AIN
SHAMS
UNIVERSITY
Faculty of Computer &
Information Sciences
Information Systems
Department

Effectiveness of Adopting Software Quality Models on the Software Industry in Egypt

Thesis submitted to the Department of Information Systems

Faculty of Computer and Information Sciences
Ain Shams University, Egypt

In fulfillment of the requirements for the Master of Science Degree (MSc) in Computer and Information Sciences

By

Sarah Khaled Abdelrahman Amer

Teaching assistant at the British University in Egypt

Under the Supervision of

Prof. Dr. Ahmed Mohamed Hamad

Professor of Computer Systems,

Faculty of Computer and Information Sciences,

Ain Shams University

Prof. Dr. Nagwa Lotfi Badr

Professor of Information systems,

Dean of Faculty of Computer and Information Sciences

Ain Shams University

Late Prof. Dr. Osman Mohamed Ibrahim

Professor of Software Engineering,

Faculty of Informatics and Computer Science

The British University in Egypt

April 2019

[page left blank]

Copyright

ABSTRACT

Software must undergo quality assurance to meet the criteria for which it was developed with minimal hazards, especially in the critical sectors of a community. Software process improvement (SPI), and by extension Software Quality Assurance (SQA), is the approach to understand the software development process lifecycle and implement necessary changes to the processes to achieve a high-quality, maintainable product. Egypt, along with several MENA-region countries, are in need of locally-focused research on its existing software development organizations in regards to quality appraisal methodologies in use. Egypt's software industry is reliant on small developers, as is the case around the world, and their exports contribute to the global industry, excelling in offshored projects. Local innovative technology remains low in the near-absence of powerful R&D despite quality methodologies in place.

Small software developers face enormous challenges to gain a competitive advantage in the software industry, especially with the presence of large conglomerates. Much of these small-to-medium enterprises (SMEs) adopt agile models such as Scrum to quickly react to clients' demands. Agile methodologies lack proper addressing of maturity in process, project and product that larger enterprises are capable of. It is important for the software engineering community to aid in enabling SMEs to have process maturity without compromising their agility. CMMI – a widely-recognized software quality assurance methodology, is used to address shortcomings in the Scrum model. Specific practices are carefully selected out of eighteen process areas from the second and third levels of maturity to address missing elements. The proposed model is titled SQA-SCRUM, and keeps Scrum completely intact while allowing small developers to produce high-quality software without compromising agility or going over budget, thus reducing the 'low-quality' stigma associated with small software developers around the world.

Acknowledgement

Thank the Creator of the Heavens and the Earth for his grace and mercy upon me.

I would like to thank everyone who helped me along the way to finish this work. You advice was invaluable.

Thank you ***Professor Nagwa Badr****,* ***Professor Osman Ibrahim****, and* ***Professor Ahmed Hamad****. You've all helped me in your own unique ways. Thank you* ***Dr. Walid Helmy*** *for putting me on the right track.*

This work could not have been completed without the help from my colleagues at ***SECC*** and the *Egyptian Ministry of Communication, whom assisted me in obtaining the relevant, useful data required to propel this research forward. Special thanks to* ***Dr. Hossam Osman****,* ***Eng. Ahmed Abdelaziz****, and* ***Mz. Sherine Faragallah.***

Thank you my professional colleagues in Egypt's software industry for your assistance and answering of all my enquiries. Egypt needs more people like you. Special thanks to ***Engs. Hany Gaber****,* ***Eng. Waleed Hassan****,* ***Eng. Amr Khair, Eng. Hanan Elnouby****,* ***Mr. Fady Ealia, Eng. Marian Tadros****, and* ***Eng. Mahmoud Hussein.***

Sincerest thanks go to our helpful colleagues and educators at the Software Engineering Competence Center (SECC) of Egypt and the Ministry of Communication. As for our software quality consultants, thank you for being our advisors and providing us with knowledge of the industry and quality assurance within it. Thanks to all the professionals who helped make data available for study, to further bridge the gap between the academic and industrial sectors in Egypt.

Special thanks to my parents, siblings and family. You are the best, and no words are enough.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Explanation
CM	Configuration Management
CMMI	Capability Maturity Model Integration
DAR	Decision Analysis and Resolution
IPM	Integrated project Management
GG	Generic Goal
GP	Generic Practice
MA	Measurement and Analysis
OPD	Organizational Process Definition
OPF	Organizational Process Focus
OPP	Organizational Process Performance
OT	Organizational Training
PA	Process Area
PI	Product Integration
PMC	Project Monitoring and Control
PP	Project Planning
PPQA	Process and product Quality Assurance
QPM	Quantitative Project Management
RD	Requirements Development
REQM	Requirements Management
RSKM	Risk Management
SAM	Supplier Agreement Management
SCAMPI	Standard CMMI Appraisal Method for Process Improvement
SECC	Software Engineering Competence Center
SG	Specific Goal
SME	Small-to-Medium Enterprise
SP	Specific Practice
SPI	Software Process Improvement
SQA	Software Quality Assurance
SVN	Apache Subversion
TFS	Team Foundation Server
TS	Technical Solution
VAL	Validation
VER	Verification

CHAPTER 1
Introduction

Chapter 1

Introduction

Software applications cover a vast array of industries and purposes, from the entertainment sector to critical health-related services. In addition to local development, offshoring and outsourcing software projects have become commonplace, especially to developing countries. As software becomes integral in humanity's daily life, the quality demanded to maintain it is no longer a luxury, but a mandatory requirement. Till this day, only about 30% of software meet criteria for project success (on budget, on schedule, and fully functional [1]), and billions of dollars are wasted in the form of cancelled or problematic solutions. [2] This is one of the main reasons it became crucial to undergo process improvement in all aspects of software development and its rising complexity – to be able to compete in the international market, and to increase the software industry's contribution to the countries' economies and quality of life.

Low quality software applications can lead to severely underutilized systems, and is an issue that has plagued the software industry since its conception, in both developing and developed countries [3]. In critical sectors, one fault can lead to loss of human life. And yet, quality remains the most elusive of software's characteristics. There are two ways to achieve quality in software – the process-based approach (proactive) and the testing-based approach (reactive). [4]. Software Process Improvement is therefore essential.

Software Process Improvement (also known as **SPI**) is the systematic procedure for improving the performance of an existing process by identifying, analyzing and finding solutions to improve the software development process and by extension, its performance, compliance, and maturity [4, 5]. To make the importance of SPI clear, a software project deployed to a client is the product of the well-managed processes that led to its creation. The end product is influenced by the *quality of the process* used to develop it. [6]

SPI has been constantly improved upon for the last two decades [7,8]. This challenges the notion that Quality Control in software development by itself can solve the issue of wasted resources in software development, which it doesn't, as quality control is a *reactive* process that is only concerned with testing the product for faults after completion. When the focus is on the software development process itself, not just the final stages, it allows problems to be addressed while the software is still in development. Furthermore, SPI's lower development costs, less rework, higher productivity, shorter project cycle time and most crucially, client satisfaction.

The majority of small-to-medium software development enterprises (SMEs), which make up the bulk of the international software industry, do not use software process improvement (SPI) methodologies, instead relying on ad-hoc culture. Software engineering is therefore crucial for a software development organization to manage its projects and provide a means for reducing costs, and establishing a quantitative and qualitative process control for improvement. SPI in turn gives rise to **Software Quality Assurance** (**SQA**) models. These models are designed to *proactively* ensure conformance to requirements and pre-set quality standards. [1, 9,10] Literature in general lacks in developing methods to effectively implement SPI programs in SMEs, with this problem being even more apparent in developing countries. [3]

Small-to-Medium Enterprises (SMEs) must use agile methodologies to survive the intense competition and deploy solutions quickly. Scrum is a popular choice for many, with 85% of agile adaptors embracing Scrum as their preferred method [10, 11]. SMEs make up the bulk of the global software industry and are the most common segment of the industry, making up more than 85% worldwide [9, 10, 11, 12, 13]. Many SMEs make negligible profit from implementing solutions. Rather, their source of income is *maintenance*, servicing the client after the software has already been deployed. This is all the more reason for the software engineering community to aid by introducing process maturity practices without compromising agility.

Currently, most studies on the determinants of process maturity are conducted in medium-or-large scale organizations. The existing data focuses on parts of Europe and North America in particular. Eighty two percent of software companies face difficulty implementing agile methods, and implementing any adjacent form of quality assurance [17]. Most have difficulty following any long-term vision or applying quality models to their projects to compete with their larger competitors and produce high quality software. Therefore, it is necessary to find an alternative solution for these organizations so they can deploy mature software that can exist independently, and guarantee high quality performance and thus returning clients.

Scrum, like other agile models, lends itself to a sprint culture, iterative problem-solving and speedy communication with clients very well [18]. An iteration is a full software development lifecycle consisting of all the phases from planning to acceptance testing. Changes in the customer's needs are discussed during the review meeting, and updates are made to the product backlog. After the review meeting and before the next planning meeting, a retrospective meeting is made to analyse strengths, weaknesses, problems and improvements of the team performance, methods and tools used, and the project itself. The feedback is incorporated into the next sprint [19]. This is to validate the product to the fullest, keep the consumer involved and reduce risks. The output of every sprint is a working product that the product owner, client (and other stakeholders) can test. However, Scrum does not address gaps in the quality of process, project and product that would guarantee a successful deployment of a mature product without a hemorrhage of time, money and resources. It does not curb rework or slippage, and explicit risk management is not addressed. This is the reason much effort by academics and industry professionals goes into incorporating SPI in various forms to accelerate the progress of software development.

Software organizations do not adopt quality assurance models because of the belief that they are documentation-heavy and oriented towards large organizations with plenty of resources [11, 14, 15]. However, SPI is strongly recommended across literature for its