TAMPERE UNIVERSITY OF TECHNOLOGY

Mikko Pohja

Maximizing quality in a small budget software project

Master of Science Thesis

Examiners: Tarkastaja 1
Examiners and topic approved in
the Information Technology
Department Council meeting on
xx.xx.xxxx

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Tietotekniikan koulutusohjelma
Mikko Pohja: Maximizing quality in a small budget software project
Diplomityö, xx sivua, x liitesivua
Xxxxxkuu 201x
Pääaine:
Tarkastajat:
Avainsanat:

Ensimmäinen kappale

Toinen kappale

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Information Technology
AUTHOR : Title
Master of Science Thesis, xx pages, x Appendix pages
xxxxxx 201x
Major:
Examiner:
Keywords:

First paragraph

Second paragraph

PREFACE

Tämä (*d-tyo.tex*) on LaTeX-pohja Tampereen teknillisen yliopiston opinnäytetöitä varten. Samaan pakettiin kuuluu myös tiedosto *tutthesis.cls*, joka sisältää taittoteknisiä lisäyksiä LaTeX:n alkuperäiseen *report.cls*-luokkatiedostoon.

Lisäksi otsikkosivua varten tarvitaan tiedosto *tut-logo.xxx*, jonka tulee sisältää TTY:n logo. Tiedoston tulee olla joko *.eps*- tai *.pdf*-muodossa riippuen LaTeX-versiosta.

CONTENTS

TERMS AND DEFINITIONS

$\hbar$	Redusoitu Planckin vakio
SNR	Signaali-kohinasuhde (engl.: Signal to Noise Ratio)

1. INTRODUCTION

ECO: Economics of software quality ROI: Improving the ROI of software quality–

- Technical debt ECO: Chapter 7

2. SOFTWARE QUALITY

This chapter describes software quality and methods for improving quality in software projects. The first section describes the software quality in general and the motivation behind improving quality. In the second section the most popular active and passive methods used to improve quality, are introduced.

2.1 Software quality in General

2.1.1 Software quality defined

Quality is an attribute of the item in question. It is often described as a combination of qualitative and quantitative attributes. Quality is an ambiguous attribute, because it is subjective and thus differs when viewed from different perspectives. American Society for Quality has two definitions for technical quality: 1. the products ability to satisfy its needs; 2. the products lack of deficiencies [2]

In software business, quality has also multiple viewpoints. Quality has different meanings for vendors, customers, end-users etc.

2.1.2 Motivation

Software is one of the most used type of product in human history. At the same time it has one of the highest failure rates of any type of product mainly due to poor quality. With those facts, it is clear that the total influence of low quality software is considerable in both money and time. Still it is a known fact that a common practise to cut costs is reducing the effort used in software quality. Convincing the payer to allow using effort to achieve good quality can be difficult but crucial. The topic is widely researched and the results speak in behalf of quality.

Phil Crosby has made popular a concept that establishing a quality program will return in savings more than the program costs and thus "quality is free" [VIITE Quality is free]. Even though Crosbys concept is used mainly in manufacturing sector, it has some truth that can apply to software business. In addition, the "cost of quality" is a slightly inappropriate term, >considering that quality in itself doesn't create costs but the lack of it.

Studies have shown that software quality has huge impact on project costs and success. Measurements on 10000 function point projects show that about 31% of projects that size come to an end by cancellation. The average cost of these cancelled projects is about $35,000,000. Succesfull projects the same size with good quality have about 40% lower costs. These figures endorse the effort put to the software quality and make it clear that at least in large scale projects, quality control shouldn't be ignored.

Capers Jones has listed several points that make high quality a major enocomic benefit. In the development of large systems, high quality from the beginning can reduce the propability of cancellations. Software projects can also benefit by achieving shorter development schedules. Shorter schedules with high quality also lowers the costs of a project. Lower development costs, maintenance costs and warranty costs can add up to considerable amounts of cost savings. In addition to the quantitative benefits, high quality raises the satisfaction of customers, end-users and developers.

Jones expresses concerns towards the poor measurement of software quality causing executives and even quality personnel to treat software quality as an expense. Those participants may also treat quality as an issue that is raising the development costs and increasing development schedules. As an equivalent, Jones summarizes the benefits of high quality: "However, from an analysis of about 13,000 software projects between 1973 and today, it is gratifying to observe that high quality levels are invariably associated with shorter-than-average development schedules and lower-than-average development costs" [1]

2.2 Methods for improving quality

- ECO: pretest and tests
- ECO: Chapter 3 preventing defects
- ROI: Three main activities: Review, process audit and testing

JAKO AKTIIVISET/PASSIIVISET??

2.2.1 Testing

- Crash, Smoke and Kattava testaus
- ECO: Chapter 5

2.2.2 Reviews, Inspections and other pretest procedures

- ECO: chapter 4, pretest defect removal

2.2.3 Continuous integration

2.2.4 Lean

3. QUALITY ASSURANCE IN SOFTWARE STARTUP PROJECT

3.1 Startup

3.1.1 Life cycle

- Startup financing cycle (Valley of Death)
- Iteratiivinen kehitys
- Lyhyet sprintit

3.1.2 Scope

- Analytics
- Ominaisuuksien priorisointi (esim. käytön mukaan)
- Scope
- MVP
- Valitaan oikeat feature
- Validoinnit: I Know I When I See It
- Leanista jotain tännekin

3.2 Evaluation of QA methods

- ECO: We use these quality metrics to compare a number of quality improvement techniques at each stage of the software development life cycle and quantify their efficacy using data from real-world applications.

3.2.1 Methods in different phases of the life cycle

3.2.2 Point of diminishing returns

3.2.3 The Don’ts - Things to avoid

- Liittyy vahvasti Leanin Wasteen
- Älä raportoi bugeja, joista tiedät, ettei niitä korjata
- Ei turhia raportteja
- ECO: Cost per defect => paras tulos bugisimmassa projektissa

4. CASE: PÄIKKY

4.1 Quality Assurance in Futurice

4.2 Project objectives

4.3 QA methods and processes used

- Käytetyt QA-metodit
- Kerrotaan prosessista ja tiimistä

4.4 Achieved quality in the project

- Kartoitetaan bugit (Mailit, Pivotal, Repo)
- Asiakkaan ja loppukäyttäjien tyytyväisyys

5. CONCLUSIONS

BIBLIOGRAPHY

[1] C. Jones, O. Bonsignour, and J. Subramanyam. *The Economics of Software Quality*. Addison-Wesley, 2011.

[2] American society for quality. Quality glossary, 2013.