The Software Engineering Department
Achi Racov Engineering School
Kinneret College on the Sea of Galilee

The Fifth Kinneret Conference on Software Engineering Education

February 21th, 2017

Presentation Abstracts

Welcome to the Fifth Kinneret Conference on Software Engineering Education!

A few months ago, the Israel Ministry of Finance announced that there is a severe shortage of engineers, especially in the computer-related areas, and, consequently, import of foreign engineers is being considered. We all know that the number of candidate students for computer science and engineering is not growing sufficiently fast, and the rate of students' dropout is still large. However, in the long term, the quality of Software Engineers significantly outweighs their quantity, and therefore Software Engineering Education must remain in focus and be constantly improved.
In order to increase the effectiveness of Software Engineering Education, with respect of the needs of the industry, the industry should take a more active role in this process. This can be achieved in several ways:

- Participation in Advisory Councils of the academic institutions;
- Encouraging and supporting practitioners to take an active part academic teaching;
- Encouraging and supporting students' projects in industry and industrial internship;
- Inviting students to attend courses and workshops in industry;
- Establishing activity centers (sites) in or near peripheral academic institutions.

It is not just a coincidence that the Software Engineering Education conference is conducted at Kinneret Academic College. The Software Engineering Department was established six years ago with great motivation and intentions to incite and instil a new spirit and inspiration to the education of Software Engineering students, preparing them for a life-long career as professional Software Engineers in the well-advanced Israeli Software Industry. As we are now witnessing our graduates accepted so well in the best jobs offered by industry, we are encouraged that our program is well on the right tracks. This conference is another important layer in this trend.
We would like to wish you an enjoyable and fruitful conference, and we look forward to seeing you again in next years' conferences.industry, we are encouraged that we might be on the right tracks. This conference is another important layer in this trend.
We would like to wish you an enjoyable and fruitful conference, and we look forward to see you again in the next years to come.

Prof. Amir Tomer,
Head,
Software Engineering
Department

Prof. Avraham Shitzer
Dean,
Achi Racov School of
Engineering

08:30	09:30	***Registration***	
09:30	10:00	***Opening and Greetings*** ***room 812***	
		Prof. Amir Tomer, Head of SE Department **Conference Organizer** Prof. Shimon Gepstein, President, Kinneret College	Ruthy Kaidar, Chief Startups and Developer Relations Lead, Microsoft Israel **Conference Chair** Prof. Avraham Shitzer, Dean, School of Engineering
10:00	11:00	***Morning Guest Keynote***	
		room 812	
		Books, MOOCS and More: How I Teach Programming ***Prof. Bertrand Meyer, ETH Zurich***	
11:00	11:15	***Coffee Break***	
11:15	13:15	***Morning 1: SEE - Students Aspects***	***Morning 2: SEE Beyond Academy***
		room 812	***room 813***
		Chair: Dr. Reuven Gallant, JCT	***Chair: Prof. Opher Etzion, YVC***
		Software engineering studies - mirroring the real world Michal Chalamish, Ashkelon Academic College	**YUP - A Curriculum Integrated Innovation Center** Opher Etzion, Yezreel Valley College
		Evaluating the Engagement of Software Engineering Students Arnon Sturm, Ben - Gurion University	**Jump into a working place** Zeev Copel, Philips Medical Care
		Just Enough Rigour (REJ): Teaching Modeling Semantics sans Mathamatical Formalism Reuven Gallant, Jerusalem College of Technology	**Information Projects Quality Model** Ran Berman, State University of Moldova
		To TDD or not to TDD: Incorporating Test Driven Development into Software Engineering Education Naomi Unkelos - Shpigel, University of Haifa	**How Developers Make Design Decisions about User's Privacy** Oshrat Ayalon et al, Tel Aviv University
13:15	14:15	***Lunch***	
14:15	15:00	***Afternoon Guest Keynote***	
		room 812	
		From Nand to Tetris: Applied Computer Science from the Ground Up Prof. Shimon Shocken, IDC Herzliya	
15:00	16:30	***Afternoon 1: Project - Based Learning***	***Afternoon 2: Collaborative Learning***
		room 812	***room 813***
		Chair: Dr. Reuven Gallant, JCT	***Chair: Prof. Opher Etzion, YVC***
		Are Students Prepared for their Capstone Projects? Mira Balaban and Arnon Sturm, Ben - Gurion University	**Share If You Care: Collaborative and Gamified Technological tools to Enhance Software Development Education** Naomi Unkelos - Shpigel, University of Haifa
		The Neonatal Intensive Care Unit (NICU) Digital Revolution Dror Ben Ami et al, Zefat Academic College	**CAPELLA: An Integrated and Adaptive Ecosystem for Life - Long Learning** Amir Tomer et al, Kinneret College
		Robotics Based Projects Oriented SE Rina Zviel - Girshin, Ruppin Academic Center	
16:30		***Closing***	

Guest Keynote

Books, MOOCS and More: How I Teach Programming

Bertrand Meyer
ETH Zurich, Eiffel Software, Innopolis University

Bertrand.Meyer@inf.ethz.ch

Teaching programming today raises many challenges, beginning with the wide diversity of students' initial backgrounds and the high expectations of students' future employers. For the past 14 years I have taught the introductory programming course at ETH Zurich, whose model has also been adopted by other universities such as Innopolis. The course places a constant emphasis on not just programming but quality programming, uses object-oriented concepts and the Eiffel language, teaches concepts of Design by Contract, covers a variety of programming techniques such as event-driven programming, and extends over some advanced topics not generally touched by introductory courses.
The course is supported by the "Touch of Class" textbook (Springer) and has given rise to several MOOCs (Massive Open Online Courses) available from EdX under the title "Computing: Art, Magic, Science" (CAMS 1 and 2). The effort has also led to the development of advanced cloud-based tools supporting the teaching of programming, available for all major programming languages (codeboard.io) and used by a fast-growing number of universities for diverse courses.

A complementary experience is the "Distributed and Outsourced Software Engineering" course project, running every year since 2007, where students from different universities around the world cooperate in developing a system; this effort has both helped and benefited from extensive research into the challenges and techniques of distributed software engineering.
The talk will present the pedagogical principles that underlie these education projects, the role of new technology such as MOOCs and the cloud, and my assessment of this multi-decade experience.

Bertrand Meyer is a professor for software engineering at ETH Zurich and at Innopolis University, where he is also the Director of the Software Engineering Laboratory. Prof. Meyer is the inventor of the Eiffel programming language and the founder and CEO of Eiffel Software. Prof. Meyer is the author of a number of well-known books on diverse software engineering topics.

Session: Morning 1
Software Engineering Education – Students Aspects

Chair: Dr. Reuven Gallant, Jerusalem College of Technology

Software engineering studies – mirroring the real world

Michal Chalamish, Ashkelon Academic College and Bar Ilan University

michal.chalamish@gmail.com

Engineering is defined as "the art or science of making practical application of the knowledge of pure science..." [www.dictionary.com]. Practicality is the essence of engineering, thus the teaching of software engineering must put a special emphasis on the practical issues concerning a full software development process.

Following this principle, the studies of software engineering at Bar Ilan University and at Ashkelon Academic College include the following:

(1) Theoretical studies of the various software development processes and models. This provides the student with the knowledge of how a software project should be managed in either a plan-driven or an agile environment. The student is provided with a bird's eye view of the development process, giving him (or her) an advantage over a programmer.

(2) Practical studies of the various UML diagrams and of some of the most used design patterns. These give the student the ability to discuss design and implementation issues in future team meetings, during studies and later on, at work.

(3) Development of an actual project, chosen by the students, producing a real software product. The students experience working under a limited and harsh time table, having to hand the documents and the project in on time. They learn how to prioritize requirements, when they realize they are unable to meet all of them. They end up with their first software product,

KINNERET College on the Sea of Galilee

full of pride, usually, after having to deal with various and unexpected difficulties, as in the real world.

(4) Most students are accustomed to make their own programming decisions since in most courses team work is not allowed. While developing their software product they work in couples or trios. Single work is not allowed here. They have to deal with scheduling meetings of all team members. They have to make design and implementation decisions which all team member should agree on. They also have to deal with integrating bits of code written by different team members into a single, working, software.

(5) During the development process, the teams are required to hand in a number of documents: (i) Project Proposal, (ii) Software Development Program, (iii) Software Requirements Specification, (vi) Software Design Specification. The Actual development is divided into a number of milestones. The students experience having to express their progress and to justify their decisions in writing.

(6)When we are able to, we bring in engineers from the industry to give the students a short lecture on the way a real project is being developed in the company where they work. The students can get an impression of the "real world" and ask questions.

(7) While developing their product, the teams are required to present the current development status in front of the entire class. For most students, this is the first opportunity they encounter to stand in front of an audience and talk. Since the audience is their class members it's a friendly crowd, making it a relatively easy first experience.

Dr. Michal Chalamish, is a senior lecturer in the Department of Computer Science at Ashkelon Academic College. Her research interests focus mainly on Artificial Intelligence, Negotiations and Multi Agent Systems (MAS). She has been teaching the Software Engineering course in the faculty of Engineering at Bar Ilan University and the Seminar in Software Engineering at Ashkelon Academic College for the past 6 years.

Evaluating the Engagement of Software Engineering Student

Arnon Sturm, Ben-Gurion University of the Negev

sturm@bgu.ac.il

The new age of digital media and devices requires new investigation into its apparent and potential effects on motivation, attitude and engagement (MAE) of students towards their studies, particularly in STEM fields. The digital age is having a multifaceted affect including huge amounts of distraction and stimulus, major shifts in student work opportunities and study patterns, as well as instant access to information and knowledge of all kinds.

Engagement, which in our view is the overarching concept, has three inter related aspects: cognitive, behavioral, and affective. Cognitive addresses the extent to which students are attending to and expending mental effort in the learning tasks encountered; Behavioral addresses the extent to which students are making active responses to the learning tasks presented; and Affective addresses students' investment in, and their emotional reactions to, the learning tasks. Student engagement is considered one of the most influential factor in educational success, with a wealth of research aimed at determining how best to foster this in higher education.

Our (subjective) initial observation, in general, is that MAE of students towards learning within STEM programs in higher education institutes and in particular in software engineering is decreasing over the years as reflected in students' behavior and in their explicit statements. For example, at Ben-Gurion University of the Negev, although students have the ability to learn many things quickly, we see that classrooms have limited attendance, there seems to be limited desire to learn within the university, there is a preference towards hands-on experience rather on fundamental principles, the culture is increasingly permissive, there is a decrease of reading/summarization capabilities, there is an increase in use of on-line educational resources including consulting experts, resulting in an increasing reliance on just-in-time available knowledge, which limits the motivation for and practice of deep learning.

KINNERET College on the Sea of Galilee

In this work take a first step into measuring the extent to which our initial observations hold. Specifically, we used a version of the National Survey of Students Engagement (NSSE) to learn about the students' perceptions over their studies. The questionnaire includes various engagement indicators such as: Higher-Order Learning, Reflective & Integrative Learning, Learning Strategies, Collaborative Learning, Student-Faculty Interaction, Effective Teaching Practices. We report on our findings and discuss their implications with respect to the software engineering study programs.

Arnon Sturm holds is a faculty at the department of Software and Information System Engineering in Ben-Gurion University of the Negev. His research interests include modeling approaches, human aspects of software engineering, and software engineering education.

Just Enough Rigour (REJ):

Teaching Modeling Semantics sans Mathematical Formalism

Reuven Gallant, JCT Lev Academic Center
(Jerusalem College of Technology)

rgallant@alum.mit.edu

Much ink has been spilled regarding the trials and tribulations of adapting formal methods to the needs of software engineering practitioners With the exception of computer scientists with a passion for algorithm design and optimization, a plethora of Greek letters and symbols can be an anathema to those whose first love is writing code. The advent of graphical modeling languages such as UML and supporting tools that generate production quality code, executable modeling behavioral simulations for bridging the gap between formalism and coding. This paper proposes, with illustrative examples, an exploratory learning modality, by which the student and practicing engineer can investigate and empirically learn the semantic mapping of UML syntax to the semantic domains of system instantiation and reactive behavior.

Reuven Gallant received his B.S.E.E and M.S.EE from M.I.T. and Ph.D. from Yale University. He worked at Sikorsky Aircraft as group leader for Avionics Digital Systems and at Israel Aerospace Industries, as a senior researcher in software process improvement and software development methodologies. He is presently vice-chairman of the Computer Department, JCT Lev Academic Center, with responsibility for the software engineering program. Present research is in cognitive challenges in Model Based Engineering.

KINNERET College on the Sea of Galilee

To TDD or not to TDD: Incorporating Test Driven Development into Software Engineering Education

Naomi Unkelos-Shpigel

Naomiu@is.haifa.ac.il

In recent years, TDD (Test Driven Development), has been in practice in many software development projects, enhancing the development process and its' outcomes. As agile development becomes more and more popular, tests are often written and performed prior to or during code development.

However, in software engineering degree curriculum, we often teach the students to create software tests after the code has already been written. This usually results in students focused more on code writing, and less on thinking and planning possible faults and problems in their code.

In this lecture, I describe the use of TDD in a Software Engineering course, taught in University of Haifa, to last year IS students. The students built a full-stack java project, in groups of four. They developed an enhanced game of snakes and ladders, along with a collaborative environment of course-related multiple answers questions. The students worked in three iterations, guided by the SCRUM framework. In the first iteration, they designed the architecture of the project. In the second iteration, they were asked to write the control layer of the system, and the third iteration, they were asked to finish the project.

Before the second iteration, the students learned what TDD is, and had a 90 minutes workshop when they practiced writing unit tests for their peers, and writing code based on those tests.

During the second and third iteration of the projects, students were asked to write tests for a feature of their system that will be later on developed by their peers.

The presentation will include the description of the work process, where this method was used, as well as feedback from students, and the interesting results which were obtained from their feedback.

Naomi Unkelos-Shpigel is a PhD student at the IS department, University of Haifa, advised by Dr. Irit Hadar. She lectures at Kinneret College, Braude College and University of Haifa. She has worked for several years in real-time programming and in Information systems design and implementation. Naomi graduated with honor BSc in Information Systems at the faculty of Computer Science, Technion. She graduated her Master's degree at the IS department, University of Haifa.

Session: Morning 2
Software Engineering Education
Beyond Academy

Chair: Prof. Opher Etzion, Yizrael Valley College

YUP – A curriculum integrated innovation center

Opher Etzion, Yezreel Valley Academic College

Ophere@yvc.ac.il

The Information Systems Department of the Yezreel Valley Academic College has placed the education for entrepreneurship and creativity as a flagship activity within the curriculum. It is a combination of academic courses, and an accelerator for selected projects. While innovation centers exist throughout academia, many of them are disjoint from the academic program. Our unique characteristics are: (1). the innovation center is fully integrated with the curriculum; (2). the focus is on societal initiatives.

The education starts in the first year. The first course is a course about business models which serves as a prerequisite for all courses in this

KINNERET College on the Sea of Galilee

area. In the second semester the students experience ia workshop for creativity and entrepreneurship. The lectures are given by a team of three people: one from the economics side, one from the start-up practice side, and one from the creativity side; during the workshop, the students experience a whole day marathon to start the work on their own initiatives which is presented at the end to external evaluators. In the second year, the students can take a seminar in societal entrepreneurship, in which they are working on societal ideas, they also learn marketing in social media. In the third year the students take the senior project. In the past all projects were development of information systems, typically in small enterprises. Currently, we encourage the students to dream, and work on their own initiatives. The students write detailed design, and implementation of a running prototype. These projects are presented to a panel of evaluators, each project is going through thorough review of external evaluator. The last phase is accelerator that helps mentoring a group of selected projects. All these activities is part of YUP – The Yezreel UP innovation center. The activity is in different phases of implementation, and all of it is implemented within the current school year.

In the talk we'll report on YUP and all its derivatives, partnerships with other departments in the college, and with external factors, the vision going forward, and about each of the activities outlined above. Examples of running projects will be presented.

Opher Etzion serves as Professor of IS, Head of the Information Systems Department and head of the Technological Empowerment Institute in Yezreel Valley College. During the years 1997-2014 he served in various roles in IBM, most recent Senior Technical Staff Member and Chief Scientist of Event Processing. Prior to joining IBM in 1997, he has been a faculty member and Founding Head of the Information Systems Engineering program at the Technion, and held professional and managerial positions in Sapiens and in the Israel Air-Force. He won several prestigious awards over the years, such as the Israel Air-Force highest award for introduction of new technologies towards widely usage, IBM Outstanding Innovation Award (twice) and IBM Corporate Award (the highest IBM award) for the pioneering work on event processing.

Languages - Jump into a working place

Zeev Copel, Philips Healthcare - Israel

zeevicopel@gmail.com

A new and fresh SW Engineer is joining a matured SW Development team. Welcome. Wait. Are you ready to jump in?

Facing many SW Engineers from the interviews phase, that just graduated their engineering degree, and later on during their first months in the SW Development Industry, is the right timing to put a mirror in front of us to see how these excellent engineers (bolded and not by mistake) are well trained to code and how far they are from the Industry practical needs.

I am sorry to disappoint, but SW Development is not only coding and mathematics. Yes. Not taking it back. Not meaning that the investment in these areas is redundant, but it is not enough to "Jump in" as there are more expectations.

SW Development is starting from Understanding, Planning, Designing, Discussing, meeting the customer (yes, the SW Engineers meet the customer), Developing, Testing, Unit Testing, Automate, Document, working according to Development methodology (Aka Agile), meeting the standards, knowing that there are standards and leading. Which is taking me to the next question: "Does SW Engineer Ownership being discussed in the academy?"

New Engineers, not just "fresh" ones, are joining a new working place with a lot of motivation. It is something that happens to an engineer only few times during his career. While experienced engineers know to be ready for misalignments between the expectations and the reality, for new SW Engineers it might end with a big disappointment which drags the motivation down, and the industry in Israel is losing in few months great potential with great motivation.

As today's SW Development methodologies are becoming more agile, it is enabling the SW Engineers to define their timing commitments and to plan, instead of the old tradition where the "boss" is defining when the task will end. I am calling it "Power To the People", But, it hands the SW Engineers the Ownership on all the development activities, when the managers are guiding and directing. It is a very important point, and we see many engineers that are not able to adopt. Experienced and juniors.

Let's invest in the Juniors first, and let's do it as early as possible.

The term Ownership is obligating us to cross this discussion into the sociology area, which is less discussed in the Engineering studies, and the bundling needs to happen in the academy, since it is already in the industry and the expectations are raising.

You can imagine it as you are standing behind the potential SW Engineer that is standing on the diving board and planning to jump, while you need to equip him well so he will have a perfect dive and he will swim easily after raising his head above the water.

Well, are you ready to push?

Zeev Copel is carrying a SW Engineering degree from Ort Braude College since 2004. Also carries an Electrical Practical Engineer degree. Jumped into a working place on 2003 as a SW tester in Marvell Semi-Conductor and became a SW Engineer later on. On the middle of 2005 joined Rafael as an Embedded SW Engineer which was the "real" school in the industry. On Sep-2008 joined Philips as a SW engineer and became a global Development Manager for a group of 75 Engineers in Israel and India.

Information Projects Quality Model and the Global Volume of Data

(Providing Quality of Information Projects)

Ran Bergmann, Dr. Julia Sirota

Department of Informatics, State University of Moldova, Chisinau, Republic of Moldova

bergman.rani@gmail.com , jsirota1976@gmail.com

he global volume of data. Today, if a certain e-mail user gets just 5 e-mail letters per day, it is not difficult for her to distinguish between important and "garbage" messages. However, if we speak about thousands messages per day, is becomes too complicated and takes too much time just to read the messages, not to classify them. In our era of information and "big data", any users, even the most clever ones need some technological help in order to manage the information they get. According to IDC statistics, from 2005 to 2020, the global volume of data – the digital universe – will grow by a factor of 300, from 130 Exabyte to 40,000 Exabyte, or 40 trillion gigabytes (more than 5,200 gigabytes for every man, woman, and child in 2020). From now until 2020, the digital universe will approximately double every two years. IDC estimates that by 2020, as much as 33% of the digital universe will contain information that might be valuable if analyzed, compared with 25% today. While earlier, about one-half of the digital universe sprang forth just from the United States and Western Europe, and emerging markets accounted for less than 20%, the share of the digital universe attributable to emerging markets was up to 36% in 2012 and will reach 62% by 2020 [1]. The increase in quantity and decrease in quality of information [2]. If the problem of the volume of the data was not so severe, there was no necessity to manage the quality of information [3] [4].

- The requirement to establish a quality model has been felt by users for the purpose of evaluating the software quality quantitatively and qualitatively.
- Information quality literature has provided a great amount of proposals for assessing the quality of information, but there is still a need to develop frameworks for assessing and improving the quality of information from the information consumer and the organizational point of view in the perspective of the information project classification.

KINNERET College on the Sea of Galilee

- Moreover, for each dimension there must be set a clear definition what it represents, in order to be able to compare it for any type of Information Project (i.e. Information System) throughout its life cycle.

Ran Bergmann Moldova State University, PhD part-time student since November 2012, Mathematics and Informatics Faculty. Over 15 years' experience in information systems analysis and realization, which includes experience in implementing ERP and PLM information systems, engineering and logistics modules. Lecturing experience in information systems analysis courses and information system project management.

Julia Sirota Kinneret College, Carmel College, lecturer and academic advisor , PhD, Economics and Management. Over 20 years' experience in teaching, statistics analysis research, in courses of Economics, Industrial Engineering, Decision Making, Game Theory.

1. IDC. (2014). IDC Big Data and Business Analytics Forum 2014.

2. Torr, J. D. (2003). The Information Age: Greenhaven Press.

3. Bergmann, R. (2015). Quantifying Information Quality. Studia Universitatis Moldaviae, 7(87): 86-97.

4. Bergmann, R. (2016). Information Projects Quality Model. Information Projects Quality Model, 2(11): 38-56.

How Developers Make Design Decisions about Users' Privacy

Ayalon Oshrat[1], Toch Eran[1], Hadar Irit[2] and Birnhack Michael[1], [1]Tel Aviv University, [2]University of Haifa Department of Informatics, State University of Moldova, Chisinau, Republic of Moldova

{oshratra, erant, birnhack}@post.tau.ac.il, hadari@mis.haifa.ac.il

Privacy by Design (PbD) is a concept calling to design information systems (IS) in a privacy-respectful way, thus, mitigating privacy threats from the very beginning. While PbD holds many promises, scholars criticize it for being abstract and difficult to implement. The challenge to implement PbD highlights the need for studying developers' privacy decision-making processes. As developers are the ones who can make or break PbD implementation, we need to ask what are the barriers of designing systems with privacy in mind.

We are interested in understanding the relationship between IS developers' professional privacy attitudes and practices and between several factors, including organizational (organizational privacy climate, business and legal contexts), professional (exposure to privacy knowledge sources) and personal (personal perceived privacy). As we aim to explore behavior and attitudes of a large population, we chose to conduct an online survey. The survey was targeted to participants who work in the area of IS development, and was available between February and November, 2015. Participants were recruited using rolling sample and by advertising the survey in relevant forums in online social networks, such as Facebook and LinkedIn.

The survey's design yields seven constructs, according to the research objective. Our dependent variables are represented by the constructs professional privacy attitudes and privacy practices. We found that security is perceived as important and frequently taken care of by the developers. This confirms previous qualitative results that point to security as the way through which many developers view privacy (Hadar et al., 2014).

Next, we performed a regression analysis and Spearman correlation tests, including Bonferroni correction, in order to investigate which

KINNERET College on the Sea of Galilee

variables affect professional privacy practices. Within the regression we included all the independent constructs. In addition, we referred to part of the developers' demographics and to the organizations' features. The regression model results (adjusted R2=0.289) points to three significant predictors, predicting professional privacy practices: (1) privacy climate (β = 0.305, p = 0.034), which refers to the participants' perceptions of how their organization refers to privacy; (2) personal perceived information control (β = 0.314, p = 0.01), which refers to the participant's personal perception of information control from the point of view of an end-user; and (3) former personal information collection (β = 0.117, p = 0.045), which refers to the participants' former experience in collecting and storing personal information.

We further explore the relationships between professional privacy practices and the study's other continuous constructs. We found that professional privacy practices is significantly positively correlated with organizational privacy climate and with exposure to privacy knowledge sources, which refers to the participant's frequency of using privacy knowledge sources.

The aim of this study was to understand how personal and working environment features affect developers' professional privacy attitudes and practices by surveying IS developers. Understanding these effects will help to understand how privacy controls and user interactions are designed, as well as providing some guidelines for enhancing privacy in IS design, which can have a significant effect on the developers' communities.

References

Hadar, I., Hasson, T., Ayalon, O., Toch, E., Birnhack, M., Sherman, S., & Balissa, A. (2014, March). Are Designers Ready for Privacy by Design? Examining Perceptions of Privacy Among Information Systems Designers. In 2014 TPRC Conference Paper.

Oshrat Ayalon is an information systems scientist with an interest in human-computer interaction and usable privacy and security; she is a PhD candidate at the Department of Industrial Engineering of Tel Aviv University.

Eran Toch is a senior lecturer (tenure-track) at the faculty of Engineering at Tel-Aviv University. His research is in the field of usable security and privacy, human-computer interaction, and information systems. Eran's research group is working on various engineering challenges that revolve around computationally understanding human behavior, and applying this knowledge to solve real-world security and privacy challenges. Prior to joining Tel Aviv University, Eran was a post-doc fellow at Carnegie Mellon University, School of Computer Science. Eran has a Ph.D. from the Technion – Israel Institute of Technology.

Irit Hadar is a senior lecturer (tenured faculty member) at the Department of Information Systems, University of Haifa. She is the Head of the Software Architecture Laboratory at the Caesarea Rothschild Institute for Interdisciplinary Applications of Computer Science. Her main research area is cognitive aspects of software architecture, design, and analysis. She holds a Ph.D. from the Technion – Israel Institute of Technology.

Michael Birnhack is a Professor of Law at the Faculty of Law, Tel Aviv University. Michael studies and teaches privacy law; he was a member of the Israeli Public Council for the Protection of Privacy; a member of an expert committee on the Israeli data protection law, and a sub-contractor for the EU Commission on the matter of the adequacy of the Israeli data protection legal regime. His current privacy studies explore privacy by design, biometrics, smart cities, and school surveillance.

KINNERET College on the Sea of Galilee

Guest Keynote

From Nand to Tetris: Applied Computer Science from the Ground Up

Prof. Shimon Schocken, IDC Herzliya, Israel

schocken@gmail.com

We present an educational approach that synthesizes many abstractions, algorithms, and data structures learned in CS courses, and makes them concrete by building a complete computer system from the ground up. The methodology is based on guiding students through a of 12 projects that gradually construct and unit-test a simple hardware platform and a modern software hierarchy, yielding a surprisingly powerful computer system. The hardware projects are done in a simple hardware description language and a hardware simulator supplied by us. The software projects (assembler, VM, and a compiler for a simple object-based language) can be done in any language, using API's and test programs supplied by us. We also build a basic OS. The result is a general-purpose Von Neumann machine, simulated on the student's PC. We start the course (and this talk) by demonstrating some computer games running on this platform. The approach is completely self-contained, requiring only programming as a pre-requisite (joint work with Noam Nisan, Hebrew University). This work has led to a best-selling MIT Press textbook, a TED talk, two Coursera courses, and courses that are now taught in more than 100 universities and schools around the world. All our materials are available freely and in open source in www.nand2tetris.org.

Shimon Schocken is the founding dean of the Efi Arazi School of Computer Science at IDC Herzliya.He was also a tenured professor at NYU (1985-1995), a visiting professor at Harvard (2005) and Stanford (2012), and chairman of the Israeli ministry of education's computer science committee. Shimon is also interested in early-age mathematics education, and is co-founder of Matific, a software company whose award-winning math learning games for mobile devices are used in more than 30 countries and languages.

Session: Afternoon 1
Project-Based Learning
Chair: Dr. Arnon Sturm, Ben-Gurion University

Are Students Prepared for their Capstone Projects?

Mira Balaban, Ben-Gurion University of the Negev
Arnon Sturm, Ben-Gurion University of the Negev

mira@cs.bgu.ac.il, sturm@bgu.ac.il

Software Engineering (SE) studies have an embedded built-in contradiction: software engineering targets complex, large-scale systems that require stability, interoperability and reliability, but the education framework can handle only small, short-term, relatively simple software. In an attempt to address this challenge, the standard software engineering curriculum usually includes a capstone project, intended as a framework for combining the overall plethora of tools taught during previous years.

The capstone project in our program at Ben-Gurion University of the Negev is a group project of 2-5 students that construct a software solution to a problem, where each group selects a specific problem. Project development specification imposes agile iterative development, with emphasis on careful modeling, testing and validation, prototyping and requirement analysis. In general, the resulting projects are impressive and demonstrate high software development capabilities.

Nevertheless, while SE students gain quite wide SE-related knowledge and capabilities during their studies, there is no opportunity for instructive integration of methods into a working system. Furthermore, as the capstone projects are group-specific, they do not allow for effective SE-education. Frequently, associated SE products are neglected in favor of direct programming tasks.

In order to address this gap, we devised, as part of the curriculum, a project workshop where the students cope with a natural SE task in a guided manner. The goal of the workshop is to acquire experience in using SE methodologies, methods and technologies, and prepare the students for their independent fourth year capstone project. In particular, the learning outcomes are the following:

- The students will be able to select SE relevant methods and techniques

for their future development;

- The students will be able to reproduce the software development process;
- The students will be able to analyze alternative design solutions;
- The students will be able to apply modeling activities within the various development stages;
- The students will be able to assess the challenges in team work; and
- The students will be able to compose working software in multiple versions.

The workshop is centered on a single software development task, which enables specification of evaluation criteria and deep and tight guidance. This is done via joint lectures and group mentoring. Project development is done by versions, allowing to demonstrate development principles. Version specification is provided by the course stuff, and includes functional, non-functional, and invariant constraint requirements.

The efforts required on the student part are enormous, yet both instructors and students feel that this is a highly productive framework. We believe that this framework achieves its goals, and equips students with an increased level of software development maturity. In this work, we elaborate on the goals of the workshop, the way of achieving these goals, and perform reflection both from the students and instructors point of view.

Mira Balaban is a faculty at the department of computer science in Ben-Gurion University of the Negev. Her research interests include conceptual data modeling, object-oriented system analysis and design, software engineering, knowledge representation, theory of database and knowledge base systems, and programming languages.

Arnon Sturm is a faculty at the department of Software and Information System Engineering in Ben-Gurion University of the Negev. His research interests include modeling approaches, human aspects of software engineering, and software engineering education.

The Neonatal Intensive Care Unit (NICU) Digital Revolution

Ben-Ami D., Zefat Academic College
drorb2@zefat.ac.il

Cohen M., Shmailov Sh., Rubin R. , Zefat Academic College

meiravbd@gmail.com;shmuelsh.25@gmail.com;robyn.rubin@gmail.com

Keywords:
NICU(Neonatal Intensive Care Unit), NFC (Near Field Communication), SCRUM, Agile, Sprint

The digital revolution is multifaceted revolution.
As a 'revolution' it has common attributes and characteristics to other revolutions, such as: industrial, technological, information & knowledge, communication, philosophy, political and others. But in addition, the digital revolution has specific unique characteristics of its own. The digital revolution is characterized by its fast trend and speed; besides significant influences on environments, human beings, societies and cultures. Its impact reshapes the global workforce [1].
The NICU daily working environment is characterized by its complexity [8]:
The diversity of neonatal intensive care is wide. Some of the neonates are premature, some suffering from different clinical problems such as: pulmonary, cardiac functioning, hematological, very low weight, etc. All premature neonates require special continuous monitoring and treatments by professional staff. Emergency clinical surgical interventions are frequently done in the NICU.
The digital revolution offers some special and unique capabilities for the professional staff, to support them with pertaining treatments. Merging communication channels between different devices; hardware and software based, while keeping to strict and reliable standards, can empower significantly the entire NICU's functioning.
A prototype of an Information System was planned and implemented, based on SCRUM methodology (as an expansion of the Agile manifesto). SPRINT actions and scenarios have been planned, examined, tested and validated. In order to achieve an immediate availability and accessibility

of clinical data, the NFC [2][3][4] technology was merged into the development. Using NFC opened an immediate channel to the hospital databases in general, and specifically to those regarding the NICU activities. Specific unique interfaces were defined and established, such to SAP [5][6][7], PACS (radiology system) and others.
Prime critical issue had to be considered, whilst we have direct access to the data:
"Too little info" is not enough for the staff; but "too much info" might be irrelevant and can be cumbersome for the entire system and its goals. Thus, it's necessary to define properly what/which kinds of information should be transferred between systems, in order to support the staff with their "real" needs.

Beyond this overview, we need to consider the following:
FIRST: the "narrow" and LOCAL point of view, regarding the project itself: How the project can be managed in the LIVE environment, under so many limits such as: time, budget, concepts, understanding the environment, the NICU and Poriya policies (as an organization), technological limitations and others.
SECOND: The WIDE overview: relationships between academy & industry. How the academy keeps in stride with accelerated, quick, rapid trends in the industry on the one hand, and how all the studied matter can support these changes, how can us as lecturers and professional staff can leverage (influence) and merge the trends in our studied courses.

Main Conclusions:

1. ***"MUST"***: lecturers and the professional academy staff MUST be involved DIRECTLY in industrial projects
2. ***"To strive to"*** more projects during the course of study: small and medium projects during the 3-4 year of study and NOT just on the final project. Pragmatically, such study leverage and empower a wide spectrum of knowledge: theoretical and implemental
3. ***Conceptually***, a model of joint-ventures for final projects should be considered, by GOV support (budget, means) to such projects. Some of them can be expanded later into real industrial practical and useful systems and generate the basics to High-Tech companies establishment or any joint-ventures

Ben-Ami Dror, Last year PhD. Studies, Moldova State University, Moldova: Informatics (Mathematics & Computer Sciences Department);, Lecturer at Zefat Academic College, Zefat, Israel; Independent advisor & software developer

Rubin Robyn, Community Information Systems – Zefat Academic College Studying for MA in Information and Knowledge Management - Haifa University.
Working at: Baruch Padeh Medical Center - Poria, Neonatal Intensive Care Unit - Research Coordinator.

Cohen Michal, Shmailov Shmuel, Graduates, Community Information Systems – Zefat Academic College

KINNERET College on the Sea of Galilee

Robotics Based Project Oriented SE

Zviel-Girshin, Rina, Ruppin Academic Center

rinazg@ruppin.ac.il

One of the important features of software engineering (SE) course is its ability to summarize and implement previously acquired knowledge. Exist several types of software engineering course: from basic course that teaches software models, requirements, design, management, testing and maintenance to the course that teaches advanced topics in software engineering including client/server SE, advanced topics in programming, web engineering, component-based SE, Computer-Aided SE (CASE), emerging trends in SE and software systems. One of the great features of the SE course in the higher education institutions is the ability of the instructors to create their own variation of the course syllabus, to choose the list of topics and amount of time for each topic.

Using her position as teacher for 18 years and head of knowledge engineering and robotics lab, the author has very successfully tried to enhance interest to SE by adding new features and challenges to SE course. For several years now students enrolled to the course have to implement an accompany project and creatively solve problems, design and implement SE products. In a small period of time students have to choose a project that they want to implement in the field of robotics, internet of things or autonomous system. This project-driven approach adds a so called "fun factor" needed to engage students, as well as the practical realism of engineering projects.

One of the important focuses of an advanced SE course is to perform a simulation of a real life engineering process, to learn how to build software, starting from a set of requirements and ending with a final product that satisfies the specifications and is free of bugs and defects. Therefore adding a robotics project that requires a working robot at the end of the process is very important.

At the beginning of the course, students had to choose partners for their project. Later each team required to design an artifact, write all the requirements, code and perform testing. The project had certain milestones that must be met throughout the development timetable.

Among the issues for the research were:

- increasing students creativity,
- knowledge acquisition in new areas of CS and/or engineering,
- better understanding of team work,
- exploring and adding new features vs. timetable,
- better understanding of an engineering process.

The decision to add other disciplines such as robotics, networks, cloud programming or human computer interaction to the SE course proved to be very challenging. Both the course staff and students had to work hard and to learn new things all the time. At the end of the course, students were able to work in teams and across teams, learned to exploit strong points of other team members (hardware vs. software) and managed to produce a "working" artifact that was their own creation and imagination. We hope and believe that our experiment to add robotics to SE course will grow and inspire others to use our research and conduct more experiments in this field.

Dr. Rina Zviel-Girshin, Head of Knowledge Engineering and Robotics Lab, Ruppin Academic Center; Head of compilation field at Open University. Fields of interest are AI applications, Programming Languages, Educational Software, Robotics, Internet of Things, STEM Education.

KINNERET College on the Sea of Galilee

Session: Afternoon 2
Collaborative Learning
Chair: Prof. Amir Tomer, Kinneret Academic College

Share if You Care: Collaborative and Gamified Technological tools to Enhance Software Development Education

Naomi Unkelos-Shpigel

Naomiu@is.haifa.ac.il

Project-based courses, where the students are required to develop a prototype as final assignment, are an integral part of software engineering degree curriculum. The teachers of these courses face several challenges, such as equal participation of all students in the workload, and creating projects which will be both of high quality and utility, all in parallel to teaching a large amount of theoretical background.

Teaching several courses of this nature, mainly those requiring developing a full-stack project or a website, I developed and incorporated a number of techniques which I believe can be used by other lecturers teaching similar courses. First, the students built a simple example of their project, in groups of four. As a second assignment, after studying the elements of the user interface design, the students learned how to use more advanced environments, and were able to create more complex projects.

To actively receive feedback on their work, students were asked to design a colleague experiment, to test potential user's response to the prototype. To encourage interaction and collaboration among students, each group presented their project to other students in class. Each student was asked to share thoughts on other group projects, using an online form. This interactive method of experiencing other group`s work, while presenting your own work, resulted in many positive feedbacks, about the projects and assessment method.

In the theoretical aspect, the Kahoot! Application was used each lecture, to test student knowledge from the previous lecture, and to present the results to all the class in a gamified manner. GoogleDocs were used to perform additional collaborative exploratory assignment during class.
The presentation will include several case studies of courses in the discipline of software engineering, where this method was used, as well as feedback from students, and the interesting results which were obtained from their feedback.

Naomi Unkelos-Shpigel is a PhD student at the IS department, University of Haifa, advised by Dr. Irit Hadar. She lectures at Kinneret College, Braude College and University of Haifa. She has worked for several years in real-time programming and in Information systems design and implementation. Naomi graduated with honor BSc in Information Systems at the faculty of Computer Science, Technion. She graduated her Master's degree at the IS department, University of Haifa.

KINNERET College on the Sea of Galilee

CAPELLA: an Integrated and Adaptive Ecosystem for Life-Long Learning

Amir Tomer and Dorit Alt, Kinneret College on the Sea of Galilee
Miriam I. Barak, Technion, Irit Hadar, University of Haifa
Georg Hagel, HS-Kempten, Germany, Juergen Mottok, OTH-Regensburg, Germany

amir@amirtomer.com

CAPELLA (Cooperative Adaptive Personalized Education for Life-long Learning and Activation) is a proposed framework for an ecosystem that supports learners, teachers and practitioners in life-long learning, throughout academic studies and professional career. CAPELLA is based on the approach that a Body-of-Knowledge (BoK) in a certain area is built-up continuously, and learners are not only knowledge consumers but also knowledge producers.

CAPELLA's pedagogical approach is based on the four pillars of life-long learning, as follows:

- Learning to know: Acquisition of knowledge;
- Learning to do: Acquisition of practical skills for applying the knowledge;
- Learning to communicate and collaborate: Teamwork and knowledge sharing;
- Learning to be: Self-development through an individual learning process.

The Body-of-Knowledge is organized in CAPELLA in 5 "Ex" layers:

- Exposure: A semantic network of concepts, topics and terms comprising the BoK;
- Explanation: Learning units (e.g., lectures, presentations, wiki pages, etc.) that provide the existing knowledge;
- Example: Relevant examples and case-studies that relate to the knowledge units;
- Experience: Real-life applications of the knowledge (e.g., projects, experiments, tools, etc.)
- Examine: Means for evaluating the level of knowledge acquired (e.g. exam questions, quizzes, etc.).

The learning process in CAPELLA is navigation-based and allows the learner to advance through an adaptive learning process both by self-choice and by recommended "routes", based upon other learner's experience and an individual learner-profile, maintained by the system. An important

ingredient in the learning process is Gamification, which turns the learning process into a "quest", with gaming attributes, such as gaining power, achieving goals and getting rewards.

CAPELLA also contains a Community Collaboration Model which supports the creation and evaluation of new knowledge produced by any user and approval of such knowledge for being included in the BoK.

CAPELLA's technical approach is based upon ubiquitous computing ("learn anytime, anywhere"), which is strongly supported by cloud computing and mobile devices.

Prof. Amir Tomer obtained his B.Sc. and M.Sc. in Computer Science from the Technion, and his Ph.D. in Computing from Imperial College, London. Between 1982 and 2009 Amir was employed at RAFAEL as software developer, software manager, systems engineer and Corporate Director of Software and Systems Engineering Processes. Amir is currently affiliated as the head of the Software Engineering department at Kinneret Academic College and as a Senior Teaching Fellow for Software and Systems Engineering at the Technion, Haifa, Israel and other academic institutes.

Dr. Dorit Alt, head of the Education and Community Department, Kinneret Academic College, is specialized in the field of creating and assessing constructivist lifelong learning environments in higher education. She has developed and instructed several academic courses related to the constructivist pedagogy in digital era.

KINNERET College on the Sea of Galilee

KINNERET
College on the Sea of Galilee

KINNERET
College on the Sea of Galilee

KINNERET
College on the Sea of Galilee

kinneret.ac.il | 1800-20-90-20

KINNERET
College on the Sea of Galilee