the next sub-section.

The focus of the modules is on material that has broad applicability. For example, SE standards represent best practice. Companies will select a certain standard over another. The modules should endeavour to give an overview of such areas, but cannot address such material in detail. It would be better for the student to learn this within his/her company context, including company specific tailoring.

A number of sources of information were consulted in compiling this curriculum:

1. Literature (Kasser 2007, Squires and Cloutier 2009)
2. Other programmes (PPI Course notes, MIT 2009, University of South Australia, 2009).
3. Personal experience.

This programme does not fully address problem definition, which occurs before RA (this will be addressed later). While some of the approaches that are used in RA can be applied, additional tools may be required. Other areas such as designing the developing organisation and enterprise integration will also need to be developed.

The Core Module

The first order of business of the core module (Figure 2) is to present the objectives, principles and overview of PBSE course. The practice-based format will be new to students, so the roles and expectations of the university and those of the employer will need to be stated.

The SE core module seeks to introduce the basic concepts and motivation for applying SE. The module starts by looking at why projects fail. This is the reason for the existence of SE and leads to its purpose. Basic concepts must be introduced, for example, “What is a system?”, “What is a system life-cycle?” and “What is systems thinking?”. These themes will be reiterated through the other modules. SE principles should be covered in this module along the lines of (PPI course notes), for example: Capture and understand the problem before committing to the solution. Modelling notations for SE are introduced in the core module. These representations support understanding, reasoning and communication about the system, which are fundamental issues in SE. This is true not only for technical aspects of engineering but also for management aspects such as planning. Architecture frameworks not explicitly presented in the module because these are application specific. However there is an implicit architecture framework underlying the modelling notation discussed in this module. The two fundamental viewpoints are behaviour and structure – essential in understanding architecture frameworks. The criteria for selecting a modelling notation needs to be presented – syntax and semantics (expressiveness), rigor and understandability (Buede, 2000), before a number of modelling notations are introduced.

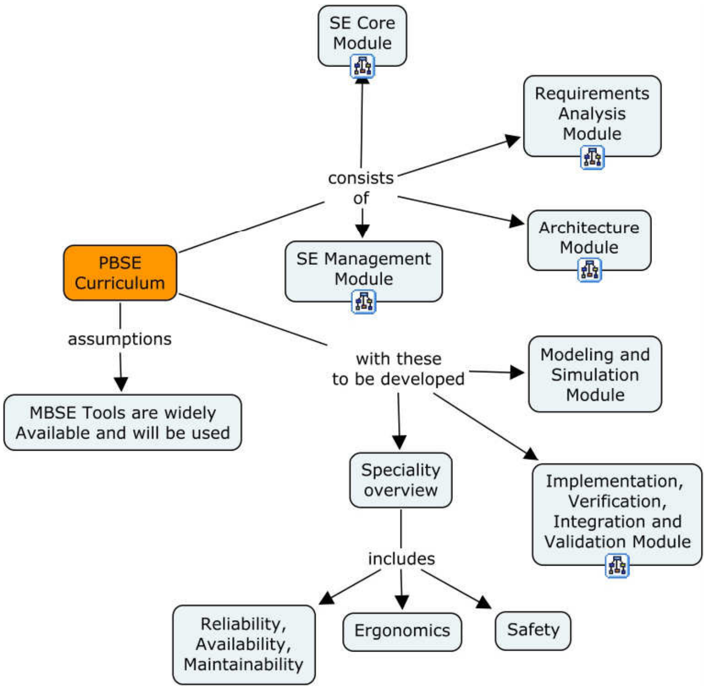

Figure 1: PBSE - Curriculum overview

Requirements Analysis Module

One of the larger modules will be requirements analysis (RA) – this is not out of line given the importance of requirements. Starting with the purpose of RA, the requirements process and requirements types need to presented. An area which needs some attention is elicitation techniques, using scenarios for example, and sources of requirements. Considerable effort is spent on techniques for RA, including the purpose and applicability of each technique. These are essential to defining the problem before any specifications are written. Students will need to develop the discipline of separating the problem from the solution. The characteristics of good requirements (requirements quality) should be addressed in conjunction with writing specifications. Managing RA ranges from planning a RA effort to creating traceability to stakeholders and operational concepts. A healthy dose of emphasis on iteration is required.

Product scoping, as proposed by (Hooks and Farry, 2000), may be very useful in the context of RA to create a common vision, draw a boundary as to what is or is not a requirement and a tool for gauging the size of the effort.

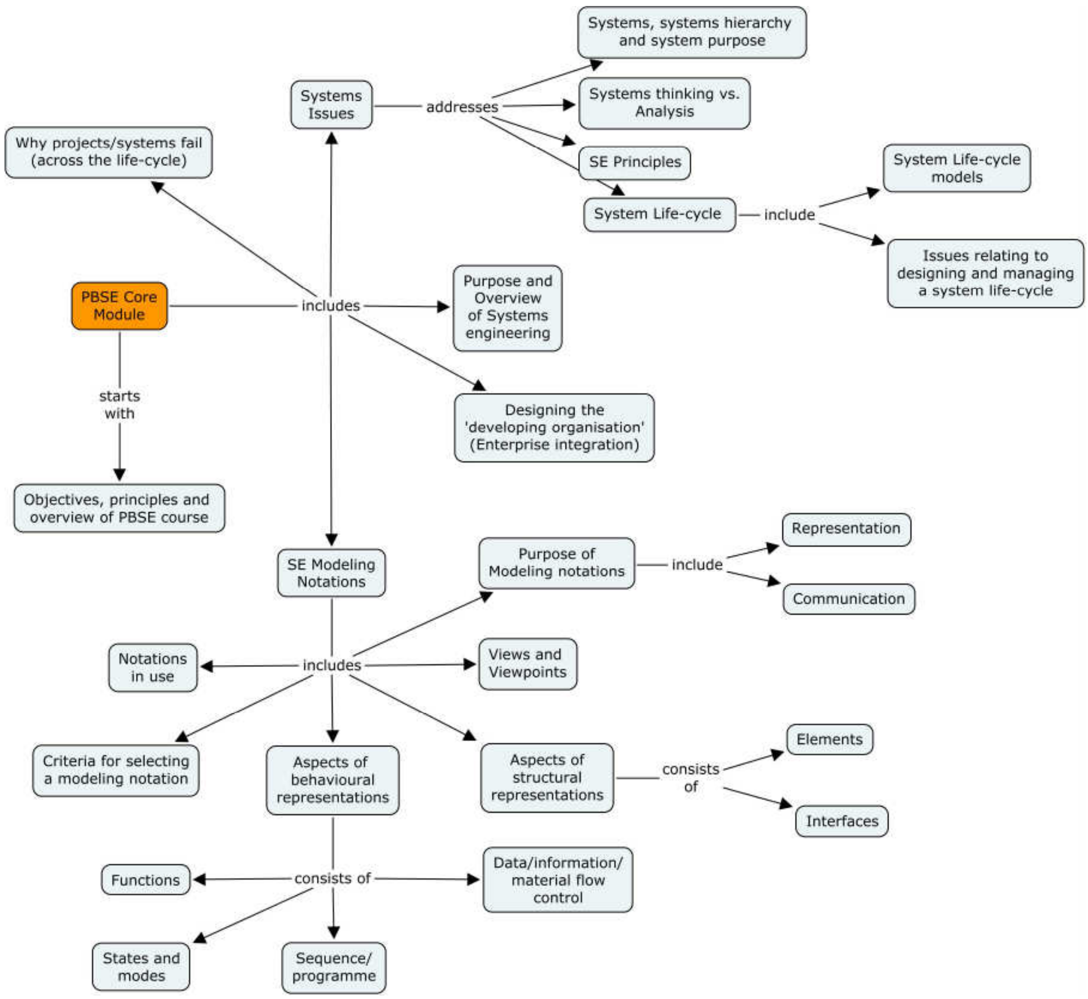

Figure 2: Core (mandatory first module) SE module

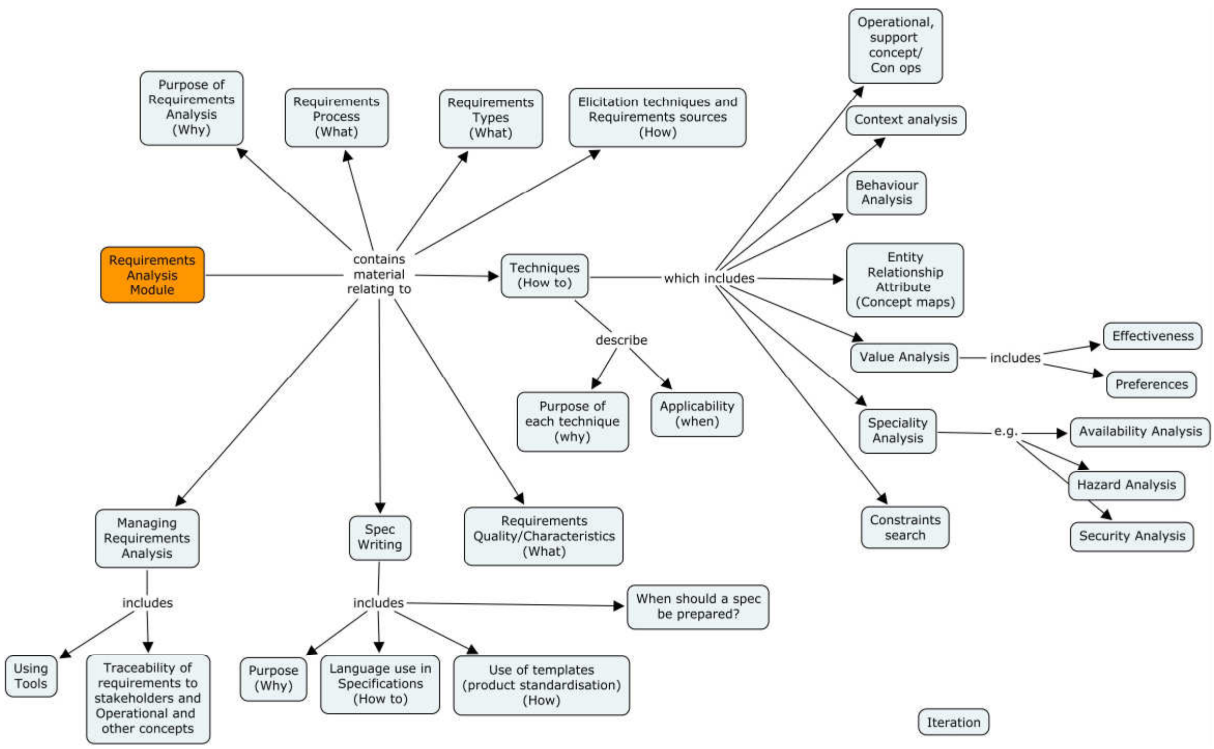

Figure 3: Requirements Analysis Module

Architecture Module

An overview of the architecture module is presented in Figure 4. Architecture is not a mature field with a widely accepted underlying theory. For this reason, a number of approaches to architecture need to be presented. This depends on whether these are software, or hardware and the specific type of hardware systems e.g. largely signal processing, like radar. The importance of identifying the drivers of architecture early on needs to be communicated to students (why, what and how). Key to architecture is creativity, dealt with in concept generation. Alternatives need to be generated both at the system level and at function level. Concept generation is supported by behaviour analysis (part of which is functional analysis). The architecture module would need to cover both the development of structural (physical) and behavioural aspects of architecting. Interfaces would be dealt with as part of the structural architecture. Concepts relating to the development of alternatives, the evaluation of these and the selection of candidate architectures needs to be presented. The issue of traceability from requirements, functions and allocation to system elements needs to be covered. The concept of technical budgeting, supported by modelling and simulation needs to be introduced. Technology as the basis of any solution and the concept of technology maturity need to be presented. Again a healthy dose of emphasis on iteration is required.

As a final point, two aspects of life-cycle need to be emphasised (not illustrated in Figure 4):

1. The architecture needs to address the system life-cycle (during development), and
2. The architecture will evolve over the system life-cycle (as new requirements emerge and other change).

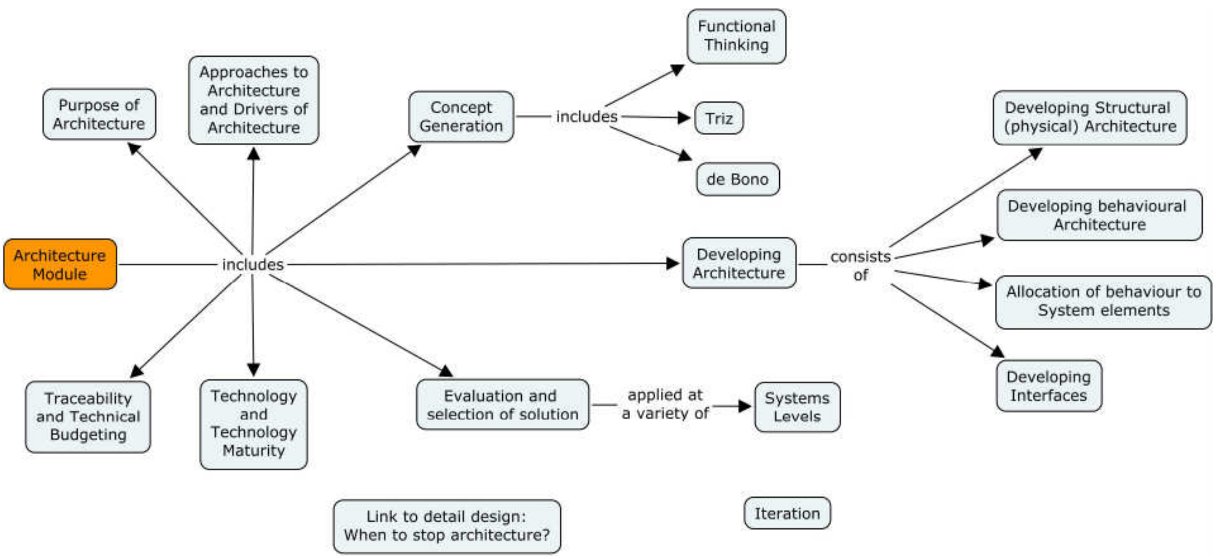

Figure 4: Architecture Module

SE Management Module

Although not initially planned as part of the pilot, SE management module may be required (Figure 5). This module introduces risk management, configuration management, technical performance management, concurrent engineering management and speciality management. For all of these, the purpose and what needs to be done and how it can be approached should be presented.

SE planning receives considerable attention in this module. The diagram shows the planning for the development phase of the life-cycle. Planning for other life-cycle phases (production, transition to operation, operation and support, disposal) will need to be introduced. The emphasis should be on how to identify, based on technology maturity and other requirements, the life-cycle phases.

Planning (in the development part of the lifecycle) deals with defining the processes to be followed, defining the SE products that will be produced (documents, models, etc.) and allocating responsibilities. How the processes will be sequenced is defined by the development model based on considerations such as risk, etc. The development strategy is dependant on application maturity (low maturity leading to an evolutionary approach) or technology maturity (incremental).

Finally, the relationship between SE and project management needs to be discussed. On unprecedented projects, it is especially difficult to cost the development before the first cut RA and architecture have been competed.

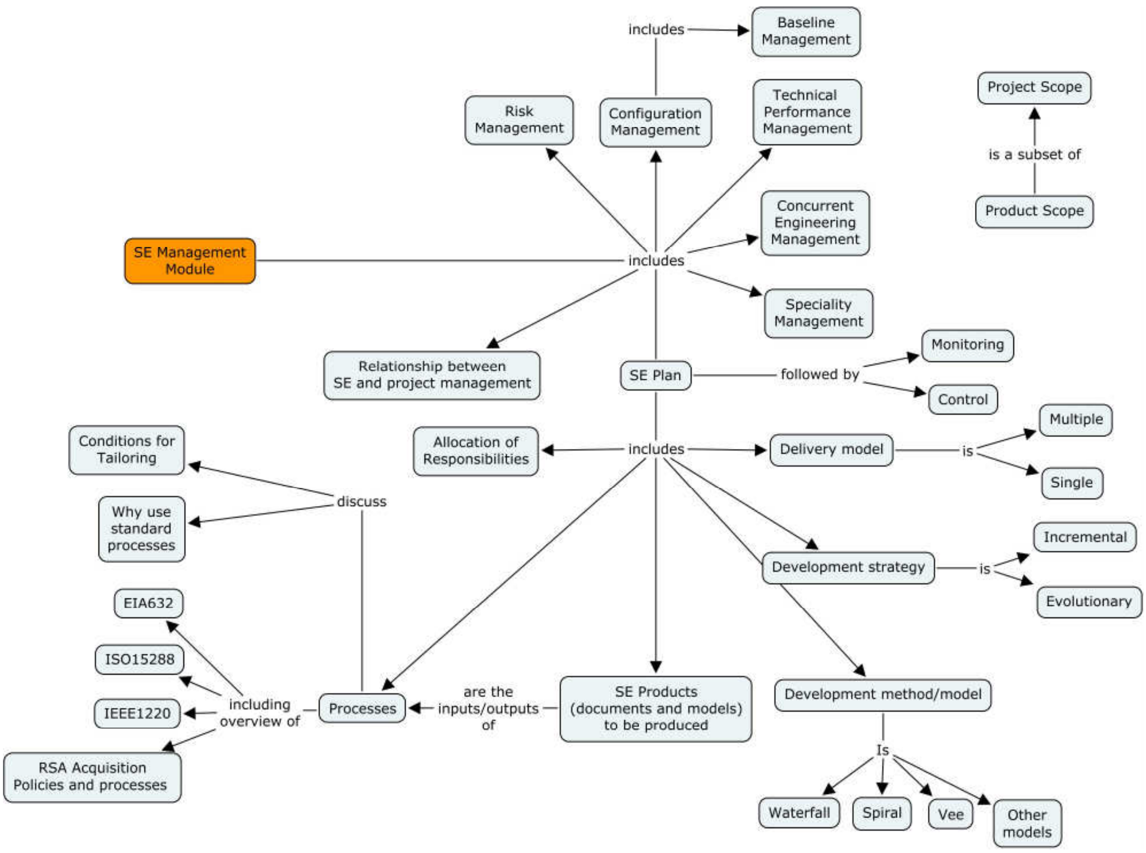

Figure 5: SE Management module[1]

Implementation, Integration, Verification and Validation

Implementation, Integration, Verification and Validation module deals with realising the solution, integrating and verifying various elements of the solution and checking that it meets the stakeholder needs. The outline for the module is presented in Figure 6. While the 'V' model is the traditional approach for explaining integration, verification and validation, it is a rather simplified model. More emphasis needs to be placed on a *plan* for implementation, verification and integration. Depending on the nature of the system, there may be a transition to operation before validation can be performed.

[1] Notes:
Separation of some SE management items from SE planning is for clarity and is artificial. Life-cycle is dealt with as behaviour.

Early validation is an important area to emphasise in this module. In the early phases of the system lifecycle, use of simulations can significantly reduce risk. This form of validation occurs before any physical implementation or integration. However, implementation and integration have been grouped with verification and validation because these are normally intimately related, a fact that is overlooked in many courses.

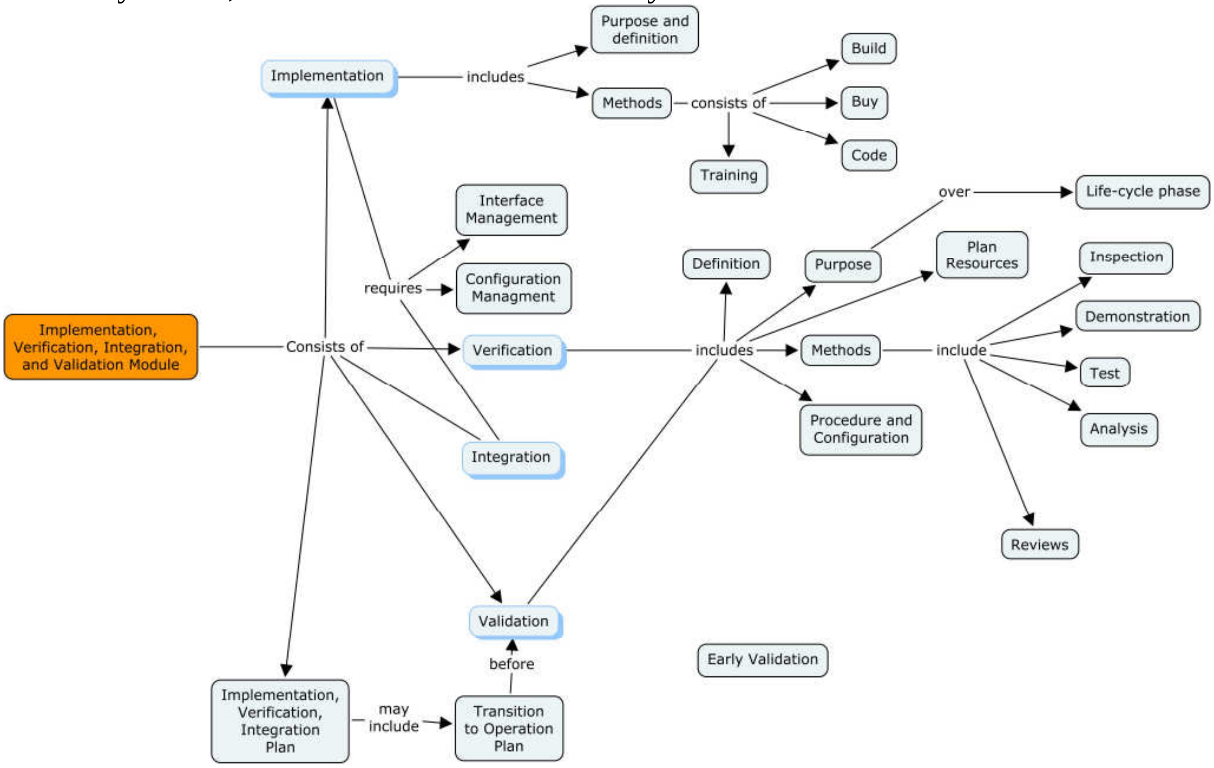

Figure 6: Implementation, Integration, Verification and Validation

Conclusion

The shortage of systems engineers in South Africa requires strategic action. This programme will contribute to developing the next generation of systems engineers. In order to accelerate the development of high competency systems engineers we will need to adopt new (old) approaches to education with current SE content. What is being proposed will require considerable effort, but is expected to yield good results. This paper does not address the broader national SE education in South Africa, but this issue is currently under consideration.

References

Bobbitt, J. *The Curriculum.* Boston: Houghton Mifflin, 1918.

Buede, D. M., *The Engineering Design of Systems: Models and Methods* Wiley Interscience, 2000.

Campbell, A and Cropley, D, Creating a Master's Degree in systems integration: Capturing and embedding industry knowledge. *Int. Journal of intelligent Defence Support Systems*, Vol. 2, No. 3, 2009.

Fournier, F., Pohl-Winkelmann, C. & Wilke, D. M. Using training platforms to develop collective skills and performance in *Systems Engineering Proceedings of the 7th biannual European Systems Engineering Conference (EuSEC)*, 2010

Gonçalves, D.P. *Developing Systems Engineers*, Portland International Centre for Management of Engineering and Technology (PICMET), July 2008, ISBN 1-890843-18-0

Gonçalves, D.P. and J. Britz, *Systems Engineering Research: Screening Candidate Systems*

Engineers. DPSS-SM-CPG-044 Rev 1, 2008.
Hooks, I. & Farry, K., *Customer Centered Products: Creating Successful Products Through Smart Requirements Management*, American Management Association, 2000.
Honour, E., "Understanding the Value of Systems Engineering", Proceedings of the Annual International Symposium of INCOSE, 2009.
INCOSE UK, 'Systems Engineering Competencies Framework', Technical report, Issue 2. 2006. Available from World Wide Web http://www.incose.org.uk/Downloads. Last accessed 06 March 2008.
Kasser, J.; Hitchins, D. & Huynh, T. "Reengineering Systems Engineering", Proceedings of the 3rd Asia-Pacific Conference on Systems Engineering (APCOSE), 2009.
Kasser, J. E., Integrated Multidisciplinary Engineering for the 21st Century: Study Guide, 2007.
Squires, A. & Cloutier, R., "Evolving the INCOSE reference curriculum for a graduate program in systems engineering", *Systems Engineering*, Vol. 12. 2009.
Halligan, R., *Systems Engineering: From awareness of need to retirement from use*, PPI-XB10-003 162-2, Project Performance International.
Massachusetts Institute of Technology: MITOpenCourseware, http://ocw.mit.edu/OcwWeb/Engineering-Systems-Division/index.htm, last accessed 2 December 2009.
University of South Australia, Master of Engineering (Military Systems Integration), http://www.unisanet.unisa.edu.au/programs/program.asp?Program=LMSI&Year=2009, last accessed 15 December 2009.

Appendix: Mapping of SE competencies to SE modules

Competencies addressed by the proposed modules are shown in Table 6. A number of competency areas remain to be addressed. Enterprise issues are not currently addressed. Other aspects that might need additional support are the integration of specialities and decision analysis.

Table 6 Mapping of SE Competencies to Modules

Category	Competency	Module
Systems Thinking	System Concepts	SE core
	Super System Capability Issues	SE core, Requirements analysis
	Enterprise & Technology Environment	Partly addressed in Architecture
Holistic Lifecycle View	Determining and Managing Stakeholder Requirements	Requirements analysis
	Systems Design – Architectural Design	Architecture
	Systems Design – Concept Generation	Architecture
	Systems Design – Design for…	Partly addressed in Architecture
	Systems Design – Functional Analysis	SE modelling concepts and notations
	Systems Design – Interface Management	Requirements analysis, Architecture
	Systems Design – Maintain Design Integrity	SE management
	Systems Design – Modelling & Simulation	Modelling & simulation
	Systems Design – Select Preferred Solution	Foundations laid in Requirements analysis module, Architecture
	System Design – System Robustness	Reliability, availability and maintainability module

Category	Competency	Module
	System Integration & Verification	Implementation Integration, verification and validation
	Validation	
	Transition To Operation	
Systems Engineering Management	Concurrent Engineering	SE management
	Enterprise Integration	
	Integration of Specialities	Speciality overview
	Lifecycle Process Definition	SE management
	Planning, Monitoring & Controlling	

Biography

Duarte Gonçalves holds a B. Eng. in Electronics and a M. Eng. in Computer Engineering and is currently employed by the CSIR. He has been involved in engineering surveillance systems for the South African DoD where he has extensive experience in electro-optical systems, ranging from modelling the environment and electro-optical observation systems, to signal and image processing. He holds a full patent in the area of imaging spectrometers. He has consulted to the Karoo Array Telescope project, the South African technology demonstrator for the Square Kilometre Array (SKA) as a systems engineer. Currently, he is responsible for the systems engineering on a renewable energy project and developing systems engineering skills at the CSIR.