ISSN: 2754-6705

Journal of Mathematical & Computer Applications

SCIENTIFIC Research and Community

Review Article

Open Access

A Game Changer Proposal for Teaching, Learning and Assessment in the Computing Sciences: In the Case of Ethiopian Universities

Estifanos Tilahun Mihret

National Taiwan University of Science and Technology, Taipei, Taiwan

ABSTRACT

The proposed radical changes arise from my experience as a Senior Lecturer, Researcher and Author in this area of Academic Endeavour (put simply, computer systems development) over the course of a university teaching career since 2012; 11 years, and the document which released by Roy Morien. I have 5 International Books and have published nearly 10 journal articles.

This experience includes leadership of students in industrial experience project development, giving the author a high level of confidence that the proposed radical changes can be successful.

***Corresponding author**

Estifanos Tilahun Mihret, National Taiwan University of Science and Technology, Taipei, Taiwan.

Received: June 23, 2023; **Accepted:** July 04, 2023, **Published:** July 25, 2023

Keywords: Teaching, Learning, Assessment, Computing Sciences, Ethiopian Universities

Introduction

Temphasize their role in mentoring and guiding students eaching academics will be styled 'Learning Leaders' to through their learning experience. Teaching Academics will have a major mentoring role, and they will have a significant responsibility to ensure students achieve at a high level by constant monitoring, mentoring, assisting and evaluating students' progress.

Teaching Academics will now be a teaching team rather than individually and alone being responsible for semester bounded subjects, often with protected 'ownership' of the subject. Students will consult Teaching Academics on the basis of the Teaching Academic's particular expertise and competence in an academic area or topic, rather than as the teacher in charge of a subject.

E-Learning & social media will become prominent, with traditional lectures and formal tutorials abandoned in favor of a substantially e-Teaching environment and the use of social media, with face-to-face learning between members of the Teaching Team and student project groups. Teaching Academics will be required to create e-Learning material, particularly video material, for students to access via the Internet on a Facebook group or a dedicated University eLearning website.

While maintaining the term/subject structure for any necessary University administrative purposes, end of terms exams could be abandoned in favor of continuous longitudinal assessment, including assessment by the Teaching Team, student self-assessment and peer assessment, where students do not 'fail' but revise, reflect and re-learn where appropriate under the guiding eye of the Teaching Team. The use of a 'super project', an industrial strength and complex development project activity, as the platform for developing student competence, skills, know-how, knowledge and deep understanding over the full period of the course. Requirements will be added and modified all during the course to ensure students experience all important aspects of software development as required in the curriculum.

Learning Principles and Practices

- Blended Learning: A combination of e-Learning, student learning groups, as needed lectures, face-toface learning with members of the Teaching Team.
- Student-Centered Learning: Students will be required to take significantly greater responsibility for their own learning. Overall, Student-Centered Learning with hands-on, practical learning will be the usual Learning approach.
- Project-Based Learning: Project-based Learning, with the 'super project', will be the primary learning vehicle, ensuring students have significant hands-on system development experience, with deep learning and gaining significant 'know-how' and 'know-why'.
- Team Learning & Soft Skills: Soft Skills, including Team Learning, with students being grouped into Learning Teams of 6 students, and undertaking all of their work in the team, with curriculum emphasis on Creative Thinking, Problem Solving, Design
- Thinking, Agile Learning etc..
- Longitudinal Learning and Assessment: 'Deep
- Learning' experiential learning rather than 'Shallow
- Learning' for the purpose of a forthcoming examination. Continuous evolution of knowledge and know-how. The 'super project' will provide a constant context and continuous learning opportunities.

- E-Learning & social media: E-Learning becomes a major aspect of the students' learning endeavors. SelfDirected Learning, self-motivation, and reflective learning practices will be the norm.
- Meeting Industry Needs: Students will gain many of the essential skills, hard and soft, that will enable them to confidently and immediately be useful to their future employer, while also having a broad education.

Curriculum Matters

- The "Super Project" foundation: The curriculum will be defined by the development requirements of the 'super project' which will be controlled and managed and extended and broadened as necessary by the Teaching Academics.
- Comprehensive curriculum: Overall, the 'super project' will reflect an appropriate and comprehensive curriculum more than sufficient to meet any academic quality guidelines.
- Effective and Efficient Curriculum Streaming: The curriculum will be streamlined and streamed in such a way as to ensure continuity of learning and eradication of the 'waste' of students' shallow learning only for the purpose of passing an exam, and deep learning by appropriate continuous repetition of development activities will be achieved.
- Creative Thinking and Problem-solving Methods will be an essential component of the curriculum which will emphasize 'soft' skills including Self-Directed Learning, Team participation, etc.
- Appropriate aspects of the curriculum will be selftaught with guidance and reflective analysis of previous experience in the 'super project'.

The overall curriculum content of any university or college course must always be scrutinized, on a continuing basis, for appropriateness and relevance. This is especially so for computer studies. It is difficult if not impossible to consider if the curriculum is 'complete' as containing all the knowledge about the academic area, but the curriculum must be sufficiently comprehensive to be able to rightly say that the students are qualified in that professional or academic area.

The curriculum overall will be dictated by the nature and content of the 'super project', and at any one time the curriculum of interest will be stated by the requirements of the 'super project' as given to the students at any point in time.

Overall, curriculum could be classified into two categories: Practical and Theoretical. Obviously, any aspect of curriculum will contain elements of both, but one or the other is prominent. For example, programming is an essentially hands-on activity and can only really be learned by doing it. There are certainly theories, practices and principles underpinning it, but unless these are put into practice, the students will probably miss the point and forget about it. Similarly, database system development. Systems Analysis could be classified as being predominantly theoretical, although the theory does form and inform the practice. Never forget, however, that there have been more than 200 Systems Analysis Methodologies published in the last 40 years, and nearly every textbook on database presents different versions of and approaches to Entity Modelling and Data Modelling. Each of these methodologies is based on theory and principle.

It is difficult to identify any part of an appropriate and comprehensive undergraduate program of IT, Business Computing and even Computer Science that is theory only or is practical only. What is a problem is where theoretical aspects are taught without practice, on the promise or assumption that the theory will be useful 'later on', 'soon', in a 'future subject'. Students may not grasp the full understanding of the theory, and/or most probably will just forget about it after the exam. So, the principle here is that theory should be taught in situ, meaning put into practice immediately to enable the students to understand its relevance and usefulness, or, better still, the theory is derived by retrospective analysis of and reflection on the experience and practice of the students in previous subjects.

Assessment and Evaluation

- Continuous assessment, peer assessment, selfassessment will be emphasized, and sit-down exams will be de-emphasized, to the point of just not having exams at all.
- Mentoring and guiding students: The primary purpose of assessment will be to provide guidance to the Learning Leaders (the Teaching Academics) for the mentoring and guidance of students, with the ultimate aim of achieving a significantly high level of learning achievement by students.
- High Learning Achievement: There should be no more satisfaction with students earning 'Ds' or 'Cs' which in reality is an acknowledgement that the student didn't really learn much. There should be an expectation that all students will achieve a Summative Assessment score in the 75%-99% range, rather than in the 45%-50% range.
- Teaching Academic Responsibility: Teaching

Academics will shoulder a greater burden of ensuring that their students do achieve these results, which will be quite possible in the Teaching and Learning processes suggested here. This is done now, but the Teaching Academic's options are restricted by the, usually unwritten, policy of not failing students.

Keeping in mind that University policies and timetables (i.e. semesters, exams, grades etc.) need to be adhered to in general, I see the reason why assessment of students can be done on a semester basis. The semester results would be essentially a longitudinal statement of learning achievement (Summative Assessment) and would also be useful for the continuing monitoring of students' further learning needs (Formative Assessment). The 'super project' is the overall continuing part of the course, and assessment needs to be done regularly and periodically as a matter of course.

I would consider the matter of assessment of students and evaluation of their projects and activities in the following way:

1. Having studied the literature on academic assessment and evaluation, I would include student selfassessment and peer assessment as an integral part of the overall assessment.
2. End of semester exams may still be required by University and Faculty policy, but we must acknowledge the negative aspects of this:

- Students tend to limit their learning to what they think will be in the exam.
- The 1-2 weeks prior to the exam week are usually seen as the real learning period in the semester, when students study hard, cram in the knowledge sufficient to pass the exam. Inevitably it means shallow learning, and much of what is 'crammed' is forgotten after the exam.
- It is almost policy in many Faculties that the Teaching Academics must give students 'exam tips' which does nothing more than narrow down the students' scope of shallow learning.
- The emphasis on the end of semester exam often causes

significant stress on students, often resulting in the student 'freezing' in the exam and failing because of the stress impact.
- The very purpose of end of semester exams should be scrutinized and the real academic benefits, if any, clearly stated.

3. Continuous assessment, or "semester papers" and small semester limited projects are open to cheating and plagiarism, students copying other students' work, students not participating in team-based projects, and so on.
4. Too often students fall behind in their studies, in a semester, and resort to cheating etc., as suggested in 3, and there is no systematic or systemic approach to monitoring students' real progress and knowledge gain.

Meeting Industry Expectations

- While acknowledging that the University's role is far more than being a training school, industry expectations must be understood and met by the production of competent, knowledgeable and skilled graduates, able to demonstrate a high level of knowhow, independence and creative thinking and LifeLong learning ability.
- Aspects of industry which are most important and can be achieved under this new radical approach include the 'soft' skills of:
- Teamwork: software developers normally work in teams. After 4 years of working in a team with perhaps 5 or 6 other students, the student will be well experienced in this.
- **Problem Solving:** Software developers are constantly faced with the need to solve problems and develop the programmed logic to solve those problems. The 4 years of 'hands-on' experience solving the problems in the 'super project' will certainly ensure that students have problem solving skills and abilities.
- **Thinking Creatively:** The T&L emphasis on creative thinking, independent action, ability to find solutions, ensures the graduate will, again, be very well able to bring these skills to their employment.
- **Ability to work independently:** 4 years of monitored, mentored but self-directed learning with significant hands-on experience and knowhow ensures that the graduate has this attribute.
- **Professionalism:** Knowing their status as educated professionals behaving in an independent, well-informed and ethical manner.
- Ability to continue Learning (Life-Long Learning, essential in professional practitioners)
- General know-how, hands-on skills, and analysis ability.
- Transitional skills: coping with Change.

Hard Skills

The graduate from the Department should have a number of essential hard skills, including:

- Programming Skills: Well-developed skills in one, perhaps two languages, and importantly, skills in debugging and code modification as well as the ability to learn other languages based on their broad knowledge of computer programming structures and concepts.
- Database Skills: In depth understanding and skill in creating databases, using SQL, using database development and development support tools. Modern
- DBMS's now termed NoSQL DB's will be included.
- Systems as a Service: Good knowledge of concepts and practices of PaaS, SaaS, IaaS, DevOps and Cloud Computing, including hands-on skills and know-how in using appropriate software.
- Ability to develop in most platforms; Windows, and smartphone and tablet computing.

What has Changed in 50 Years?

In the Computer industry: everything!

In the way universities and colleges teach, students learn very little! Obviously, what we may term 'educational technologies' have been developed and applied, and the Internet has provided substantially greater opportunities for plagiarism, but as far as the pedagogy of computer systems development is concerned, basically nothing has changed in 50 years. College and university Teaching Academics still usually give a weekly stand-up lecture, still give students a term paper and an end-ofterm exam, students still undertake, where appropriate, a small practical project. The tutorial may now have been replaced by a practical lab session. But basically, nothing has changed!

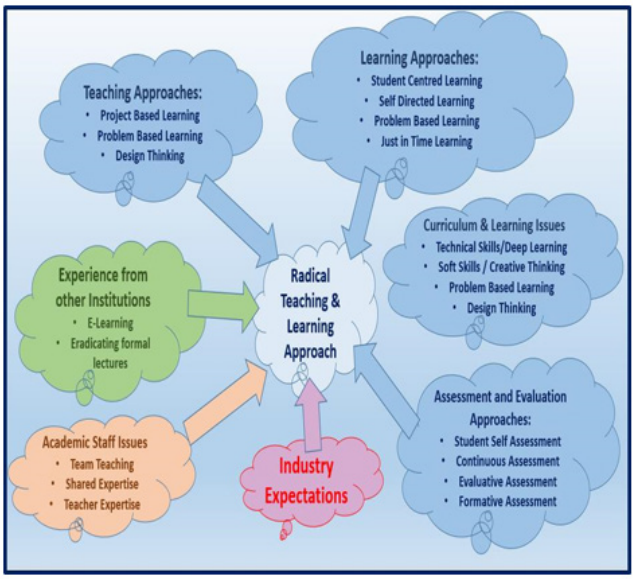

Figure 1: Other research, experience, and opinion

This figure summarizes my proposal, and shows the extent of research, experience and opinion that is available on the Internet. A significant aspect of the recent publications has been the need for an understanding of appropriate reference disciplines and drawing ideas and issues from outside the narrower scope of 'computer system development methodologies.

Summary of the Proposed "Game Changer" Approach

1. Teaching approaches include Team Teaching and ETeaching to replace the traditional single Teaching Academic per subject, giving stand-up lectures and tutorials.
2. Learning Blended Learning Approach which includes Student Centered Learning, Self-Directed Learning, Problem Based Learning, Student Responsibility for learning, Deep Learning, E-Learning in place of lecture and tutorial attendance.
3. Assessment and Evaluation Approaches of Student Self-Assessment, both Evaluative and Formative Assessment, Continuous Assessment.
4. Curriculum Matters, including teaching Creative Thinking, 'soft' skills such as Team Participation, Leadership skills and Problem Solving, as well as 'hard' technical skills, attention to employability.
5. Consider experience from other institutions in elearning.
6. Understand and meet industry expectations and the employability of our graduates. Promote our graduates by direct contact with employers, PR programs, skills 'fairs'.

7. Building a 'brand' for the Department, Faculty and University.

Subject Consolidation and the "Super Project"

- Individual subjects will be combined and consolidated but will remain 'on the books' essentially only as administrative units in accordance with University policy and requirements.
- The mainstream of Teaching and Learning over the duration of the course will be based on a 'super project' which will be developed over the whole course of the students' studies. This project will be of a very substantial nature, of 'industrial level' complexity, and will incorporate as many aspects of the contemporary Information Systems technical landscape as possible, from PC based spread sheeting through Cloud based database systems, DevOps, to Smartphone applications and 'remote' and 'mobile' computing concepts and practices.
- Teaching Academics will cooperate overall as a Teaching Team, each Teaching Academic having particular expertise that students can access as and when required. Students will therefore have available to them a Teaching Academic who is expert in programming, if needed, or another Teaching Academic who is expert in database, if needed, or one expert in iPhone App development or Android App development, or Internet programming and web design, and so on.
- Students will develop this 'super project' in teams over the whole period of their course.
- Stand-up lectures will be phased out in favor of properly developed videos, perhaps developed in collaboration with students from the Mass Communication, TV department of another Faculty.

What is the Real Value of the "Super Project"

Teaching academics have usually tried to include a 'hands on' project in the subject they are teaching. This acknowledges the importance of such practical activity. However, there are a number of weaknesses in this approach, which bring such projects almost to the point of irrelevance and failure.

- A subject-based project sufficient for one term does not give students any realistic experience in systems development which could be carried forward into their future employment.
- Subject-based projects must necessarily be small and simple, narrowly defined, which again do not give students any realistic experience in systems development.
- Because the project is assessable, all requirements need to be given to students at the start of term and cannot be changed, thereby not reflecting in any way the reality of project development.
- Students working in teams allows some students to hide behind the good work of other students.
- Students often copy from others or ask outsiders to do the development.
- Having a 'super project' as suggested here means that the students will be working in a context that evolves and in which the students' knowledge can grow with the project 'so far' providing a solid foundation for further learning. The main benefit of the 'super project' is that it gives the students an ongoing context in which to learn, in which they can see real outcomes. My experience has been that the students become engrossed in the project and are proud of their demonstrable achievement. Students usually have a significant level of creativity which is not used when students are sitting in a 2-hour lecture, and not otherwise drawn upon. A summary of the benefits of the 'super project' include:
- Students have experience over a period of time of working in a team, being part of a social group working together, solving problems, sharing knowledge and enjoying the companionship.
- The continuing project becomes an understandable and understood context for the students, providing a realistic environment for students to work in, resulting in a demonstrable outcome that the students will be proud to show.
- Students better understand the requirements for standards and appropriate documentation.
- Students will gain real-world experience of the problems, difficulties, successes and disappointments when developing a project of an industrial scale.
- Students will better understand the need for collaboration with other 'developers' and 'clients' and 'users'. The social aspects of team development are experienced, not just the technical.
- From a purely Teaching and Learning perspective, the students will learn 'deeply' and gain realistic knowledge and knowhow.
- A well-developed outcome of which the students will be proud will drive 'deep learning', in total contrast the anxiety driven shallow learning of the exam environment.
- As the project spans maybe 3 years of their course, the concept of longitudinal learning is experienced. The students' knowledge and know-how are built up over time, layer after layer, in a longitudinal way, akin to the intellectual development experienced in a research project.

Consistency with University and School Policy

While the University does have administrative policies and requirements, it is not the practice at universities to dictate curriculum nor to impose teaching and learning practices on lecturers and students. What is taught and how it is taught are usually left up to the Faculty or Department, and the individual lecturers. Similarly, while assessment is a given requirement, the style, nature and mode of assessment is not usually prescribed.

The current structures of the courses are obviously in accordance with University and Faculty policy, and Department course structure requirements. There is no reason why the proposed radical approach to Teaching, Learning and assessment must necessarily be counter to current policies. However, subjects will essentially become administrative units only, to conform to University policy and practice.

For example, students would still enroll in the Fundamental of Database Systems subject in a semester, as always. However, perhaps that would be in Semester 1 rather than Semester 3 (Year II), because the 'super project' approach introduces students to data structures, database development tools, database system analysis activities, ER Modelling and Data Modelling right at the start (and their knowledge grows, evolves and becomes deeper over time). Students would enroll in the Systems Analysis and Design subject in the normal way and be assessed in that subject based on the knowledge that they have acquired by doing the 'super project' so far.

However, the Systems Analysis and Design subject would be moved to the final semester for the purpose of enrolment. The subject would then become much more of a reflective subject with students, who would now be much more mature in the learning and intellectual attitudes, bringing their experience gained in the 'super project' to developing theories, discussing systems analysis best practices, writing a substantial User Guide for systems analysts as post-hoc reflective learning. Perhaps we can call this 'deductive reasoning' where the theories about Systems Analysis methodologies are understood by reaching

specific, logical conclusions based on previous experience and the examination of that experience.

Similarly, by the final term of their course, the students will have had sufficient experience to re-visit their team activities over the course of the 'super project' and reflect on the success of the project, the efficiency and effectiveness of their development activities, and be able to present a substantial, thoughtful, mini dissertation on Software Project Management. This would be done within a subject framework addressing Software Project Management Best Practices, comparative Software Project Management approaches and Leadership studies and Team Studies.

Experience and practice will enable students to better formulate, learn about and understand appropriate Theory, while reflecting upon and developing their own theoretical propositions and formulations. This is a significantly better approach than having stand-up lectures before the practical learning activity while promising the students that the theories presented to them will be better understandable 'later' and applicable 'sometime in the future'.

One important consideration is that regardless of which particular systems analysis or project management method is taught and examined, almost inevitably the graduate will commence with an employer who 'does it their way', often with in-house approaches. This implies that specific methods being taught is not very helpful, but better a good understanding of the need for a methodical approach is essential.

No Barrier to Implementation

How the 'super project' approach will be acceptable and accepted by 'the powers that be' according to University policies and practices is an obvious potential barrier to acceptance. However, as discussed above, it need not be fatal, and it need not even be seen as a problem; certainly not as a problem that is insurmountable.

There are two major aspects to this proposal. The first is the essential amalgamation of the curriculum, especially programming, systems analysis, database systems and project management, into an holistic curriculum approach based on the second major aspect, which is the use of a 'super project' of significant industrial proportions that will be the major learning vehicle, covering most of the course, and undertaken by the students over the full duration of their course.

As discussed above, students will still enroll in subjects in terms, essentially the same curriculum will be covered overall, but the curriculum will essentially be decided by the requirements of the 'super project', as given to the students at any point in time. Students will still be assessed, and their grades recorded by subject and by term. What will be different will probably be the order and sequence in which the curriculum will be learned, the methods of Teaching and Learning, and the Assessment methods. The curriculum will be modified to ensure that Creative Thinking, Problem Solving and other 'soft' and 'transitional' skills will be emphasized.

So, from a policy point of view there should be no impediment to implementation of the radical approach to Teaching, learning and Assessment, as proposed.

What follows in this document is written in a way that acknowledges the policy constraints and demands.

Consolidation of Subjects

Many subjects can easily be consolidated. For example, systems analysis activities can be undertaken as part of the development of a database system in which database concepts, Entity Modelling, Normalization etc. are taught, and practiced. Include the development of database processing into the mix and it is easy to see how the programming and construction of a database system, which includes data and processing ascertained from users and clients, are really just different but closely integrated parts of the whole.

By consolidating subjects, more room becomes available in the formal structure of the course to include other subjects, which may or may not be able to be taught in the 'super project' structure. 'Introduction to Computer Forensics', 'Ethical System Hacking' subjects come to mind.

Sequence of Subjects

Would it be appropriate, even better, to teach students about data structures, under the heading of database development, as a first subject, with a programming subject following that? Do students need a Systems Analysis subject only after they have started to learn how to program, but before anything about Project Management is taught? Is it really useful to spend a semester teaching students the intricacies of a particular Database System Analysis Method (such as Falkenberg's Deep Structured Sentences or the ANSI/SPARC three-level framework for database management systems, or traditional SDLC and Data Flow Diagrams) entirely as a theoretical and paper-based subject before a subject in which the students actually implement a database system?

No! Often the sequence of subjects and streams of curriculum are just matters of convenience, and not dictated by University or Faculty policy, nor are they essential in their place in the curriculum overall.

So, there is no reason whatever why students cannot start to experience System Analysis on Day 1 of the course, while being introduced to data structures at the same time, and starting to learn about the GUI development environment and language that they will very soon be using. Their knowledge in these areas will be deepened and expanded over the future period of the project.

Curriculum Content: Relevance and Comprehensiveness

Obviously, all curriculums should be relevant, but this must be considered in terms of the future. Will the curriculum now being taught be relevant in the future when the student graduates? The other real problem faced by curriculum developers is that there are time limitations, so what to choose for the curriculum must be taken from a large body of subjects of interest and cannot include all of it.

A significant aspect of curriculum must be the 'soft' skills that students need to learn, such as Critical Thinking, Leadership Skills, some management theory. Basically, the reference disciplines that actually support the concept of building systems in a systematic, productive way to support the activities of the business, or to apply technology to problems. It is addressed this situation in two papers and [1,2].

What must not be disregarded is simply teaching students how to learn!

Just-In-Time Learning

Up until now, Just-in-Time learning has meant cramming 'just in

time' for the end of term exams. This approach has encouraged, indeed almost guaranteed, shallow learning for the purpose of passing the exams. Just-in-Time learning is given a very different meaning here, though.

There is a significant problem that has been experienced by any teaching academic, and that is ensuring students learn particular subject matter by the time that knowledge needs to be applied in other subjects. Significant time has been wasted by students forgetting what they learned previously, by subject matter not being followed up and applied in subsequent subjects, and so on. Frankly, efforts at streaming and having 'pre-requisite subjects' have not been very successful for a variety of reasons (especially the problem of 'shallow learning', cramming sufficient for the end-of-term exam and then forgetting, by students).

By having the 'super project' as the basis of the learning agenda, requirements can be introduced at any time that will challenge the student to complete. The student must learn sufficient subject matter to be able to develop the requirements.

Careful consideration by the teaching staff of 'what to teach next', now more 'what the students should learn next', will guide the immediately next facets of the 'super' project, and will result in students learning the intended lessons 'just in time' to apply that knowledge. One significant and advantageous outcome of this is, given the 'super project' is thoughtfully managed, and by the end of the course students will not have learned matters that are not really important or useful.

For example, the Teaching Team (comprised of all 'Learning Leaders' in the Department), might decide it is time for students to learn how to develop smartphone apps. So, the students are advised that the 'super project' system now requires a smartphone app for managing student accommodation in and near the university, and a second app for allowing students to see where the shuttle buses are on campus, and a third app for allowing students to access their academic record and assessment results on their smartphone. Because these apps are relevant to the students, they are within the context of the project the students are working on and will therefore be seen as useful to the students, instead of being a simple 'made up' project.

Lean Education

Following the basic precepts laid down by researchers and practitioners in the area of Lean Manufacturing, and applying what are known as the 7 Wastes of Manufacturing, the concept of Lean Product Development has emerged as a well published and researched topic, as has the more relevant concept (in this context) of Lean Software Development. These concepts can readily be applied to education, and the 7 Wastes of Education have been identified as central issues in what is now known as Lean Education. The 'waste' of unused or forgotten subject matter is one important aspect addressed under the heading of Lean Education, a concept that which published in 2014 and 2015 at international conferences [1] [2] [23].

Post-Practice Reflection

Consider an example, that of Systems Analysis Methodologies. Since the 1970's there have been many different systems analysis methods published. These have been categorized in many and various ways, such as being 'soft systems analysis methods', 'Waterfall model methods', 'Datadriven', 'Object Oriented, 'Agile Methods'. To have a separate and bounded subject called Systems Analysis Methods which is limited to one term means that very little, if anything, can be taught about most of these approaches. Usually a single approach is taught, selected usually according to the particular lecturer's view and knowledge. Such as subject is also usually taught towards the beginning of the course, on the assumption that that particular method will be applied in subsequent subjects.

Would it not be better to give students experience of actual hands-on analysis and development, perhaps allowing them to develop their own analysis approach, for better or for worse, and then allow the students to reflect upon their experience? This would give them a reflective framework upon which to assess various systems analysis methods, and to understand better why some methods are more appropriate, more successful than others. Even perhaps to decide that having defined methods that must be learned and followed, accompanied by process auditing, is really not necessary.

Such post-practical reflection, which would be done in the students' final year, can be applied to many areas. By first having mentored, assisted, hands-on practical experience in a realistic setting, followed by reflection on accomplishments and failures, allowing thoughtful analysis of the previous behavior and practice, will surely provide greater insights and critical ability and intellectual development, which we call 'education' as distinct from 'training.

Suggestions and Recommendations

Overall assessment and grading of students should comprise four heads of assessment:

1. Continuous assessment by the Teaching Academic over the course of the semester, with students informed of this assessment outcome weekly – 40% of total assessment.
2. Student self-evaluation by way of peer evaluation within the student project teams – 40% of total evaluation.
3. Six small spot tests on particular topics, of 30 minutes duration, at various times during the semester, totaling 20% of the total assessment mainly for Formative Assessment purposes.
4. Final end of semester exam – It is suggested that these will be discontinued.

The spot tests (3.) and the end of semester exam (4.) are conventional, are within University policies and practices, and need little explanation, except, for the spot tests, these are an on-going encouragement to students to keep up with their learning, and also to assist Teaching Academics to monitor students individually and overall for the purpose of providing help and extra information as required. This qualifies as Formative Assessment as well as, together with the final exam, Summative Assessment.

In the 'super project' approach, Continuous Assessment can be made realistic, useful, and free from corrupt practice (which cheating, and plagiarism is!) in a number of ways.

With Students Working in Teams

1. The team must prepare and update on a weekly basis, a Team Project Diary, recording the decisions made by the team in regard to standards agreed upon, documentation requirements agreed upon, design decisions agreed upon, commentary on overall progress etc. etc.
2. The team must prepare and update on a weekly basis, a Team Project Backlog, showing the known tasks, in order of development priority, that the team must undertake (i.e. basically the Requirements List), or has completed, showing the estimated time to complete, and in the case of finished tasks, the actual time to complete, and indicate which team

members or members were given the task.

3. For each individual member of the team:

a) An Individual Sprint Plan for the next week (or two weeks), showing the task(s) taken from the Team Project Backlog by the team member for them to develop or undertake. Each task should be directly linked to a task on the Team Project Backlog. This plan will include a comment by the students indicating completion or achievement.

b) An Individual Diary by the student with a commentary of the student's views and opinions, comments on their learning, self-assessment of their progress etc., and notes on their view of the contribution of the other members of the team, negative and positive.

At the end of each period (week or two weeks) the Teaching Academic is able to check that what the student has planned to do (Individual Sprint Plan) concurs with the task allocation of the team from the Team Project Backlog, check if the Individual Sprint Plan indicates completion of the planned tasks, or effort expended on the planned tasks, and then see the evidence of that effort which may be a document, or specific set of code or coded functionality created by the student. The code could then be examined, and the student required to explain the code. This is the highly desirable Plan -> Do -> Validate -> Feedback cycle stated by such as Deming in what is known as the Deming Cycle.

The Teaching Academic can also view the student's Individual Diary to see what videos or documents have been viewed by the student as part of their learning activity.

These documents provide a longitudinal narrative of the students' progress and activities which can be used for Teaching Academic continuous assessment, Teaching academic monitoring of student participation and other aspects of the student's behavior to enable the Teaching Academic to provide a reasonable and valid assessment.

Student self-assessment and peer assessment can be evaluated by the Teaching Academic as to its validity when compared to the Teaching Academic's view. Students often have the tendency to give all members of the team a high mark, regardless of the individual student's actual contribution. Also, peer assessment imposes peer pressure on the student to keep up, behave appropriately and contribute.

I have attempted many different ways to examine, assess and evaluate students, including take-home exams, practical exams where code or database schema is developed in the exam room, provision at the start of semester of sample questions to guide students' study and learning etc.

One approach which I believe has been reasonably successful is to evaluate the team as a whole, and provide a gross percentage for the team members to distribute among themselves (e.g. 6 team members, my view was the team overall was worth 70%, so 420% was given to be allocated by the students according to the team's view of each members achievement).

Work Load Imposed on the Teaching Academic

There is no doubt that these assessment suggestions impose a greater workload on Teaching Academics than just having a semester essay and a final exam to mark. However, the assessment activity brings the Teaching Academic and students more closely together, resulting in a fairer and more correct assessment. Weekly, or, better, two-weekly meetings of each team with a Learning Leader (no longer termed Teaching Academics) do take up time!

However, after an initial busy period preparing e-learning material such as videos, staff will have their workload reduced by probably 4 hours a week by not being required to have 2 hour lectures each week, and a 2-hour practical class each week.

My experience is that I, as a teaching academic, have always enjoyed the interaction with students in a more informal way than by standing up the front of the lecture room talking. My experience also is that students learn better, are more willing to learn and act, when the Teaching Academic shows an obviously interested and collaborative attitude, and is readily available to mentor, encourage and help the students.

Teaching Approach – The Learning Pyramid

The Learning Pyramid, often referred to in educational publications, is shown here. There has been a lot of criticism of this concept, particularly of the percentages applied to each teaching style. Nonetheless, we can take from the Pyramid the fact that students retain little of what they are told in a lecture and retain a greater proportion of the information they are given as the teaching method becomes more participatory. What Teaching Academic has not, themselves, understood the reality that when we teach, we inevitably learn deeply about the subject matter we are teaching.

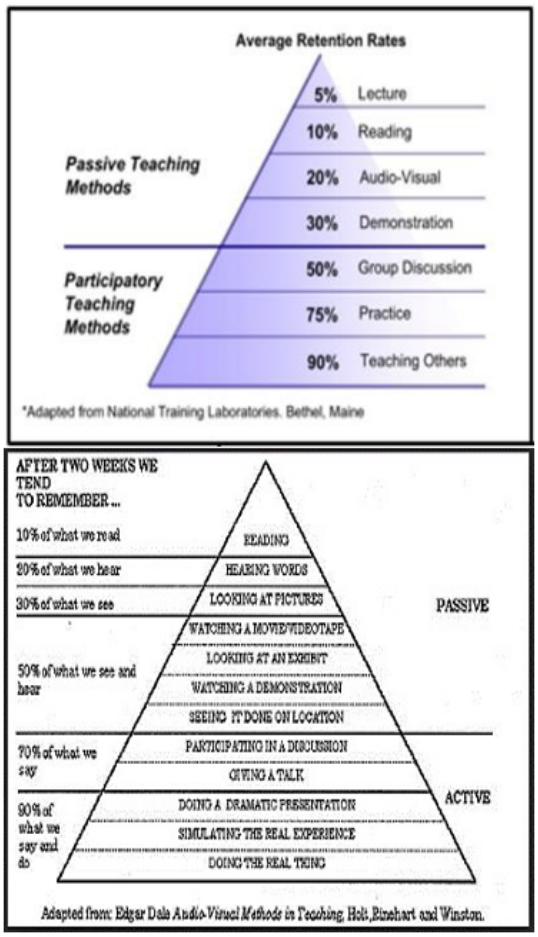

Figure 2: The Learning Pyramid

Another version of the Learning Pyramid is more relevant to the concept of having a 'super project' as the primary Teaching and Learning Method. This indicates longer-term retention rates. If we consider the concept of deep learning as being the ultimate

retention rate, i.e., learning over the longer term by practical hands-on learning, with continuous reinforcement and enhancement, we can see that the bottom of the pyramid is the most relevant. 'Doing the Real Thing' i.e., a long-term projectbased learning activity that is designed to be like 'the real thing'. 'Simulating the Real Experience' i.e., continuous development over the long term of an industrial strength project tailored as needed to lead the students into a different but associated learning experience which also reinforces the prior learning, resulting in deep learning, which is actually knowledge and know-how.

Learning Styles

According to Wikipedia, there are many different theories on Learning Styles. One prominent model divides students into four styles of learning: Visual, where the student retains information visually presented, Auditory, where the students retains more of what they hear, Read/Write Learning, where the student is a good note taker, and learns most from reading texts, and tactile/kinesthetic learning where the student prefers to learn via experience—moving, touching, and doing (active exploration of the world, science projects, experiments, etc.).

This is only one model of learning styles, but what it does tell us is that a variety of different Teaching Approaches are necessary to provide a good learning environment for students. This is properly termed a Blended Learning approach.

Blended Learning

A combination of learning approaches, including e-learning, student learning groups, face-to-face discussion with members of the Teaching Team, small group learning based on student project groups, lectures only as needed. Discontinuing the practice of regular stand-up lectures is part of this proposal. This has been done elsewhere, so it is not exactly leading edge (or 'bleeding edge' as some would have it). For example, the Business School at University of Technology Sydney has a new building that has no lecture rooms, because formal lectures in the traditional style are no longer presented . Similarly, Charles Sturt University in South Australia has introduced an entirely new Teaching and Learning Model in the Faculty of Engineering that does not include stand-up lectures, but is more project-based, hands-on experience. To put it bluntly, stand up lectures, variously classified and indeed disparaged as 'chalk and talk, or 'the sage on the stage' are usually boring for the students, and their effectiveness is usually measurable by the amount of inattention displayed by the students.

E-Learning

In both institutions just mentioned, and in other universities such as Curtin University of Technology in Perth , greater emphasis is placed on e-Learning by having an extensive library of videos of lectures available to students, on the Internet.

Social Media Use

I personally have little experience with using social media, such as Facebook and Telegram support Teaching and Learning. However, I am certainly aware that there are great possibilities that could be achieved using modern social media. Part of the new strategy for Teaching and Learning must include these.

Student Responsbilities

The role of the university teaching academic is two-fold. First, to advise students on what they must learn to become accomplished professionals in their field of study, and second, to guide the students in their study and help them achieve their learning outcomes.

The role of the Teaching Academic in a university is not to tell students the specific, bounded information that will be tested in a final exam, requiring only that the students memorize that specific bounded knowledge.

Students must be handed greater responsibility for their own learning and knowledge acquisition. As, usually, future professionals in their field, notions of Life-Long Learning, Professional Development, expanding their knowledge through self-motivated personal learning, are demanded. This then is the first 'learning matter'. But students coming from secondary schools cannot be immediately expected to understand this and understand how to do this. Creative Thinking, conceptual thinking, personal responsibility etc. need to be taught and can be taught.

Team Based Learning

Inevitably, graduates entering employment in the system development industry will be placed in a team, possibly with a 'mentor'. They will work in teams for the next few years, if not for their entire careers. Therefore, a graduate who has experience in working in teams, as an integral part of the university learning activity, will be in demand by potential employers.

At the start of the 'super' project, students will be allocated into teams, essentially randomly chosen at the start. The team comprising 6 or 8 students will be the primary learning group. The team members will have the joint and several responsibilities to work in the team according to the team's standards of behavior, and produce outcomes which will be in accordance with team standards.

Providing Evidence of Activities and Outcomes

As discussed elsewhere, certain documentation must be kept by the individual student, and the team as a whole, that provides evidence of each student's personal endeavor and learning, and participation in the team activities. This documentation includes weekly (or fortnightly) Personal Learning Plans, Personal Activity Logs and subsequently demonstrable outcomes, such as program code, form designs, project documentation etc. A strict chain of evidence of achievement is established linking specific planned activities (in the Personal Learning Plan) with the actual achievement of the plan (stated in the Personal Activity Log) and further linking to the demonstrable outcome. This documentation will be assessed for its excellence and veracity, and as evidence of endeavor and outcome by the student.

Learning "Best Practices"

There is an overwhelming volume of literature available on the Internet and in a hundred internationally learned journals on what can be termed 'Learning Best Practices'. All of ProjectBased Learning, Task-Based Learning, Problem-Based Learning, Self-Directed Learning, and Student-centered Learning etc. The overwhelming view to be found in this research is that the active learning by way of hands-on projects is superior to the passive learning of stand-up lectures and tutorials. I would add to this the almost useless practice of having semester limited, simplistic projects, particular to the subject.

The Social Aspect of Learning

What I mean by this is fundamentally Do the students enjoy their learning activities? Are lectures basically boring? Would a hands-on practical activity with fast turnaround feedback and immediate evidence of achievement make learning more enjoyable? Would enjoying the learning activities together with your teammates and friends enhance the learning experience, and thus enhance the learning per se?

As published in two papers at conferences and in international journals, clearly demonstrated both that the social aspects of the learning environment and immediate feedback and evidence certainly were important and were significantly enhanced in a Project-Based Learning activity.

Almost all of the 150 plus students involved in the industrial experience projects demonstrated significantly greater enthusiasm and interest, and obviously enjoyed the experience. There was always a level of excitement when the students could demonstrate the software that they had developed.

Industry Expectations

While it must be acknowledged that a college or university degree course must be more than a training course in specific software products, the fact is that students are always looking forward to employment. This demands that industry expectations must be considered. This has been a matter of concern expressed by a number of educational institutions and employers alike.

Institutional Experience

In two computer science professors at the University of California, Berkeley, interviewed enterprise developers to find out what they wanted from new graduates. The consensus from industry was that fresh college graduates were not equipped to deal with corporate development scenarios. In effect, the enterprises were asking for students that understood how to dive into an existing application's codebase, then modify, fix or otherwise work with it. That is, hands-on, real-world and realistic experience! "When employers do hire from college, the evidence suggests that academic skills are not their primary concern," says Peter Cappelli, a Wharton professor and the author of a new paper on job skills. "Work experience is the crucial attribute that employers want even for students who have yet to work fulltime." This quote comes from which, admittedly, is talking more about graduates gaining experience in internships. However, the point is made; educational institutions are more and more acknowledging the necessity of hands-on, 'real world' project-based experience in new graduates entering their first employment.

Department Facilities Availability and Access

Equipment Facilities

With the 'super project' being the main learning vehicle, making computer facilities available to students almost on a 24/7 basis would be extremely advantageous in supporting student learning. If the idea of reducing or eradicating stand-up lectures is adopted, large 'lecture room' spaces would become available for computer facilities to be installed.

Software Facilities

There are many useful and easy to use software packages available on the Internet. Two of my favorites for the purpose of supporting my teaching and being useful to students are TeamViewer®, a remote access and control product, and any number of virtual printer packages. These are free downloads. The Department should compile a sophisticated set of software packages to support all aspects of the 'super project' and ensure that students are fully trained and skillful at using them. This is now an extremely important aspect of any CS or IT student's skillset, especially given the plethora of tools that can be categorized under the heading of DevOPS. As well, proper licenses should be obtained, to demonstrate the Department concern for correct and ethical behavior.

A Word on Using Teamviewer

One problem I always had in another Faculty was to do with assessing students' database projects. When the students brought their development efforts to my office, they usually demonstrated their project on their laptop. They had no understanding of the problem of developing on one system and running in production on another system. They also usually had a Database Connection Object in each screen processing program, which meant that they had maybe 10 DCO's. If the students had been required to install their system on my computer (thereby demonstrating an appropriate skill), it would have been necessary to update the connection string on every Database Connection Object in their system, demonstrating another skill and understanding. If they had only one DCO, then only one edit would have been needed, not 10. I couldn't advise them on how to do that because I didn't know the language or the IDE that they were using. In the Teaching Team situation, as envisaged in my proposal, I could have called in an expert to teach the students how to define a single Database Connection Object and make it visible to all programs in the system, and to read the new connection string from a small text file set up for the purpose.

Alternatively, if the students had downloaded and used TeamViewer, I could have then used TeamViewer to connect to their laptop in their room and control their computer, viewing their effort in situ.

Building a Research Profile

As discussed elsewhere, teaching academics will probably have more time to do their own research. As well, the whole 'super project' concept as discussed opens up a vast opportunity for all academics to publish at least one paper per year in a significant education journal or IS education journal. By definition, this adds to the research profile of the University. Research into Computer Education is just as welcome as research into image processing and Internet security issues and taxonomies. Improving computer education should certainly be high in the mind of academics, Deans and Academic

Administrators

Attracting Graduate Research Students and Other Researchers

By building the brand, applying to this Department, this Faculty, will be an attractive option for Graduate students particularly, both Masters and PhDs. The corollary is that by having more researchers in the Department, the brand is enhanced.

A Leading Name in CS and it Education

This is what could be achieved for the Department of Computer Science and Information Technology; gain a high reputation and become a leading name in CS and IT education, especially in Thailand, but still 'Going World Class'. In a world where there is an on-going demand for well skilled and welleducated professionals, but where there are also a huge number of education providers, maybe 'Building the Brand' might even become a matter of survival. It can certainly be a matter of pride and of known and acknowledged excellence.

Acknowledgment

I would like to thanks more Roy Morien who wrote my base document. And, I wish to dedicate this paper for my beloved country, Ethiopia.

References

1. Morien, Roy (1992) Prototyping Large On-Line Systems: Using a concept of a Focal Entity for task identification, Proceedings of the Third Australian Conference on Information Systems, Wollongong 5-8.

2. Morien Roy (2005) Agile Development of the Database: A Focal Entity Prototyping Approach, Agile Conference, Denver, Colorado, USA.
3. Morien Roy, Olive Schmidenberg (1994) Educating IS Professionals: The Tertiary Education Challenge, APITITE'94 Conference, Brisbane.
4. Chang Elizabeth, Roy Morien, Chin Kum Leng, C Cheah (2004) Systematic Approach to Triple Feedback Systems for Teaching Enhancement, Teaching & Learning Forum, 2004, Murdoch University, Perth, Australia.
5. Morien, Roy, E Chang, KL Chin (2004) Seeking Systematic Feedback for Enhancing Curriculum Quality and Relevance, International Conference on Software, Engineering Research and Practice (SERP'04: June 21 24, 2004, Las Vegas, Nevada, USA.
6. Morien, Roy (2005) A Critical Evaluation of Database Textbooks, Curriculum and Educational Outcomes, Information Systems Education Conference (ISEC2005), Columbus, Ohio, USA.
7. Information Systems Education Journal http://isedj.org/4/44/.
8. Morien, Roy (2004) Insights into Using Agile Development Methods in Student Final Year Projects, Informing Science + Information Technology Education Joint Conference (INSITE2004), Central Queensland University, Rockhampton.
9. Journal of Issues in Informing Science and Information Technology Vol.1, 2004.
10. Morien Roy (2005) Student Experience of Using Agile Development Methods in Industrial Experience Projects, Information Systems Education Conference (ISEC2005), Columbus, Ohio.
11. Roy I. Morien (2006) Student Experience of Using Agile Development Methods in Industrial Experience Projects. ISEDJ 4: 1-12.
12. Morien Roy (2010) An Agile Project Management Manifesto – A Reference Disciplines Framework for Agile Development, iNCEB2010, Kasetsart University, Bangkok, Thailand.
13. Morien Roy (2013) Seeking a New Paradigm for Software Project Management: Reference Disciplines for Agile Development, 5th International Conference on Internet (ICONI 2013).
14. Morien, Roy (2010) Seeking a new Paradigm for Software Project Management – is it Agile, Lean or a Model of Concurrent Perception, iNCEB2010, Kasetsart University, Bangkok, Thailand.
15. Morien, Roy (2013) Seeking a New Paradigm for Software Project Management: Reference Disciplines for Agile Development, 5th International Conference on Internet (ICONI 2013).
16. Morien Roy (2014) Streamlining Business Computing Education, 6th International Conference on the Internet (ICONI 2014).
17. Morien, Roy (2017) Business Computing Education: A Radical Approach for Efficient Streamlining of an Effective Education Process and Relevant Curriculum, International Journal of Advanced Media and Communication (to appear) 1.
18. Morien, Roy (2015) Agile and Agility in Computer System Development Education, 7th International Conference on Internet (ICONI 2015).
19. Morien, Roy (2018) Pedagogical Agility and Agile Methodologies in Computer System Development Education, International Journal of Advanced Intelligence Paradigms (IJAIP), (to appear) DOI: 10.1504/IJAIP.2018.092943.
20. http://www.abc.net.au/news/2014-11-11/uts-new-business-schoolbuilding-defies-convention/5883506?section=nsw.
21. http://www.abc.net.au/news/2016-02-18/no-exams-lectures-or-nerds-inthe-tower-csu-engineering/7180538.
22. http://elearn.curtin.edu.au/.
23. Morien Roy (2004) Insights into Using Agile Development Methods in Student Final Year Projects, Informing Science + Information Technology Education Joint Conference (INSITE2004), Central Queensland University, Rockhampton, (paper accepted in "Best Paper" Category, for presentation, and awarded Conference Silver Medal). And in Information Systems Education Journal http://isedj.org/4/44/.
24. Morien, Roy (2005) Student Experience of Using Agile Development Methods in Industrial Experience Projects, Information Systems Education Conference (ISEC2005), Columbus, Ohio, October 2005). And in Information Systems Education Journal http://isedj.org/4/103/.
25. Alex Handy, Making computer science class more like the real world, http://sdtimes.com/making-computer-science-class-more-like-the-realworld/.
26. Derek Thompson, The Thing Employers Look For When Hiring Recent Graduates, http://www.theatlantic.com/business/archive/2014/08/thething-employers-look-for-when-hiring-recent-graduates/378693/.
27. http://www2.smumn.edu/deptpages/tclibrary/search/subjects/ed_scholarl y_journals_11.pdf.