





Dear friends,

It is a great honor for us to host the first AISTQ Conference (Academia Industry Software Quality & Testing Conference).

The Shamoon College of Engineering is creating a dynamic, new generation of engineers and technology, committed to academic excellence, cutting edge research and development, and innovation.

As Israel's largest engineering college, SCE is inculcating dynamic young professionals with the academic knowledge, and creative vision to meet the demands of today's market, and help develop Israel's southern periphery. Therefore, we offer a challenging academic program based on a unique interdisciplinary approach that ensures that our graduates can immediately contribute to Israel's industry and high tech companies.

After almost two decades of rapid growth, with over 5,000 students in our two campuses in Beer Sheva and in Ashdod, we have seen the vision of our founders turn into a reality. We are committed to continuing to be bastion of excellence while expanding educational opportunities.

Our college emphasizes its relations with other academic institutions, as well as with industry and research facilities. Our faculty members have been developing contacts throughout the world and have established joint ventures in a wide variety of areas.

The theme of the conference signifies our commitment to Quality as part of our educational perception.

I am sure that the conference will lead to stronger ties and closer contacts among the participants and to greater scientific collaboration.

Best Wishes

Prof. Jehuda Haddad
President, SCE-Shamoon College of Engineering

Welcome to AISTQ 2016,

We are delighted to host the first Academia Industry Software Quality & Testing (AISTQ) conference here, in Shamoon College of Engineering. The AISTQ conference brings together researchers and leaders from academia and industry who are interested in contemporary software quality and testing topics. The AISTQ conference provides an opportunity for academia and industry practitioners to meet, present, and discuss novel approaches, methods and techniques for improving software quality.

This year is the first annual conference and its theme is – “Digital Israel” initiative. We believe that quality in the digital environment is a crucial ingredient that requires a broad perspective beyond technology per se, including the focus on the human factors, privacy, security, legislation, and environmental issues that are part of the big picture.

Our Department of Software Engineering and its unique software quality track initiated this conference in order to highlight the related topics of software quality and testing as practiced in academia to the industry and vice versa. We wish to make this platform a home-base for graduate students to present their research, get feedback, and to call for a fertile ground for cooperation between academia and industry.

Enjoy the conference.

All the best,

Dr. Hadas Chassidim
Conference Chair
(*hadasch@sce.ac.il*)

Dear Participants,

The software engineering department at SCE is privileged to host the annual summit in software quality and testing. This conference is designed to bring together researchers from academia and leaders from the industry who are interested in contemporary software testing and quality topics. We in the Department of Software Engineering believe that symbiotic relations between academia and the industry can mutually influence one another. We believe that pursuing quality is not a one-time effort; quality is a foundation stone of software development. It must be injected into the curriculum of every academic activity from day one – "Quality from Day One". This conference was established in order to promote international cooperation for advanced studies and research, home-base and designated track for graduate and post-graduate education, and a source for synergy in research and education between academia and industry in the field of quality and testing.

We are pleased to meet you here today; and hope we will see you again at the second AISTQ Summit.

Prof. Shlomo Mark, Dean
Department of Software Engineering, SCE-Shamoon College of Engineering
(*marks@sce.ac.il*)

Keynote Speakers

Our Journey in the Science of Testing

*Dave J. Miller (djmillertn@gmail.com)
Retired VP IT, FedEx Services*

Research, published papers, business solutions, community involvement – these are key outcomes from establishing strong and vibrant academia-business partnerships. There are fundamental success criteria which lead to these encompassing results, when followed diligently. Dave will talk about the model for successful partnerships, what it takes to make them thrive, and the exciting results which can be achieved when injecting energies into the success criteria for the partnerships. Come learn how you, too, can be a leader in academia-business partnerships.

Cultural Learning of Testing for Make Benefit Glorious Nation of Startup

*Gil Tayar (gilt@wix.com)
Software Architect, Wix*

How would development of software look like if developers stop relying on QA and write the tests themselves? Would it be beneficial to the company? Essential? Developer testing is the ability of the developer to ensure, via automated testing that the developer writes, that the software is (mostly) without bugs. Most developers grumble that their superiors don't allocate time to write automated tests because "their is no time". In the talk, Gil Tayar explains the reasoning behind the methodology that guides Wix developers.

Quality Engineering for the Internet of Things

*Prof. Dr.-Ing. Ina Schieferdecker (ina.schieferdecker@fokus.fraunhofer.de)
Director of Fraunhofer FOKUS, Berlin*

For all economical relevant present topics, e.g. Industry 4.0, Internet of Things, Smart Cities, Smart Grids, etc. linking of technical components and software-based systems is the core concept.

This means new requirements for the quality management as closed systems have to be opened and connected flexibly to systems-of-systems.

Especially for safety-and-security critical systems, the traceable end-to-end quality regarding functionality, interoperability, robustness, reliability and trustworthiness needs to be verified and maintained all the time, which requires also new methods for testing. Some of them are presented and discussed in the talk.

Special Sessions

Where Industry and Academia Come Together

Prof. Robin Poston (rposton@memphis.edu)

Papasan Family Professorship for Exemplary Leadership

Associate Professor and Interim Dept Chair | Business Information & Technology

Director of the Systems Testing Excellence Program

Memphis, TN, USA

Together industry, academia, and government can address the big problems facing the information systems community. At the University of Memphis, we created the Systems Testing Excellence Program (STEP), which has been partnering for over 10 years with our industry and government partners to address the burning issues in testing and quality assurance organizations. As our program continues to stay at the leading edge, we are moving into the cyber security testing area for our research, executive education, and curriculum focus, through the Cluster to Advance cyber Security & Testing (CAST). Robin will talk about the framework for academic and industry partnerships that is needed to fill the gaps of government grant and corporate research & development to ensure today's big problems are solved. Ideas of collaborative success will be shared to position you to lead your own academia-business partnerships.

Accessibility - For Whom?

Virgínia Chalegre (vivichalegre@gmail.com)

Co-Founder and CEO at t-access

As the digital environment becomes increasingly important to lives of people with and "without" disabilities, we must address the concern of accessibility. But how can we make software more accessible for blind people, for example? This question is being posed in a wide range of academic and work environments. This lecture aims to show some researches and practical ways of making accessibility to digital environments more viable.

Quality Driven by Requirements and Design

Improve Quality by Requirements Validation -An Empirical Approach from a Non-Functional Perspective

*Dr. Hadas Chassidim, (hadasch@sce.ac.il)
SCE – Shamoon College of Engineering*

Today, many software products base their business values on the willingness of the user to expose personal information. Usually there is a reason for the users to do so; however, there might also be some barriers, some of which may be related to privacy, human–computer interaction, or other Non-Functional Requirements (NFR) that might be determined to be low priority during the requirements engineering stage. We claim that nowadays high demand development environments might lead to missing NFR due to the Agile development life cycle, which is significantly shorter compared to the traditional methodologies. However, there is no real option to compromise on the standard of product quality. In this talk, I will use two case studies to demonstrate the implications of possible lack in the requirements and the design, particularly from privacy and Human Computer Interaction (HCI) perspectives. We propose a method for evaluating access control choice architectures by measuring how well the default options cover the existing users' choices, using Facebook data of 266 users and compare the optimal to the existing defaults. Outputs of this approach can improve the requirement engineering process by integrating HCI and privacy aspects into the early stages of the product development and as such to reduce costs and increase users' satisfaction.

Design in Test Driven Development Methods

*Dr. Reuven Yagel Azrieli (robi@jce.ac.il)
The Jerusalem College of Engineering*

We will explore the connections between various design and testing practices. We will survey several styles (or schools) of development, starting from basic TDD - Test Driven Development. Then BDD – Behavior Driven Development. We will discuss also the implications of these methods to architecture, design, modeling and other methods. We will discuss the rationale for using those practices, pros and cons, and bring some empirical evidence. If time allocated, all these will be demonstrated along the way with a working example.

PhD Symposium

Search-based Fuzzing with Neural Networks as a Fitness Function

*Martin A. Schneider, System Quality Center - Fraunhofer FOKUS, Berlin, Germany
(martin.schneider@fokus.fraunhofer.de)*

*Ina Schieferdecker - Research Advisor (ina.schieferdecker@fokus.fraunhofer.de)
Technical University of Berlin, Fraunhofer FOKUS, Berlin, Germany*

So far, search-based fuzzing approaches were dedicated to a certain vulnerability and employing handcrafted fitness functions. The idea presented employ a generic search-based fuzzing approach aiming at revealing different classes of vulnerabilities. To achieve this goal, artificial neural networks (ANNs) are employed as a fitness function, taking as inputs the inputs and outputs of a certain SUT as well as non-functional indicators such as CPU load and memory consumption. The purpose of the ANN is to detect dependencies between the inputs and outputs of the SUT and the non-functional indicators in order to assess the degree of possibly vulnerable behavior of the SUT. The infinite input space of input data and message sequences is searched using evolutionary algorithms by mutation, crossover, and selection based on the results of the ANN when the corresponding test case has been executed. Research questions for investigating the approach to be developed are concluding the paper.

A Mathematical Model for Testing Artifacts Reuse

Havatzelet Cohen (havatzelet@sce.ac.il)

Lior Aronshtam (liorar1@sce.ac.il)

SCE-Shamoon College of Engineering

All software must be tested; however this task is very expensive. There are different approaches to reduce the cost of testing. A new approach suggests that since testing is of a repetitive nature, it would be useful to implement a repository of testing artifacts. This repository will be useful only if the users can find the artifacts efficiently. This paper gives a mathematical model for managing the repository. The model allows us to reach units, detect connections, and to assess the quality of units by functions.

Mutation churn model

Michal Mnich (michal.mnich@uj.edu.pl)

Adam Roman (roman@ii.uj.edu.pl)

Piotr Wawrzyniak (piotrek.wawrzyniak@gmail.com)

Jagiellonian University, institute of informatics and computational mathematics, Cracow, Poland

Mutation testing is considered as one of the most effective quality improvement technique by assessing the strength of the actual test suite. If no test is able to kill a given mutant, this means that the tests are not strong enough and we need to write additional one that will be able to kill this mutant. However, mutation testing is very time consuming. In this paper we investigate if it is possible to reduce the scope of the mutation analysis by running it only on the new or changed part of the code. Using data from the real open-source projects we analyze if there is a relation between mutation scope reduction and effectiveness of the mutation analysis.

Ethics as a Quality Indicator within Software Development

Hisham Abdulhalim (Hisham88il@gmail.com)

Dr. Yotam Lurie (yotam@som.bgu.ac.il)

Ben Gurion University of the Negev, Be'er Sheva, Israel

Prof. Shlomo Mark (marks@sce.ac.il)

SCE-Shamoon College of Engineering, Be'er-Sheva

As technology became a key player in our lives, it enhanced leisure time and enriched our experience by increasing the level of information accessibility.

The impacts of these changes have been categorized on the positive side of the bar, leaving an open question of their cost and raising ethical issues. Ethical theories provide ideas and procedures for determining what is ethically relevant. In recent years the spread of technology created new dilemmas, which propose re-evaluating the practice of the information technology profession. When it comes to defining the key performance indicators for measuring quality, there is no solid answer that unifies what all organizations believe and practice in their day-to-day work. The noticeable reasons leading to that include the industry specialty, organization size, and resource allocation strategy. From research a year ago, we believe that professional ethics can assist in leveraging quality definition within the software domain.

Data-Augmented Software Diagnosis

Amir Elmishali (amir9979@gmail.com)

Roni Stern (roni.stern@gmail.com)

Meir Kalech (kalech@bgu.ac.il)

Ben Gurion University of the Negev, Be'er Sheva, Israel

Using machine learning techniques, software fault prediction algorithms predict which software components are likely to contain faults. Software diagnosis algorithms identify the faulty software components that caused a failure using model-based or spectrum-based approaches. We show how software fault prediction algorithms can be used to improve software diagnosis. The resulting data-augmented diagnosis algorithm overcomes key problems in software diagnosis algorithms: ranking diagnoses and distinguishing between diagnoses with high probability and those of low probability. We demonstrate the efficiency of the proposed approach empirically on three open source domains, showing the significant increase in accuracy of diagnosis and efficiency of troubleshooting. These encouraging results suggest broader use of data-driven methods to complement and improve existing model-based methods.

Quick Sessions and Posters

Software Testers Related Factors and their Impact on Patterns of Participation in Professional Forums for Software Testing

Efrat Weinberg (efratwein@gmail.com)

Dr. Paul Gursky, Ph.D- Advisor

Open University of Israel

This research examines connections between Software Testers-related factors and participation patterns in Professional Virtual Forums.

An important aspect of the Software testing role is the On-the-Job-Training – the need for ongoing learning and fast implementation of new innovations.

This research focuses on virtual discussion groups dedicated to professional Software Testing (hereinafter forums). We examined whether there is a connection between the professional characteristics of software testers, their patterns of learning, and their activity in forums. The research tool was an online questionnaire containing 57 multiple choice questions. The questionnaire was posted on forums dedicated to software testing in Israel and abroad.

We got total of 108 responses. Our findings display a number of interesting differences between testing engineers and testing managers in the areas of work characteristics, learning habits, and their attitude to forums. We found few differences between professionals in Israel compared to professionals overseas. We also found that professional testers in niche roles – which require special training, such as automation testing engineers or testing managers, will use forums more often than manual testers, and will implement advice and tips from these forums much more than other testing professionals.

Testing Your Testers - How to Measure Your Testing Team Correctly

Ori Bendet (ori.bendet@hpe.com)

HP

In QA organizations today, a tester must have technical know-how, good communication skills, and be attentive to detail. We know that a tester's main responsibility is to test the software that developers develop to ensure that the product meets the quality standards expected of today's applications. But apart from that, it's difficult to measure what exactly makes a good tester.

QA managers and their team members are constantly under pressure to test faster and more efficiently, and deliver software with fewer defects. The role and importance of QA in today's R&D teams is evolving from simply finding defects to protecting the corporate image. As a result, your testers have to be more productive and more efficient, and change their mindset to think about quality over quantity. It's not just about finding bugs; it's about continuing to measure and improve, and finding the right bugs to make the end-user's experience better.

In this lecture I will share with you some of the key performance indicators (KPIs) that we use to measure our own testing efforts: Percentage of high/critical, escaped defects, Time to test, Defect resolution time, and Percentage of rejected defects, what we've learned from each of them and how our team improved its efficiency and productivity as a result.

Agile and DevOps in a Multi Vendors Business Transformation Project

*Sharon Elgarat (Sharon.elgarat@amdocs.com)
Test Solution Architect, AMDOCS TESTING*

This study presents a methodology for applying Agile in a multi-vendor environment during a business transformation project. The unique challenges faced in this environment force modification of the responsibility balance and the core considerations that define the scope of a marketable feature, based on which the Agile cycles and work load is determined. The article will describe the challenges that led to the methodology and its base principles.

Unit Test: The Unit Behind the Test

*Dov Benyomin Sohacheski (b@kloud.email)
SCE-Shamoon College of Engineering*

Software engineers may have different sentiments in regard to implementation and development methodologies, nonetheless, there is a shared desire for quality outcome. Quality is itself a dialectic process; unit testing has and continues to be a respected and powerful tool in the software engineering (and broad programming) community that assist in achieving higher quality products. Despite its broad adoption in the development process, there is still no consistent or standardized definition for unit tests. Does unit testing directly ensure a higher quality outcome? Could a recognized and consensual definition of unit tests provoke a change in the development process and provide developers with an improved means to scrutinize their code and subsequently enhance product quality? This paper tries to identify the origin of the general understanding of unit tests amongst developers and organizations by examining various acknowledge definitions. This process sheds light on where unit tests should be heading in the future.

Implementing the Slime Mould Intelligence in Software Testing

Aviel Buskila (abialbos@ac.sce.ac.il)

Evgeny Palcev (ibgniplz@ac.sce.ac.il)

Department of Software Engineering, SCE – Shamoon College of Engineering

The slime mold is basically an enormous amount of single cells with thousands of nuclei. They are formed when individual flagellated cells swarm together and fuse. The slime mold has some interesting features; it continuously grows in search for food, it creates a network of sensors around it to get information about food, and it marks the path that it moves in its search for food. In this article we will try to find a connection between the slime mold intelligence of finding food and bug resolution in software testing. The slime mold uses a technique called “spatial” memory to find food. The slime mold is composed of many oscillating units that function as a “sensor” that can alert the other units about new food (good news leading to creating more units in that area) or about light detected (bad news, making the slime decrease its growth there). In its continuous work of finding food the slime mold lives a trail of a thick mat of nonliving, translucent, extracellular slime that will be used in the future to acknowledge which areas it visited.

Many bugs appearing in software testing can be resolved to a “root cause” affecting several bugs that appeared in early testing stages. If we can model early-appearing software bugs as food found by the slime mold, then maybe there is a way to find the root cause as the central part of the slime

Improve Your Unit Tests by Using Chemical Bonds

Aviel Buskila (abialbos@ac.sce.ac.il)

Daniel Cohen (dnialkhn@ac.sce.ac.il)

Evgeny Palcev (ibgniplz@ac.sce.ac.il)

Adir Nahum (adirncom@ac.sce.ac.il)

Department of Software Engineering, SCE – Shamoon College of Engineering

In software engineering, a critical part of the development is to test your product. The lowest level of testing is Unit Test – individual units of source code. One of the main goals of Unit Test is to find if there is a problem at an early stage, before the production stage, but there is a catch: which Unit Test to write and how many?

How do we cover all the depths of code? In order to achieve this goal, we suggest to look at Unit Tests, at the code being tested as a chemical organism made of chemical bonds, which in turn allows us to split and refactor Unit Tests into the logical atom.

Contributors



SHAMMOON COLLEGE OF ENGINEERING

