

A statistical analysis of the connection
between test results and field claims
for ECUs in vehicles

B E N J A M I N D A S T M A R D

Master of Science Thesis
Stockholm, Sweden 2013

A statistical analysis of the connection between test results and field claims for ECUs in vehicles

B E N J A M I N D A S T M A R D

Degree Project in Mathematical Statistics (30 ECTS credits)
Degree Programme in Engineering Physics (270 credits)
Royal Institute of Technology year 2013
Supervisor at KTH was Timo Koski
Examiner was Timo Koski

TRITA-MAT-E 2013:08
ISRN-KTH/MAT/E--13/08--SE

Royal Institute of Technology
School of Engineering Sciences

KTH SCI
SE-100 44 Stockholm, Sweden

URL: www.kth.se/sci

Abstract

The objective of this thesis is to analyse the connection between test results and field claims of ECUs (electronic control units) at Scania in order to improve the acceptance criteria and evaluate software testing strategies. The connection is examined through computation of different measures of dependencies such as the Pearson's correlation, Spearman's rank correlation and Kendall's tau. The correlations are computed from test results in different ECU projects and considered in a predictive model based on logistic regression. Numerical results indicate a weak connection between test results and field claims. This is partly due to insufficient number of ECU projects and the lack of traceability of field claims and test results. The main conclusion confirms the present software testing strategy. Continuous software release and testing results in a lower field claim and thus a better product.

Keywords: ECUs, Software testing, Pearsons's correlation, Spearman's rank correlation, Kendall's tau, Logistic regression.

Acknowledgements

First of all, I wish to thank my supervisor at Scania, Jakob Cederblad for his time, help and patience. I wish also to thank my coworker Daniel Iich from Uppsala university and his supervisor Ingemar Kaj for all the insightful ideas and work. Special thanks to all the colleagues at RCIV and other departments at Scania for taking a genuine interest in this thesis.

I would also like to thank my supervisor Timo Koski at KTH for his consistent support, guidance and encouragement throughout this thesis. Last but not least, I thank all my friends and family for their support during the thesis.

Contents

List of Figures

List of Abbreviations

1	Introduction	1
1.1	Background	1
1.2	Objective	1
1.3	Research questions	2
1.4	Methods	2
1.5	Delimitations	2
2	General concepts	3
2.1	Controller Area Network	3
2.2	Electronic control unit	3
3	Software testing	7
3.1	Workflow	7
3.2	Databases	8
3.3	Development process	8
3.4	Validation	9
3.5	Verification	9
3.6	Acceptance criteria	10
3.7	Test cases	10
3.8	Test issues	10
3.9	Testing methods	11
4	Methods	13
4.1	Correlation	13
4.2	Pearson's correlation coefficient	14
4.3	Spearman's rank correlation coefficient	15
4.4	Kendall's rank correlation coefficient	15
4.5	Bootstrap	16
4.6	Logistic regression	17
4.7	Mean time between failures	18

5	Results	19
5.1	Correlation	19
5.2	Regression	24
6	Conclusion	27
6.1	Correlation	27
6.2	Regression	28
7	Future improvement	31
	Appendices	31
A	Results	33
B	Normal probability plots	35
C	Scatter plots	39
D	Correlation tables	43
	Bibliography	47

List of Figures

2.1	Electronic Control Units in a Scania truck	5
2.2	Controller Area Network	5
3.1	Workflow at RCIV	7
3.2	General V-model	8
3.3	A black-box test	11
4.1	Scatter plots of different dependencies	13
4.2	Illustration of homoscedasticity and outlier in scatter plots.	14
5.1	Dependencies of internal variables	21
5.2	Dependency of Mileage	21
5.3	Dependencies of internal & external variables	22
5.4	ICL2: Correlations	23
5.5	ICL2: Project development	24
5.6	ICL2: Interarrival time between failures	25
5.7	Regression results	26
5.8	Prediction results	26
C.1	Scatter plots of internal variables	40
C.2	Scatter plots of internal and external variables	41

List of Abbreviations

AUS	Audio System
BVT	Build Verification Test
CAN	Controller Area Network
DTCO	Digital Tachograph
ECU	Electronic Control Unit
FRAS	Follow-up Report Administration System
GLM	Generalized Linear Model
ICL	Instrument Cluster
LAS	Locking and Alarm System
MTBF	Mean Time Between Failures
SOP	Start Of Production
SWAT	Scania Warranty Administration System

Chapter 1

Introduction

1.1 Background

The automotive industry has in the last decades experienced an enormous transition of traditional mechanical systems into more complex embedded systems. A great challenge for Scania is to cope with the increase of ECUs (electronic control units) and ensure the quality of their products. An important process of quality improvement involves testing, verifying and validating products during software development. The RCIV development group at Scania, where this Master's thesis is carried out, belongs to the section driver interface in cabin development. Their work consists of verification and validation of the cabin's electronic system, which includes the ECUs. RCIV focuses on continuously evaluating field quality problems, verifying new features and simulating different test cases. The research background of this thesis is part of Scania's core values in search of continuous improvements. RCIV is interested in an exploratory study of the test results during product development and field claims in order to set efficient acceptance criteria. Determining a good set of acceptance criteria is crucial for the outcome of the projects and helps project managers to plan, execute and approve the quality of the product.

1.2 Objective

The main objective of this thesis is to analyse the connection between test results and field claims of ECUs in order to improve the acceptance criteria and evaluate software testing strategies. The purpose is to compile test results from ECU projects and investigate how well the predetermined set of acceptance criteria for them are met. The aim is finally to examine the causality of field claims and evaluate its connections to test results, where significant connections will be considered to make predictions on the outcome of new projects based on test results in previous ones.

1.3 Research questions

- What are the connections between test results and field claims?
- What is the impact on field claims with continuous product releases?
- How much software testing is required during product development to ensure a satisfactory quality?
- What is the impact on field claims with a mileage as an acceptance criteria?

1.4 Methods

The strength of the connections between test results and field claims is quantified by computing different measures of dependencies such as linear and rank correlations. The connections are visualized through scatter plots, and a resampling method called non-parametric bootstrap is used to ensure the significance of the correlations and presented via histograms. Results of the correlations are furthermore implemented in a logistic regression model to make predictions with a certain confidence interval on future field claims. MTBF (Mean time between failures) is also computed as a part of a reliability analysis to monitor project development.

1.5 Delimitations

The research is based on final project reports on only four different ECUs with various product releases. This is due to lack of documentation in other ECU projects, which makes it difficult to compile information uniformly in order to make any statistical analysis. This research considers only ECU projects which have been in the market up to three years due to poor documentation in older ECU projects. The field claims reported are only considered during the first warranty year which is the standard measure of field quality at Scania. The software testing of the subcontractors which provide Scania with the ECUs is not considered.

Chapter 2

General concepts

This chapter introduces CAN (Controller Area Network) which is an important network in the automotive industry. We will also take a closer look at several kinds of ECUs in a Scania vehicle which are part of the software testing process.

2.1 Controller Area Network

CAN is a message based protocol for network communication between electronic control units. It is a vehicle bus standard used in Scania vehicles and the automotive industry [1]. The CAN on Scania's vehicles is presented in figure 2.2. The network is divided into three CAN-buses; red, yellow and green with several ECUs connected to each bus organized according to the severity of the system [2]. The most critical systems are in the red bus consisting of systems for the driveline; engine, gearbox and brakes. The yellow bus is responsible for driver safety systems; instrument panel, lights and tachograph. The green bus is in charge of driver comfort systems; heating and climate control. All the buses are connected together through the coordinator (COO) ECU which acts as a gateway between the different buses.

2.2 Electronic control unit

An ECU consists of several physical control units such as sensors, actuators and switches which are connected together on a CAN bus network [1]. A modern vehicle has a dozen of ECUs with different functionality who all read information from sensors, receives messages from other ECUs and regulates actuators and switches to control the vehicle [3]. Figure 2.1 shows the complete ECU set of a Scania truck. Four different ECUs are part of this research and described on the next page.

Instrument Cluster (ICL)

ICL is the control unit for the instrument cluster and is the primary data source. It delivers information about the vehicle such as the engine status and parking brake warning system, among other necessary information [4]. ICL communicates with several ECU systems and is responsible for the driver's interaction with the vehicle. In this research, the second generation ICL (ICL2) with different versions depending on different SOP (Start of Production) releases are included.

Digital Tachograph (DTCO)

DTCO is the control unit for the digital tachograph. A tachograph records the driver's speed, distance and is legally required in European trucks by EU [4]. This research includes two different brands of tachograph, Continental and Stoneridge. There are two different versions of each brand, Continental tachograph 1.3 and 1.4, Stoneridge tachograph 7.1 and 7.3. The differences between the versions are updates of products with different SOP releases.

Audio System (AUS)

AUS is the control unit for the audio system of the vehicle. The second generation of AUS, (AUS2) are included in this research with three different versions; Medium, Medium with Bluetooth and Navigation.

Locking and Alarm System (LAS)

LAS is the control unit for the locking and alarm system of the vehicle. It has three different features, easily customized for the customer's requirements [4]. The most basic one is Remote Central Locking System. We have then Remote Central Locking System with Alarm System and Speed Locking System as additional features. Speed Locking System is legislated in Brazil to ensure the safety of the driver by locking doors as soon as the vehicle starts moving [4].

2.2. ELECTRONIC CONTROL UNIT

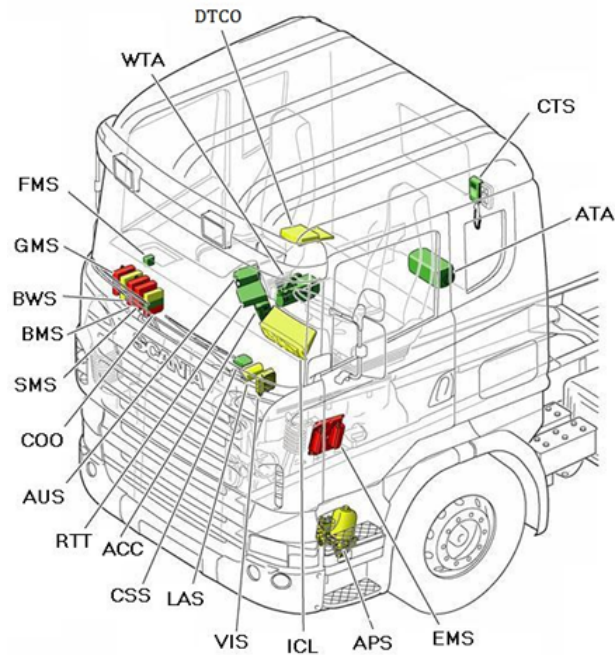


Figure 2.1. Electronic Control Units in a Scania truck

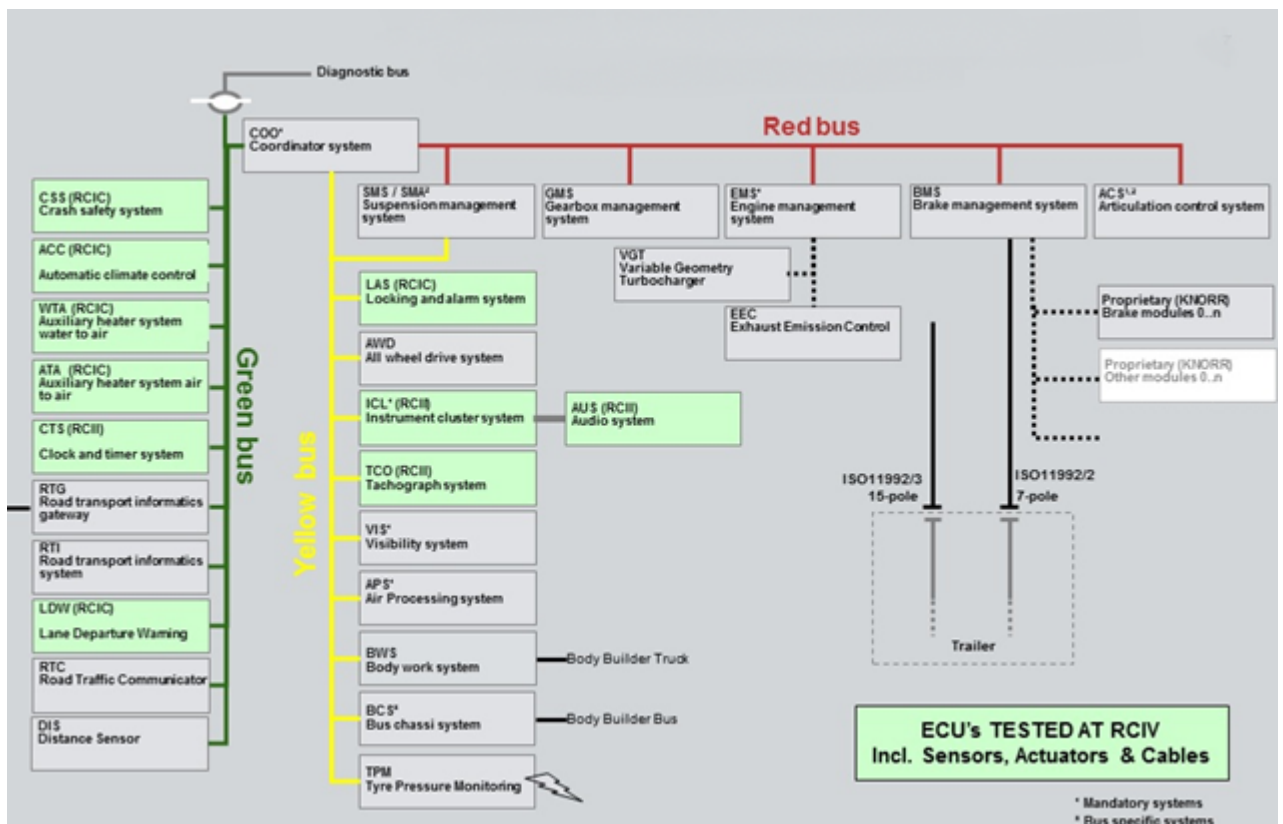


Figure 2.2. Controller Area Network

Chapter 3

Software testing

This following chapter presents general concepts, methods and terminology in software testing. We begin with an introduction of the workflow at RCIV and the databases used for compiling test results from ECU projects and their field claims. We continue by explaining the software development process represented by the V-model and defining some software testing terminology. For more information on software testing, see [11] and [12].

3.1 Workflow

The present workflow of RCIV's test process is illustrated in figure 3.1. The product development starts with a prestudy as part of the planning phase. It includes working through requirements, specifications, acceptance criteria and preparing test plans. The second phase, Analysis & design, focuses on writing test plans that include test specifications, priority list for all test cases and field test. This phase also includes allocation of tools and resources to begin bench test and prepare regression test. The third phase, Implementation & Execution, involves executing test cases, analyse and report results, in other words verification of the software. The last phase, Evaluating & reporting, results in several reports, among them a final report of the entire test process.

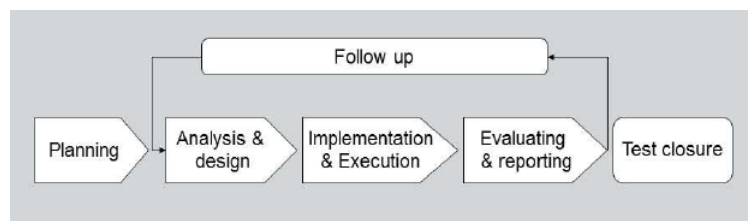


Figure 3.1. Workflow at RCIV

3.2 Databases

Test issues during software development are reported in Mantis and Jira which are Scania's bug tracking systems [5]. A bug tracking system is a software application that is designed to help quality assurance and programmers to keep track of reported software bugs in their work [6].

ECU projects have weekly reports where the development of the test cases and the tests results are summarized into status reports. Final project reports are written in the end of each project which summarizes the outcome of the ECU project and is the primary source for project managers to approve the ECU product for release in the market. Final project reports are based on information from the bug tracking system, and additional test results from field test.

Field claims are reported in SWAT (Scania's Warranty Administration System) and FRAS (Follow-up Report Administration System) [7]. SWAT summarizes the claims from customers and sorts them by different categories and has been the primary source of warranty statistics. FRAS is a database where one can examine specific articles and causality of claims with detailed information.

3.3 Development process

The V-model is the present model at Scania used in system engineering to represent product development [8]. There are two main phases, the verification and development phase on the left hand side and the validation and testing phase on the right hand side, shown in figure 3.2 [9]. There are specific requirements in each step of the validation phase which has to be fulfilled in the verification phase. Software validation and verification are explained on the next page.

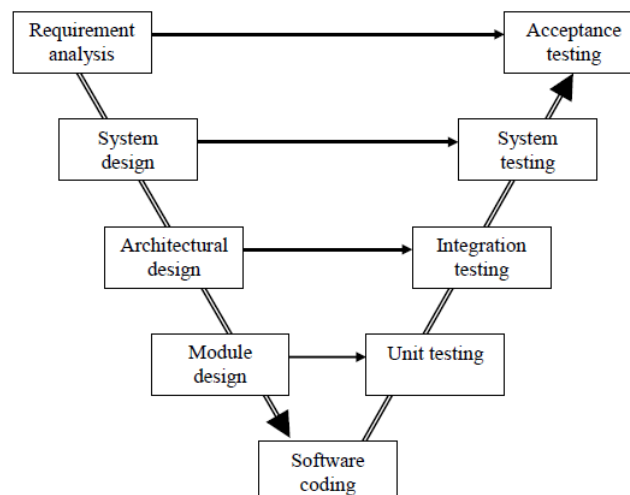


Figure 3.2. General V-model

3.4. VALIDATION

3.4 Validation

Software validation is a process of evaluating software during or the end of the development phase in order to determine whether it meets the requirements. It is corrective approach to ensure software tractability according to the customer's requirements. The following question summarizes the objective: Are we building the right product? [10]. The validation steps in the V-model are specified below.

Unit testing

The unit testing is performed by independent testers on a small component or module of software (ECU). The main purpose is to ensure the functionality is according to the requirement specification. This phase is the most cost efficient in finding defects in the data, algorithm or specification compared to other phases during development.

Integration testing

The integration testing is performed on a group of units. The main purpose is to test the interface and communications between the units such that the architecture design requirements are fully satisfied.

System testing

The system testing is performed both on the interface between the units and the hardware. The main purpose is to verify that all system elements have been fully integrated. It is performed according to a test plan which includes test cases and test procedures.

Acceptance testing

The acceptance testing is usually performed by the customers. The main purpose is to ensure customer requirements, usability and satisfaction.

3.5 Verification

Software verification is a process of evaluating software and assuring that it satisfies the requirement imposed at the beginning of the testing phase. It is a preventive method to ensure software quality and functionality. The following question summarizes the objective: Are we building the product right? [10].

3.6 Acceptance criteria

An acceptance criteria are defined as one criteria set which test leaders determine before testing. The main purpose is to facilitate the approval of a new product. An example of an acceptance criteria at Scania is illustrated with fictitious data in Table 3.1. The requirements are set before testing, and the results are compiled in the end of the project.

Acceptance criteria	Requirement	Result
Critical bugs	0	2
Mileage[km]	900000	650000
Hours bench[h]	500	700
Driving sessions[h]	300	250
Test coverage	100%	99%
Number of test iterations	7	4

Table 3.1. Acceptance criteria

3.7 Test cases

A test case may consist of a single or several tests which are usually specified at the beginning of each project. Test cases cover test requirements specified from a subcontractor or test leaders. Most test cases are created ahead of testing while some are written subsequently when new issues are found during exploratory testing or during operation of a system. Writing test cases subsequently are crucial in order to recreate errors to ensure new test issues are fixed.

An important set of test cases in Scania is executed in the vehicle under different weather conditions, in order to assure the functionality. These test cases are called field tests since they are executed in the field by different development groups at Scania. There are also a subset of field tests LP (Långtidsprov), where new ECU are installed in vehicles and monitored to verify and validate new features before introducing the product in the market. These field tests are performed by Scania's customers.

3.8 Test issues

Test issues discussed in this thesis are defined as problems and bugs that occur during testing. It can be either failures in the test cases or errors that occur when operating a system. The issues are also categorized based on severity which could be minor, medium or critical. The ambition is to solve all critical issues prior to SOP.

3.9 Testing methods

Black-box test

Black-box testing is a method of software testing which main objective is to test the functionality of an application by only evaluating the input and output. The method is applicable during all software testing phases, unit, integration, system and acceptance testing. Visualization of the term black-box testing can be seen in Figure 3.3

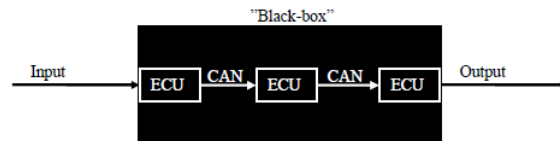


Figure 3.3. A black-box test

Exploratory test

Exploratory testing is considered to be a black-box testing technique coined by Cem Kaner [13] and described as a style of software testing that emphasizes the personal freedom and responsibility of the individual tester to continually optimize the value of her work. It is a software testing approach which combines experience and creativity of the testers and simultaneous learning to generate effective tests.

White-box test

White-box testing focuses on internal structures of an application which are considered in the design of test cases where the testers choose inputs to exercise logical paths through the code and determine the appropriate outputs.

Smoke test

Smoke testing is a preliminary for further testing and a quick assessment of the quality of the software in order to reveal simple failures severe enough to reject a prospective software release. The method is conducted and approved at Scania during software integration phase in order to proceed with new set of test and ensure the new setup is safe for vehicle testing in traffic. Smoke test is also called Build Verification Test (BVT).

Regression test

Regression testing is conducted after an introduction of new functions or fixing a bug in a system. It tests all unchanged functions to secure that the new implementation or bug fix has not introduced new bugs into the system.

Chapter 4

Methods

This chapter introduces three measures of dependencies, Pearson's correlation, Spearman's rank and Kendall's tau, in order to quantify the connection between test results and field claims. The connections are presented graphically through scatter plots and their accuracy due to small sample sizes of test results is verified by a resampling method called non-parametric bootstrap. Significant connections are furthermore considered in a predictive model via logistic regression and a reliability term, MTBF is computed. For further reading about dependencies and complete derivations and proofs, see [14], [15], [16] and [17].

4.1 Correlation

Scatter plot

A scatter plot is a simple tool for identifying relationship between two variables X and Y . It is one of the seven basic tools for quality control which is useful when examining dependency and non-linear relationships for a set of data [18]. Different scatter plots are illustrated in figure 4.1 and the linear relationship can be quantified by using Pearson's correlation under certain assumptions.

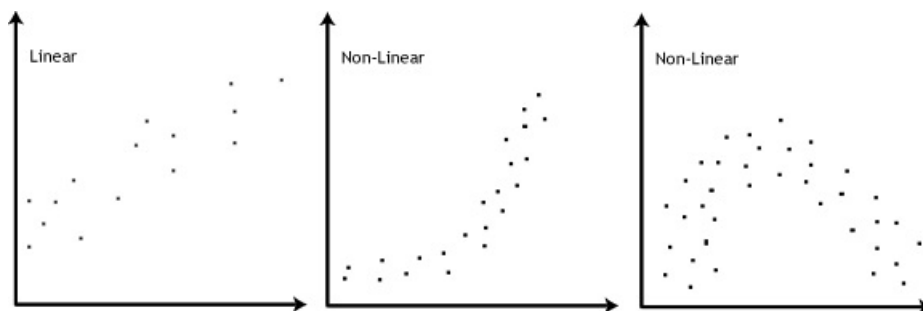


Figure 4.1. Scatter plots of different dependencies

4.2 Pearson's correlation coefficient

The covariance of two random variables X and Y is defined as

$$Cov(X, Y) = E[(X - E(X))(Y - E(Y))] = E(XY) - E(X)E(Y) \quad (4.1)$$

We can standardize it by dividing it by the standard deviation of each variable involved. This results in a coefficient called Pearson's correlation coefficient, which is the most widely known measure of dependency since it can be easily calculated by definition 4.2 for data population and equation 4.3 for sample data, where \bar{X} and \bar{Y} are the averages of X respectively Y variable.

$$\rho_{X,Y} = \frac{Cov(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (4.2)$$

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (4.3)$$

Pearson's correlation coefficient is a measure of linear dependence between two variables X and Y . The coefficient ρ has a range between $-1 \leq \rho \leq +1$ for the true population. Perfect positive or negative linear coefficient equals to ± 1 which corresponds to data sample point lying exactly on a line. Pearson's correlation has following properties and makes these assumptions about the variables X and Y .

- X and Y show homoscedasticity which can be observed in figure 4.2. The variances along the line of best fit remain similar as you move along the line. In other words, it is invariant under strictly increasing transformation. $\rho(aX + b, Y) = \rho(X, Y)$ if $a > 0$ and $\rho(aX + b, Y) = -\rho(X, Y)$ if $a < 0$
- If X and Y are independent, this implies $\rho(X, Y) = 0$ but not the converse.

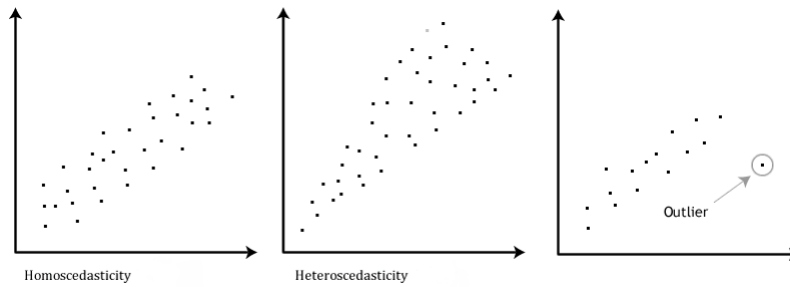


Figure 4.2. Illustration of homoscedasticity and outlier in scatter plots.

Pearson's correlation is greatly influenced by outliers, unequal variances and restricted by normality and linearity conditions. This makes it interesting to consider the Spearman's rank correlation coefficient.

4.3. SPEARMAN'S RANK CORRELATION COEFFICIENT

4.3 Spearman's rank correlation coefficient

Spearman's rank correlation ρ_S is defined as

$$\rho_S(X, Y) = \rho(F_X(X), F_Y(Y)) \quad (4.4)$$

for X and Y with cumulative distribution F_X and F_Y [17]. Spearman's rank is computed by taking Pearson's correlation coefficient on the ranks of sample data. We assign ranks to a given sample data $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ where r_i =rank of x_i and s_i =rank of y_i where $\bar{r}=\bar{s}=\frac{n+1}{2}$. Inserted in equation 4.3 results in:

$$\rho_S = \frac{\sum_{i=1}^n (r_i - \bar{r})(s_i - \bar{s})}{\sqrt{\sum_{i=1}^n (r_i - \bar{r})^2} \sqrt{\sum_{i=1}^n (s_i - \bar{s})^2}} = 1 - \frac{6}{n(n^2 - 1) \sum_{i=1}^n (r_i - s_i)^2} \quad (4.5)$$

Spearman's rank is a non-parametric measure of the strength of association where $\rho_S = \pm 1$, if there is a perfect agreement or complete disagreement between the two sets of ranks. Spearman's rank has the following properties and makes theses assumptions about the variables X and Y .

- X and Y are not required to follow a normal distribution.
- X and Y must satisfy a monotonic relationship which means that the variables either increase in value together or as one variable value increases the other variable value decreases. In other words, it is invariant under non-linear strictly increasing transformations.

The correlation is not as sensitive to outliers as the Pearson's correlation though not perfect either since zero rank is not equivalent with independent random variables.

4.4 Kendall's rank correlation coefficient

The Kendall's correlation coefficient τ is also a non-parametric measure of association and very similar to Spearman's rank correlation coefficient. We let a set of observations $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ be a joint random variables X and Y respectively such that all the values of x_i and y_i are unique. The pair of observations (x_i, y_i) and (x_j, y_j) are defined concordant if the ranks for both elements agree which means if both $x_i > x_j$ and $y_i > y_j$ or if both $x_i < x_j$ and $y_i < y_j$. They are defined discordant, if $x_i > x_j$ and $y_i < y_j$ or if $x_i < x_j$ and $y_i > y_j$. They are defined as neither concordant nor discordant if the pairs are $x_i = x_j$ or $y_i = y_j$. The correlation has a range of $-1 \leq \tau \leq +1$ that can be computed by equation 4.6.

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\frac{1}{2}n(n-1)} \quad (4.6)$$

It is not as sensitive to outliers as the Spearman's rank but has similar assumptions and properties.

4.5 Bootstrap

Bootstrap is a statistical resampling method for assigning measures of accuracy to sample estimates. The main objective is to estimate properties of an estimator such as the correlation coefficient in this research by sampling from an approximation distribution e.g the empirical distribution of the observed data.

The non-parametric bootstrap estimates a parameter of a population or probability distribution without assuming any parametric distribution which makes it a robust estimator. The observations can be assumed to come from an i.i.d (independent and identically distributed) population. Non parametric bootstrap creates new samples by random sampling with replacement from the original dataset where different measures of dependencies are calculated based on the resampled data.

Definition

The non-parametric bootstrap estimates the true distribution F defined [19] below:

$F =$ empirical distribution for our sample observation with the probability mass function which assigns $1/n$ in each of the n sample observations (x_1, x_2, \dots, x_n)

Non-parametric bootstrap procedure:

- Collect the data set of n samples (x_1, x_2, \dots, x_n) from different internal and external variables defined in chapter 5.
- Create N Bootstrap samples (x_1, x_2, \dots, x_n) where each x_i is a random sample with replacement from (x_1, x_2, \dots, x_n)
- For each Bootstrap replicate (x_1, x_2, \dots, x_n) calculate different measures of dependencies ρ . The distribution of these estimates of ρ represents the Bootstrap estimate of uncertainty about the true value of ρ .

The procedure is repeated N times and the empirical distribution of the resampled ρ values are used to approximate the sampling distribution of the statistic. A 95 % confidence interval for the estimates can be defined as the interval spanning from the 2.5th to the 97.5th percentile of the resampled estimate values. For more information on resampling methods and implementation in Matlab, see [19] and [20].

4.6 Logistic regression

Logistic regression belongs to a family of regression models, called generalized linear models (GLM). It provides a unified approach to model different sorts of response variables which are not necessarily quantitative or normally distributed. GLM is a generalization of general linear models, seen in equation 4.7 with a linear predictor part denoted in 4.8.

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \quad (4.7)$$

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} \quad (4.8)$$

GLM generalizes the general linear models through relaxation of assumptions on normality, linearity and homoscedasticity conditions. This is accomplished by introducing an arbitrary distribution which belongs to the exponential family of distributions such as Poisson, Normal and binomial distributions, among others. GLM models the expected value of the response variable through equation 4.9, where $g(\cdot)$ is called a link function and has the general form, denoted in 4.10.

$$\mathbf{E}(\mathbf{Y}) = \boldsymbol{\mu} = g^{-1}(\mathbf{X}\boldsymbol{\beta}) \quad (4.9)$$

$$g(\boldsymbol{\mu}) = \boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} \quad (4.10)$$

GLM consist of three specifications [21].

1. The distribution
2. The link function $g(\cdot)$
3. The linear predictor $\mathbf{X}\boldsymbol{\beta}$

In this research, a binomial distribution is applied since the response variable is a binomial proportion. We wish to model probabilities as functions of the linear predictor, defined in chapter 5 as Claim ratio respectively Fail ratio, where y = number of warranty claims out of n produced units, denoted in the binomial distribution below:

$$f(y) = \binom{n}{y} p^y (1-p)^{n-y} \quad (4.11)$$

$$p_i = h(\eta_i) \quad (4.12)$$

An invertible function $h(\cdot)$ is called the inverse link function which restricts the proportion to the interval $[0,1]$. The natural choice for link function in a binomial model is the logit link defined below:

$$g(p) = \text{logit}(p) = \log\left(\frac{p}{1-p}\right) \quad (4.13)$$

$$h(\eta) = g^{-1}(\eta) = \frac{e^\eta}{e^\eta + 1} = \frac{1}{1 + e^{-\eta}} \quad (4.14)$$

4.7 Mean time between failures

MTBF (Mean time between failures) is a reliability term which can be calculated as the arithmetic mean time between failures, for typically a repairable system. It is a common term in reliability growth modelling to present a graphical visualization of the reliability growth during the development phase [22]. MTBF can be defined as:

$$\int_0^{\infty} t f(t) dt$$

where $f(t)$ is the density function of time until failure defined below:

$$\int_0^{\infty} f(t) dt = 1$$

In this research, we have defined the MTBF as the inverse of Fail ratio defined in chapter 5. The application of MTBF on software issues is not generally accepted by the entire testing community, who argue it is initially adapted for hardware failures, which wear out in time and follows a Poisson distribution, implemented in reliability growth models [23].

More applications on MTBF and reliability growth models can be found in [24]. Additional information on GLM and logistic regression can be found in [21] and [25].

Chapter 5

Results

This following chapter presents findings on the connection between test results and field claims which are defined as internal respectively external variables. The main focus lies in finding correlation between Fail and Claim ratio. We consider first, all different ECU projects in the computations, in the correlation section. We further examine only ICL2 projects separately in a correlation and regression analysis due to insufficient sample sizes in other projects. All computations are made in Matlab.

5.1 Correlation

The outcomes of all projects are compiled in tables A.1 and A.2 in appendix A. The tables are sorted in two main categories of variables: 1) Internal variables, based on the set of acceptance criteria and test results where the outcomes can be considered as results of the projects during product development. 2) External variables, based on the field claims statistics which can be considered as a measure of product quality and testing efficiency. The current set of acceptance criteria are the following variables X_4 , X_6 , and X_7 and the rest of the variables are outcomes of test results. Certain acceptance criteria such as test coverage are poorly documented and difficult to quantify and thus not considered. The internal and external variables are presented and defined in table 5.2 respectively table 5.1.

External variable	Definition
X_{11} =Chassis	Number of produced chassis (units)
X_{12} =Claims	Number of warranty claims
X_{13} =Cost	Total cost of all claims
X_{14} =Cost ratio	Defined as: X_{13}/X_{12}
X_{15} =Claim ratio	Defined as: X_{12}/X_{11}

Table 5.1. External variables

Internal variable	Definition
X_1 =Issues	Number of total new and old issues found
X_2 =Closed	Number of solved issues
X_3 =Open	Number issues which are known before product release
X_4 =Critical	Number of critical issues which may cause system failure
X_5 =Test cases	Number of test cases executed. It is a descriptive variable for the complexity of the system and the requirement specification
X_6 =Hours	Number of hours bench testing and writing test cases for the requirement specification
X_7 =Mileage	Number of kilometers driven
X_8 =Iterations	Number of test iterations executed
X_9 =Days	Number of days worked during project
X_{10} =Fail ratio	Defined as: X_1/X_9

Table 5.2. Internal variables

Correlation results

Part I

The preliminary analysis and results are focused on the internal variables in order to investigate the correlations between Fail ratio and rest of the internal variables, especially the acceptance variables. Correlation table D.1 in appendix D shows the computed correlation coefficients. The coefficients are bootstrapped due to small sample sizes and a 95% CI (confidence interval) is constructed to ensure the significance of the coefficient which can be seen in table D.2 in appendix D.

Results show no strong correlation between the variables in general with exception of correlation between Fail ratio and the variables Open and Hours. They are illustrated by scatter plots and histograms of the correlation coefficients in figure 5.1. The complete scatter plots of Fail ratio and the internal variables are presented in the appendix C. The acceptance variables besides Hours show no strong sign of correlation eg. Mileage variable can be seen in the figure 5.2.

5.1. CORRELATION

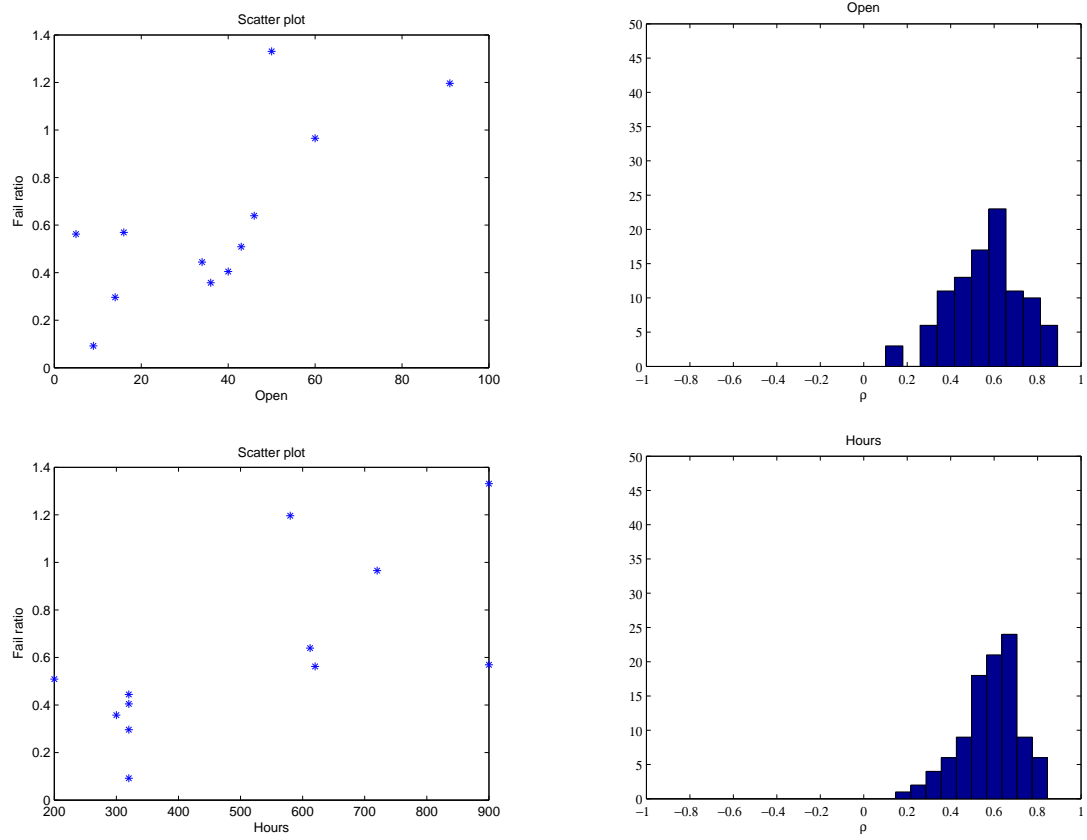


Figure 5.1. Dependencies of internal variables

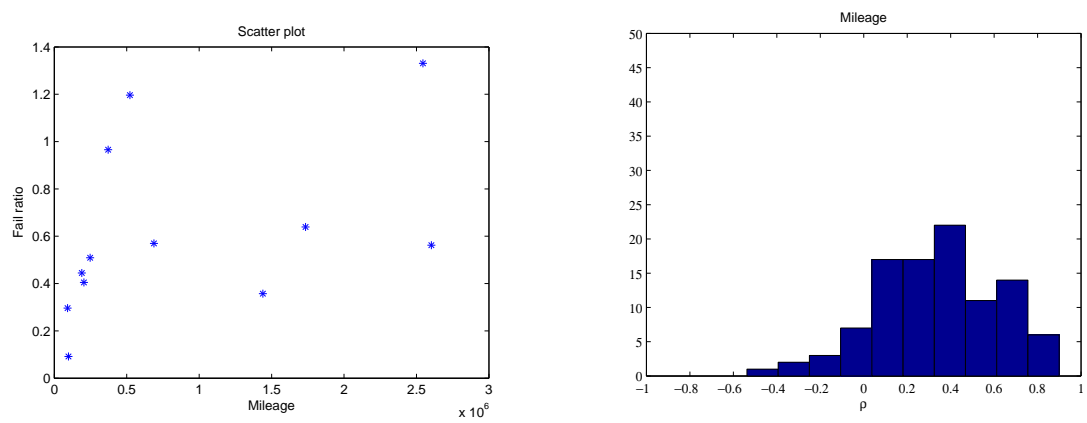


Figure 5.2. Dependency of Mileage

Part II

We investigate furthermore the relationship between internal and external variables. The main focus is to calculate correlations between Claim and Fail ratio which is the most interesting connection of this research. Correlation table D.5 and bootstrap table D.3 in appendix D shows the complete different associations with significant correlations. Project Y_8 is considered an outlier and removed since it has a substantially higher Claim ratio, seen in table A.2

Results indicate no significant correlations between Claim and Fail ratio. There seems only to be a connection between Claim and Cost ratio. Histograms based on 95% CI and scatter plots illustrates their dependencies in figure 5.3. The complete scatter plots of Claim ratio and the internal and external are presented in the appendix C.

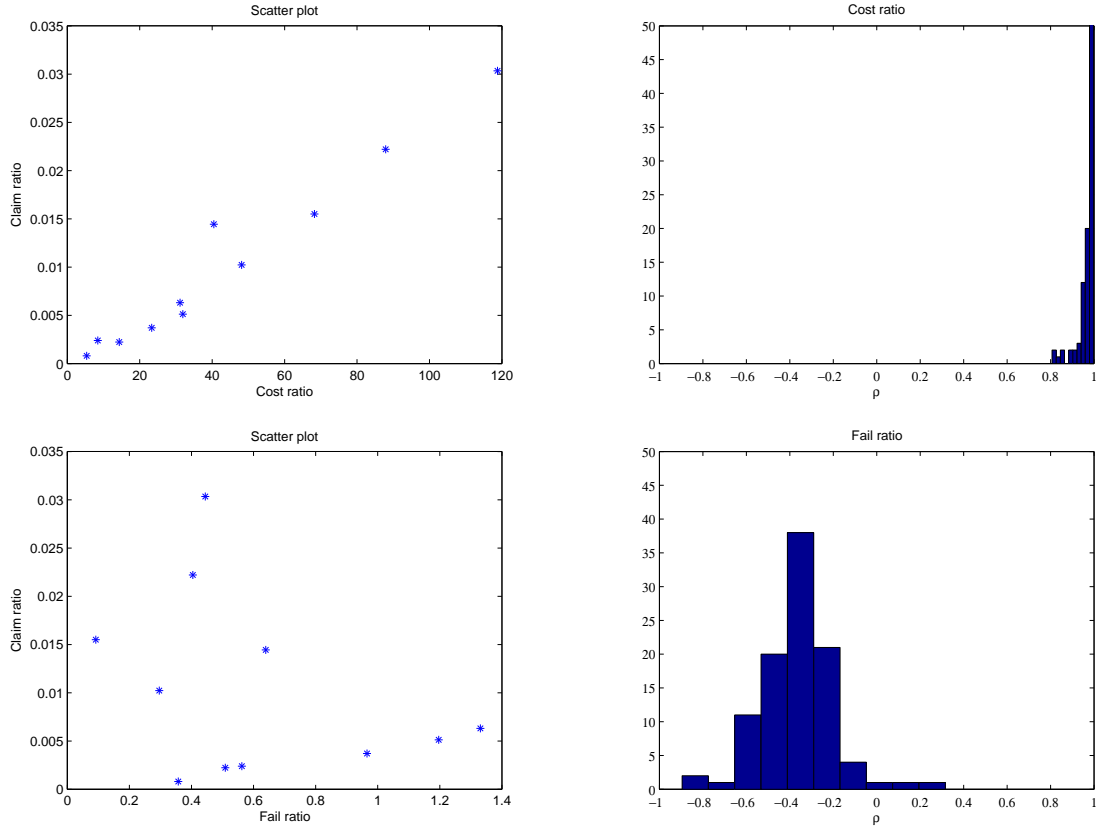


Figure 5.3. Dependencies of internal & external variables

5.1. CORRELATION

Part III

ICL2 projects are considered due to a larger sample size. Bootstrap table D.4 in the appendix shows the correlations between internal and external variables. A significant correlation exists between Claim and Fail ratio. An increasing Fail ratio results in a higher Claim ratio illustrated in figure 5.4 together with insignificant correlation between Claim ratio and Mileage.

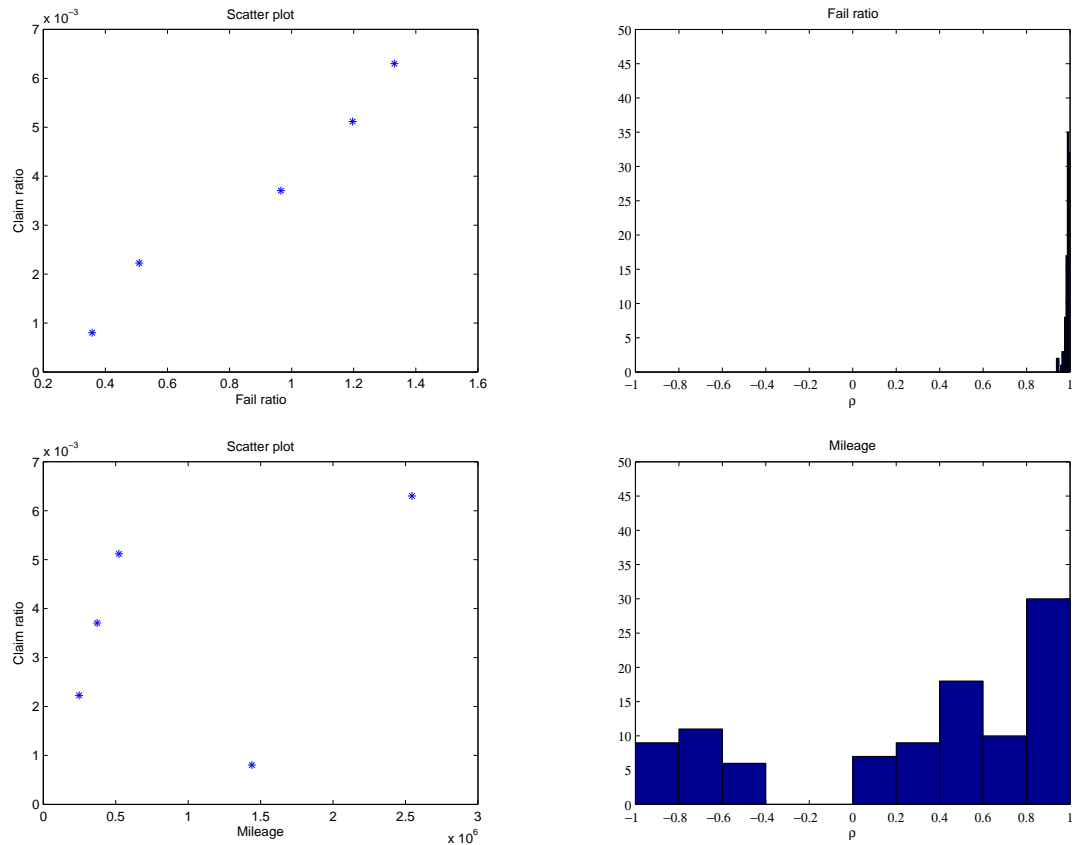


Figure 5.4. ICL2: Correlations

5.2 Regression

The regression is based on the Fail and Claim ratio for ICL2's five recent releases. Results of four of the previous releases are illustrated in figure 5.5. The slope of the curves decreases for each release which could be a sign of an improved product. The computed MTBF and interarrival time between failures, increases for each ICL2 release and confirms the improvement of the ECU, which can be observed in figure 5.6 and table 5.3

The linear correlation coefficient is computed to 0.971, and a simple regression is performed to model the dependency of Fail and Claim ratio. The simple regression graph is improved by applying a logistic regression, seen in figure 5.7. The goodness of fit is graphically presented by a normal probability plot of the Pearson's residuals. The residuals are normalized in order to have a standard normal distribution when the model is a reasonable fit to the data. The normal plot shows signs of heavy tails and does not follow a perfect normal distribution as can be seen in figure 5.8. We can validate the logistic model by a prediction plot of the expected number of claims out of 1000 produces chassis that would fail during software development. This can also be observed in figure 5.8 with a 95 %CI to ensure the accuracy of the model.

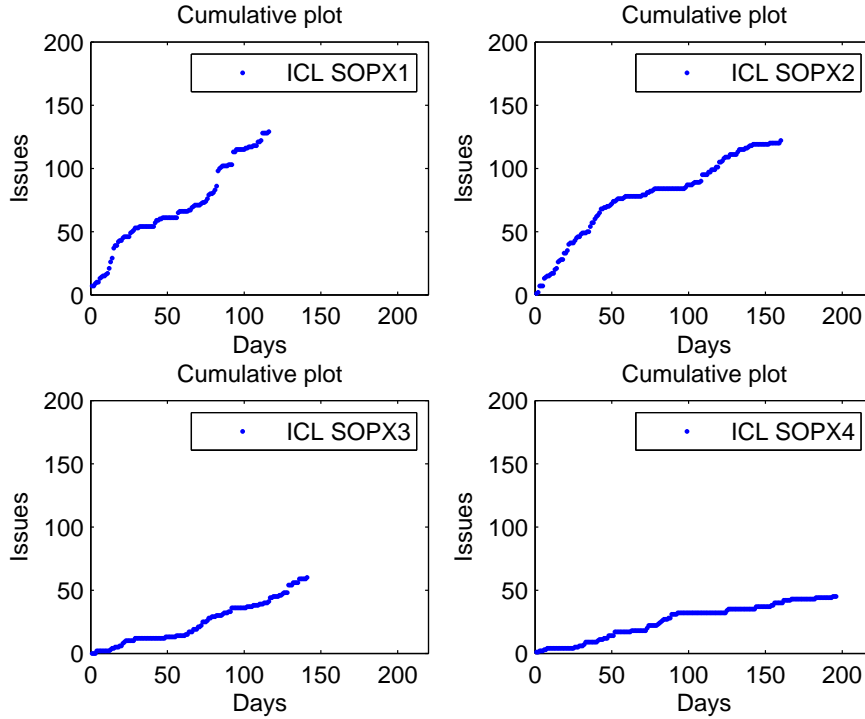


Figure 5.5. ICL2: Project development

5.2. REGRESSION

Project	MTBF
ICL SOPX1	0.7519
ICL SOPX2	0.8403
ICL SOPX3	1.0417
ICL SOPX4	1.9608

Table 5.3. MTBF results

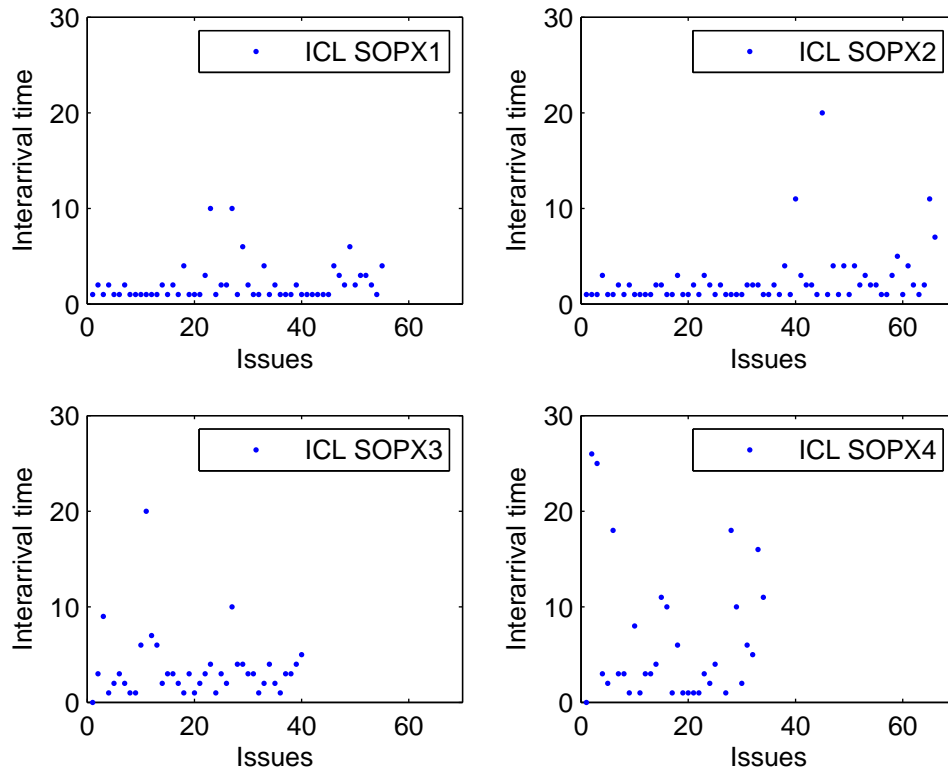


Figure 5.6. ICL2: Interarrival time between failures

CHAPTER 5. RESULTS

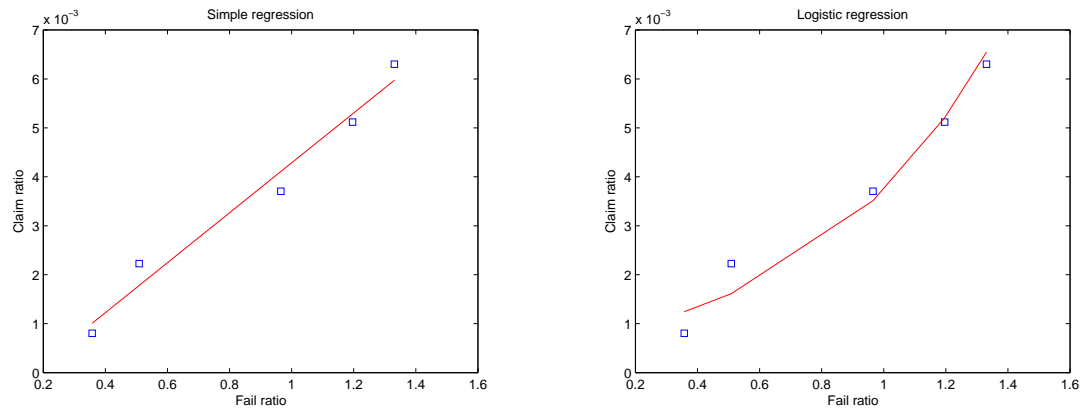


Figure 5.7. Regression results

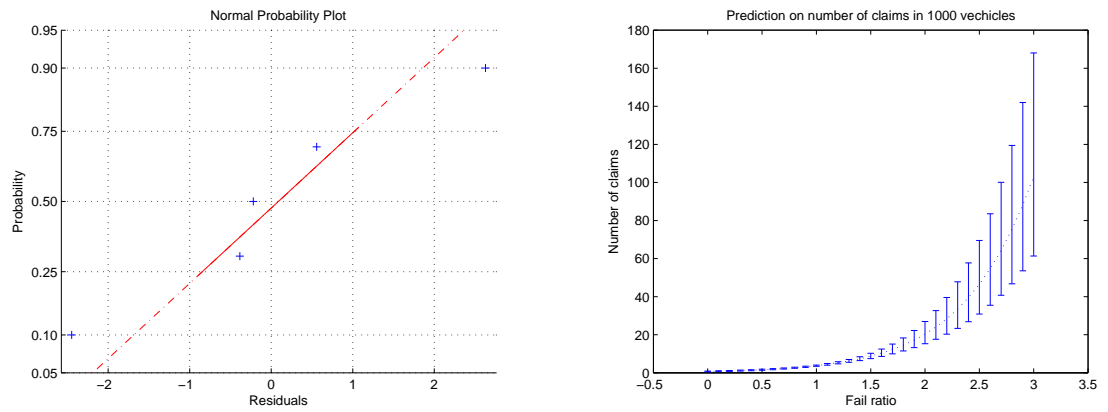


Figure 5.8. Prediction results

Chapter 6

Conclusion

6.1 Correlation

Part I

Normality plots in the appendix B shows how well the internal and external variables fit a normal distribution. Variables such as Hours and Iterations seem far from normal, while Fail and Claim ratio could be normal distributed since data is almost linear. As a consequence, the three different measures of dependencies have been considered in the calculations to ensure detection of correlation. The different calculations have all implied similar correlations with different strength. Pearson's correlation seems to imply stronger correlation than Spearman's rank and Kendall's tau. All correlations computation presented in appendices are based on Pearson's correlation and have been verified through Spearman's rank and Kendall's tau.

We found strong evidence of correlation between the Fail ratio and Open and Hours variables which affirms the general idea of spending more hours finding bugs will result in a higher detection of bugs. Fail ratio seem to be a reasonable measure of testing efficiency since a high Fail ratio indicates an increase risk of finding new issues. We have also calculated and considered Fail ratio as the number of new and old issues per software release since the time span, testing efficiency and circumstances of projects are different. Results have been in concordant and with the original definition of Fail ratio, seen in table 5.2.

Part II

We found almost no correlation between the external and internal variables within different projects. This is not surprising since the projects are based on entirely different ECUs with a variety of properties and a range of complexity. The only correlation observed was between the Claim and Cost ratio which makes perfect sense since both variables are dependent on the number of chassis produced.

Part III

ICL2 projects proved to have strong a correlation between Claim and Fail ratio. A continuous product release has in the case of ICL2 based on five releases proven to be successful in reducing the amount of claims. In conclusion, this confirms RCIV's intuition that a successful product quality improvement is reached by a continuous ECU release.

6.2 Regression

ICL2 scatter plots confirms the result of the correlation analysis. It shows strong evidence of correlation and it is reasonable to assume that the number of claims come from a binomial distribution with a probability parameter P . The simple regression seems as a good model at a first glance but there are two problems with the linear fit. The regression line predicts probabilities less than zero and greater than one which violates Kolmogorov axiom of $0 \leq P(E) \leq 1$ where E is an event from a probability space. The probabilities are also not normally distributed since they are necessarily bounded which in turn violates the basic assumptions in order to fit a simple regression.

Logistic regression is therefore considered and might be a better approach for modelling the dependency. Logistic regression is a special case of a generalized linear model, and is more appropriate than a linear regression for these data for two reasons. First, it uses a fitting method that is appropriate for the binomial distribution. Second, the logistic link limits the predicted proportions to the range $[0,1]$. The normal plot in figure 5.8 shows signs of normal distribution but one can argue that it is not appropriate to check the goodness of fit due to a small sample size. The standardization of residuals are necessary since otherwise the residuals would have different variances which violates the homoscedasticity criteria. This attempt of trying to model data might be useful in the future when there is a sufficient amount of projects with continuous releases. The main conclusions are summarized below.

- Fail ratio seems to correlate significantly with test results during software development and number of claims. This is sound since more failures detected in a shorter time interval results in higher Claim ratio. We therefore suggest the Fail ratio as a complementary acceptance criteria.
- There is evidence of a positive impact on reducing field claims by a continuous product release. Five consecutive ICL2 releases and two Conti tachographs have resulted in an improved quality. In conclusion, we support the idea of continuous product release to ensure quality improvement.
- The allocation of resources should prioritize the outcome of Fail ratio and correlation of test results. This means spending more time on writing sufficient test cases and increasing bench hours which seems to be the main parameters

6.2. REGRESSION

for finding issues. Field tests have been efficient and fruitful when executed by colleagues in RCIV or other development groups. We have on the other hand, not observed many new issues reported from field tests by Scania's customers and can not endorse the extensive field tests since drivers in general are not fully exploring the functionality of all features in an ECU.

- A reliability analysis through computations of MTBF and interarrival time between failures would help monitor the software development process. This could help project managers to estimate the software testing efforts to ensure a satisfactory product quality. We therefore recommend, computing the number of failures during each software release and keeping track of new issues during the development period. Pareto principle could be e.g applied in each software release, to report only releases where at least 20 % of the test cases have been run.
- We endorse the logistic regression as a predictive model for modelling future Claim ratios as a function of Fail ratios. The sample sizes have been insufficient in this research, but the regression model can be useful in the future when one have compiled sufficient number of projects of similar kind.
- The current set of acceptance criteria seem to be uncorrelated to the number of claims. Mileage is a parameter for validation of the product and seem to have no impact on the number of claims. This confirms RCIV's intuition on how inappropriate mileage is in software testing for detecting failures. We suggest an update of the acceptance criteria on the basis of the correlations and outcome of claims
- The quality of Scania's subcontractors has a large impact on the product development and field claims, e.g project Y_8 was never approved by RCIV due to unsatisfactory quality of its ECU supplier which resulted in a high Claim ratio. We believe, Scania's sales and market, responsible for ECU subcontractors should prioritize the quality of the product in order to ensure a satisfactory quality for its customers.

Chapter 7

Future improvement

We consider the follow up of projects and field claims to be a extremely important part of quality improvement at Scania. In order to facilitate a continuous evaluation in the future, one needs to report all projects uniformly to make it easier to compare and analyse test results and field claims. We suggest also to specify day, test iteration and software release of the project when issues are found in order to trace issues accurately.

This research is based on a small set of data, e.g. LAS projects had only one set of data and the DTCO had two sets of data which makes the assumptions and calculations not as accurate as desirable. We endorse a larger set of projects for a continuously evaluation of correlation between test results and fields claims.

A future challenge would be to separate software and hardware related issues and investigate if one can include causality of field claims in SWAT. A more flexible and connected database where ECU projects are directly connected and linked to RCIV's development projects would help the evaluations of field quality issues.

Finally, another interesting method for estimating software testing efforts and connections between test result and field claims, would have been by using Bayesian statistics.

Appendix A

Results

Project	Chassis	Claims	Cost	Cost ratio	Claim ratio
Y_1	16026	101	499125	31.1	0.0063
Y_2	31070	159	991243	31.90	0.0051
Y_3	29418	109	655830	23.31	0.0037
Y_4	29178	65	418934	14.36	0.0022
Y_5	38568	31	206645	5.36	0.00080
Y_6	*	*	*	*	*
Y_7	*	*	*	*	*
Y_8	321	52	648606	2020.58	0.16
Y_9	45082	651	1824384	40.47	0.014
Y_{10}	23047	55	194460	8.44	0.0024
Y_{11}	10386	161	708889	68.25	0.016
Y_{12}	32335	718	2842965	87.92	0.0022
Y_{13}	391	4	18822	48.14	0.010
Y_{14}	4515	137	536371	118.80	0.030

Table A.1. Results of external variables

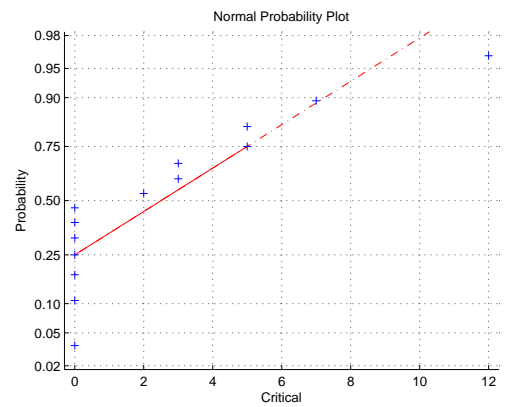
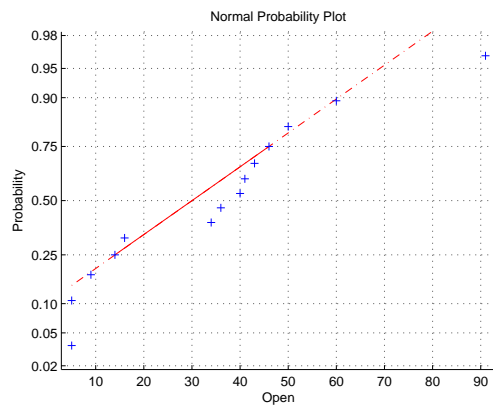
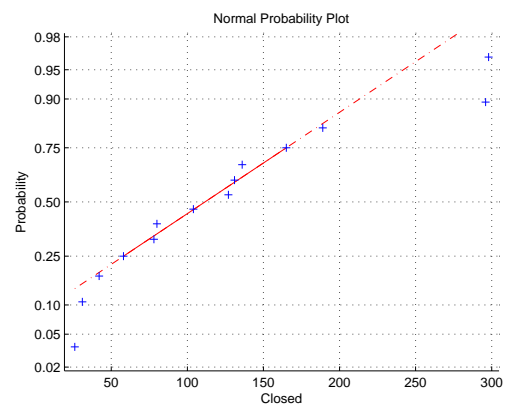
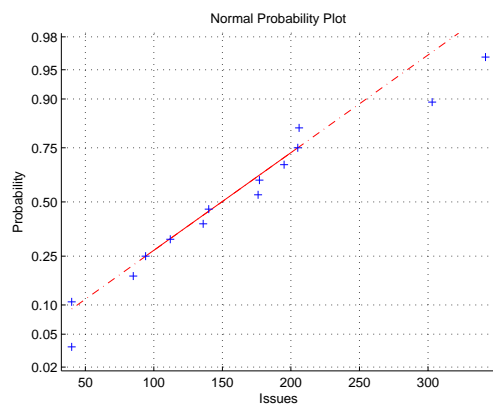
APPENDIX A. RESULTS

Project	Issues	Closed	Open	Critical	Test cases	Hours	Mileage	Iterations	Days	Fail ratio
Y_1	177	127	50	3	697	900	2544461	7	133	1.33
Y_2	195	104	91	0	744	580	522000	4	163	1.19
Y_3	140	80	60	3	762	720	371500	7	145	0.96
Y_4	85	42	43	2	499	200	247103	3	167	0.51
Y_5	94	58	36	0	773	300	143968	6	263	0.36
Y_6	136	131	5	5	162	780	727314	13	244	0.56
Y_7	206	165	41	7	455	1200	874500	13	295	0.66
Y_8	205	189	16	5	143	900	687676	16	360	0.57
Y_9	342	296	46	12	455	612	1734749	26	535	0.64
Y_{10}	303	298	5	0	181	620	2602558	39	539	0.56
Y_{11}	40	31	9	0	107	320	98033	7	435	0.09
Y_{12}	216	143	73	1	129	320	204654	16	435	0.50
Y_{13}	40	26	14	0	246	320	91374	2	135	0.30
Y_{14}	112	78	34	0	153	320	190100	3	252	0.44

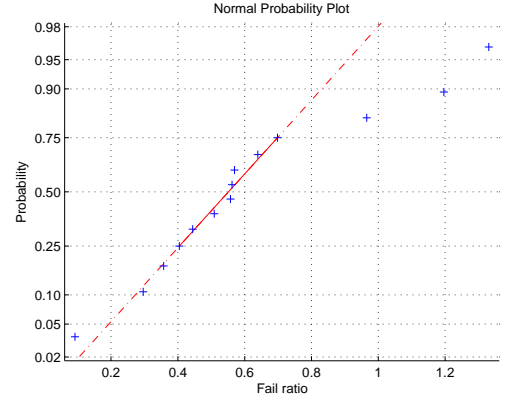
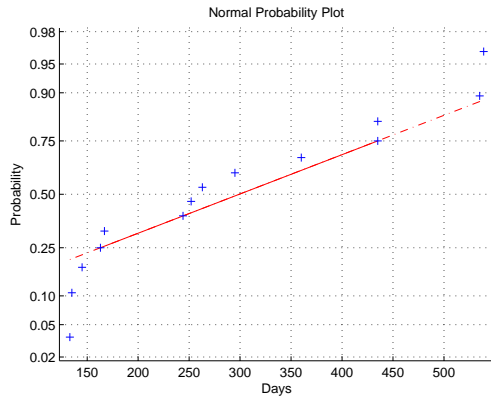
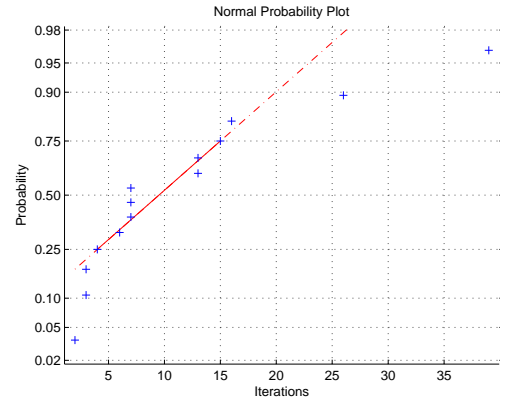
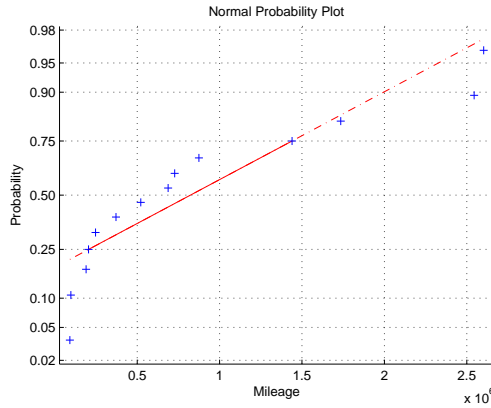
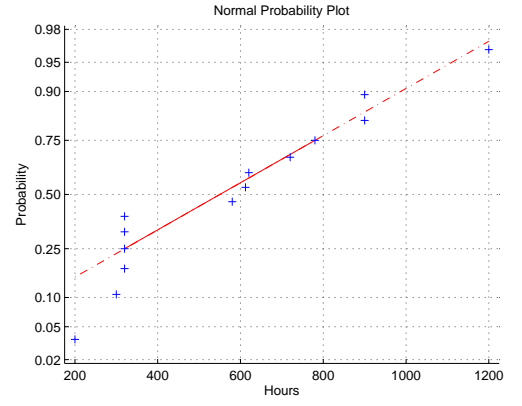
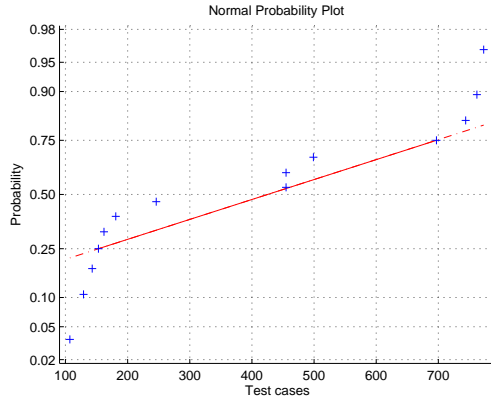
Table A.2. Results of internal variables

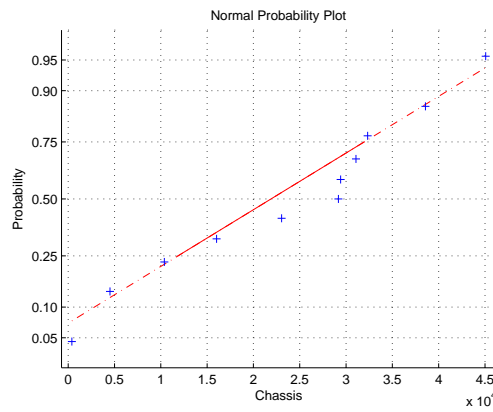
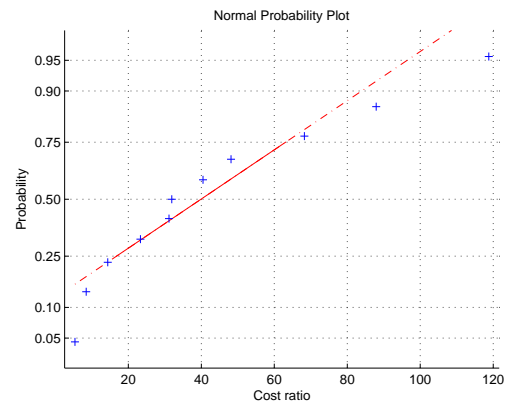
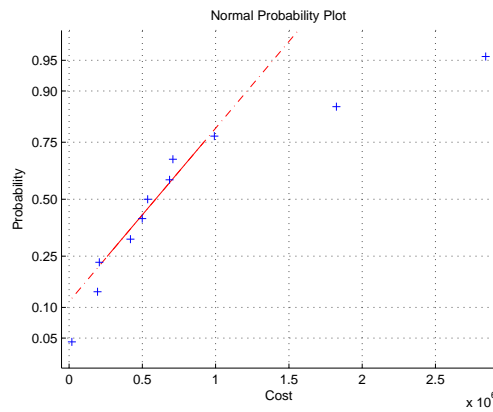
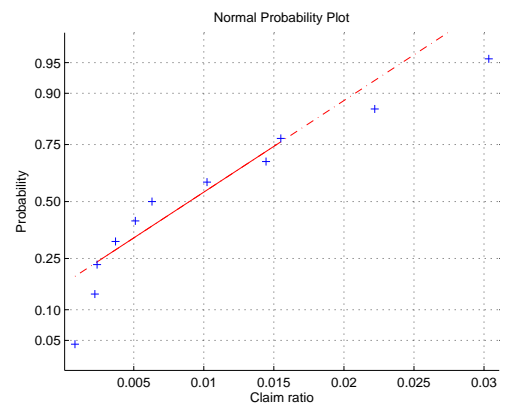
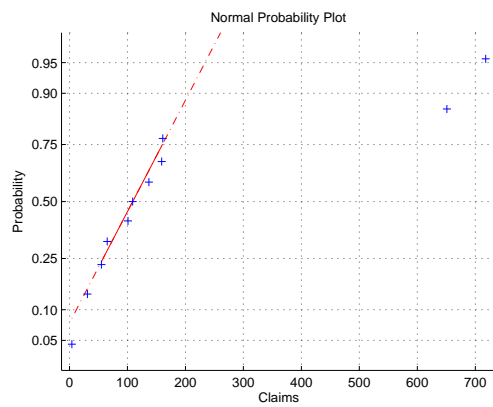
Appendix B

Normal probability plots



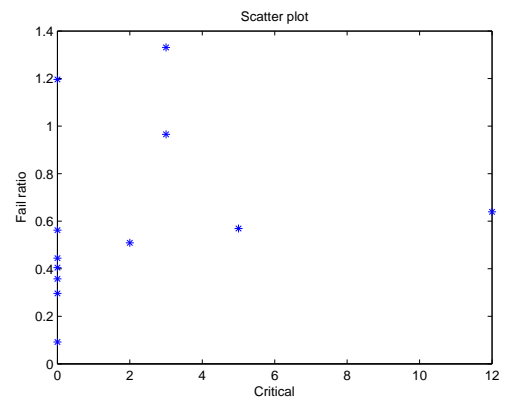
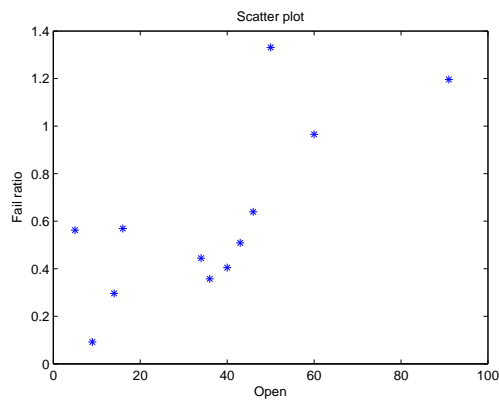
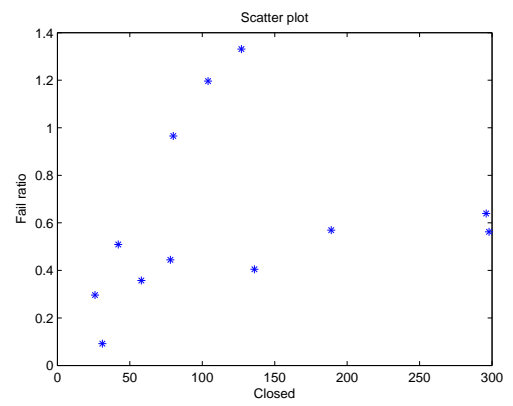
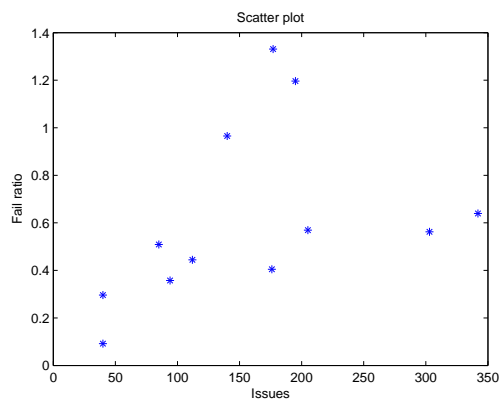
APPENDIX B. NORMAL PROBABILITY PLOTS





Appendix C

Scatter plots



APPENDIX C. SCATTER PLOTS

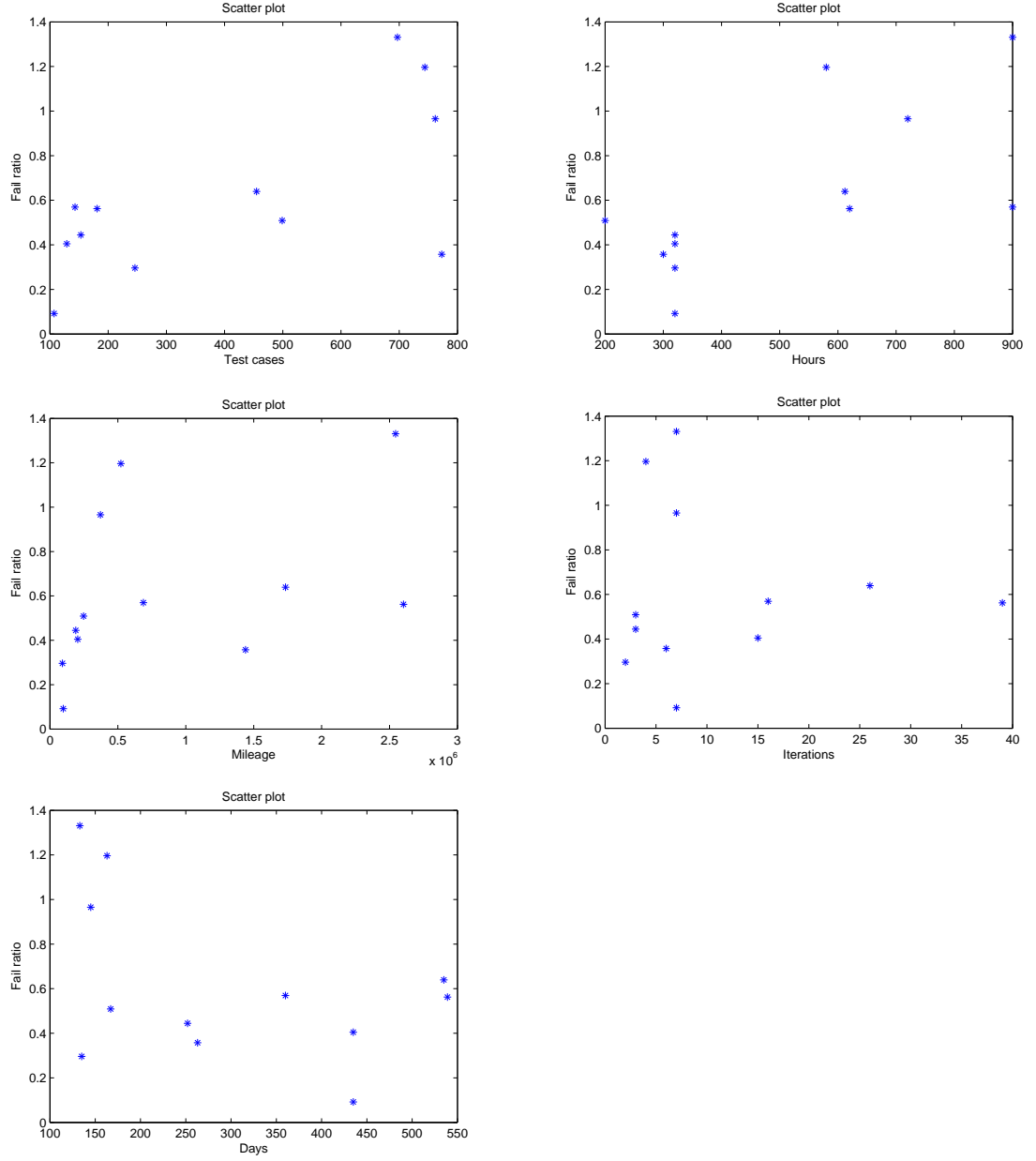


Figure C.1. Scatter plots of internal variables

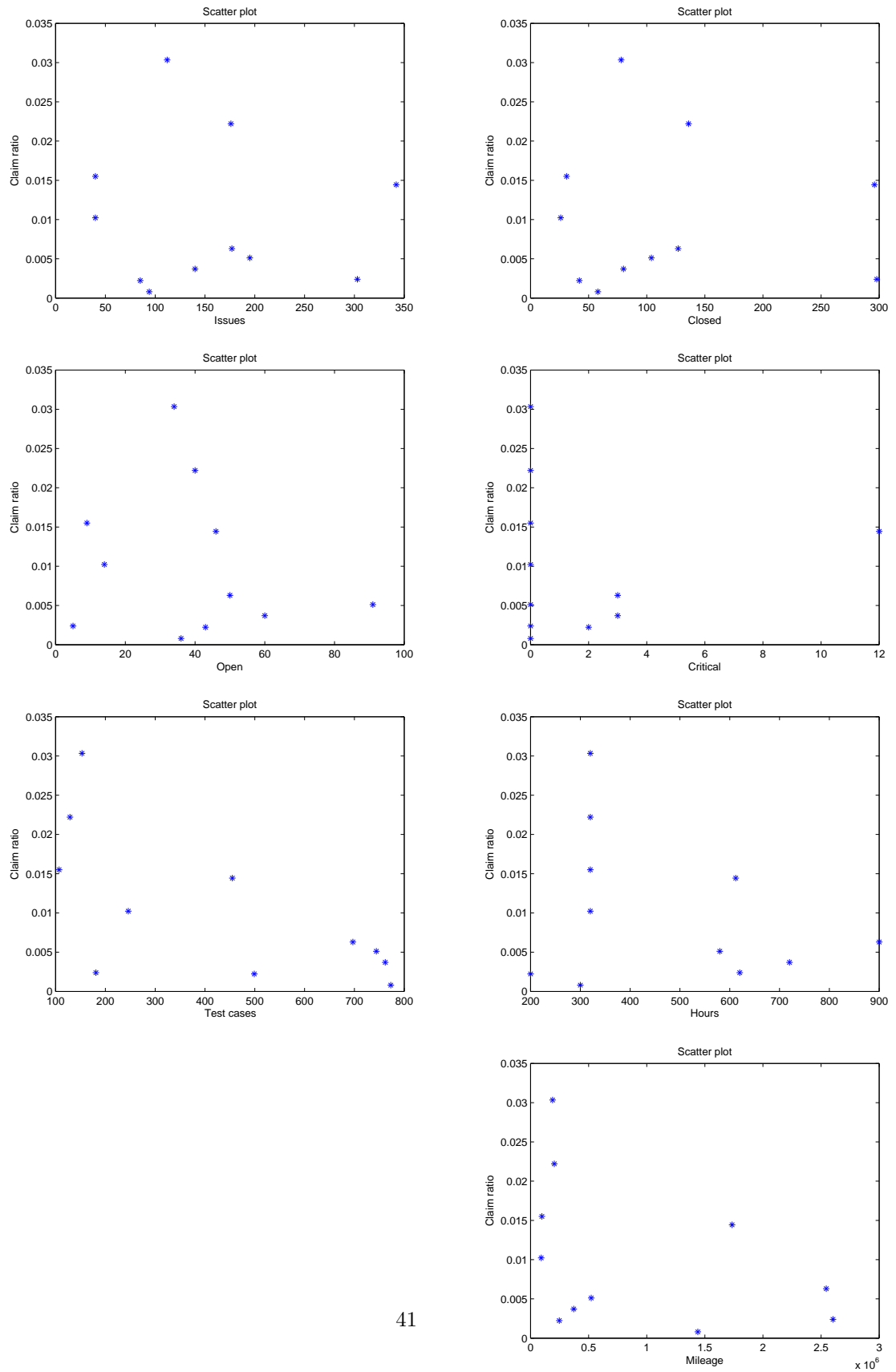
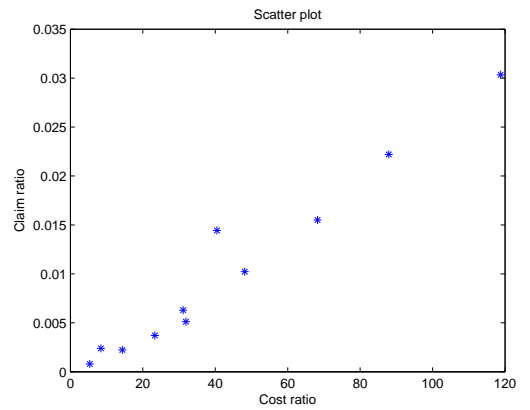
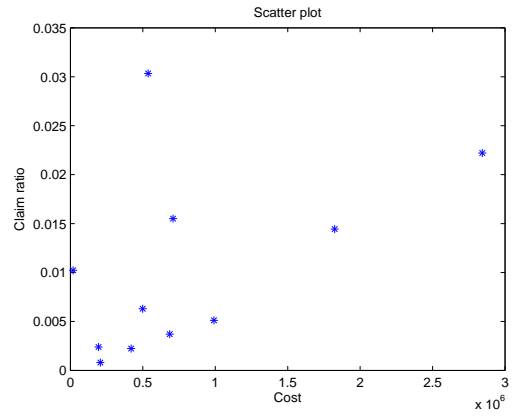
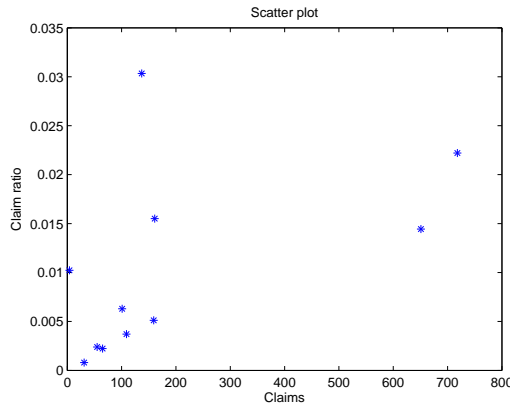
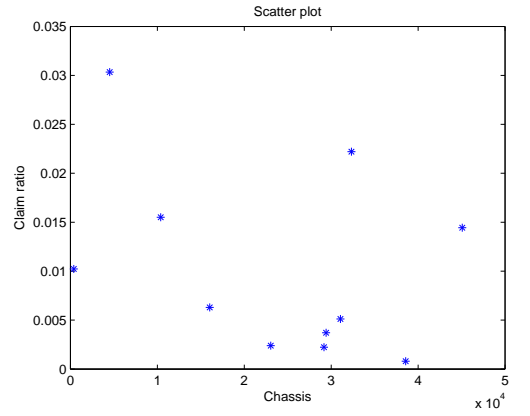
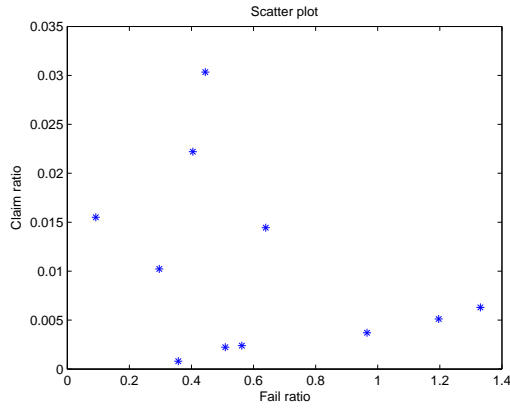
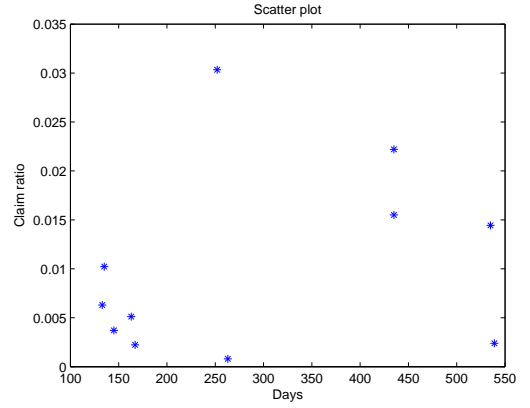
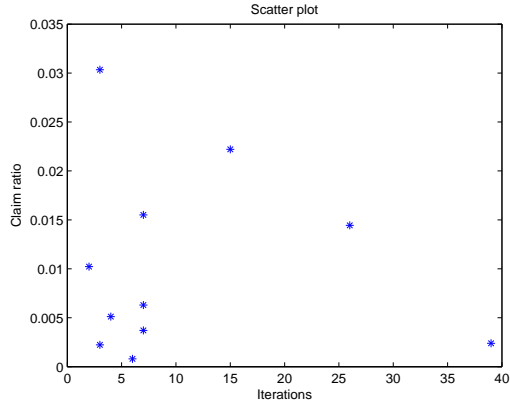


Figure C.2. Scatter plots of internal and external variables

APPENDIX C. SCATTER PLOTS



Appendix D

Correlation tables

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_1	1									
X_2	0.963	1								
X_3	0.169	-0.104	1							
X_4	0.5880	0.578	0.058	1						
X_5	0.0475	-0.162	0.766	0.08	1					
X_6	0.4995	0.483	0.075	0.579	0.149	1				
X_7	0.646	0.664	-0.045	0.263	0.266	0.391	1			
X_8	0.818	0.921	-0.347	0.354	0.315	0.298	0.626	1		
X_9	0.580	0.698	-0.409	0.271	-0.524	-0.01	0.299	0.811	1	
X_{10}	0.399	0.208	0.706	0.208	0.663	0.556	0.414	-0.044	-0.43	1

Table D.1. Correlation table: Part I

	95% CI for Fail ratio	
-0.0796	0.745	Issues
-0.269	0.589	Closed
0.354	0.916	Open
-0.433	0.533	Critical
-0.0471	0.902	Test cases
0.0883	0.763	Hours
-0.074	0.865	Mileage
-0.503	0.346	Iterations
-0.733	0.173	Days

Table D.2. Bootstrap table: Part I

APPENDIX D. CORRELATION TABLES

	95% CI for Claim ratio	
-0.34	0.739	Issues
-0.184	0.847	Closed
-0.72	0.158	Open
-0.567	0.842	Critical
-0.54	-0.289	Test cases
-0.306	0.747	Hours
-0.387	0.301	Mileage
-0.458	0.696	Iterations
-0.207	0.508	Days
-0.385	0.264	Fail ratio
-0.79	0.099	Chassis
-0.615	0.866	Claims
-0.223	0.512	Cost
0.942	0.999	Cost ratio

Table D.3. Bootstrap table: Part II

	95% CI for Claim ratio	
-1	0.975	Issues
0.393	0.997	Closed
-1	0.984	Open
-0.761	0.995	Critical
-1	1	Test cases
0.336	1	Hours
-1	1	Mileage
-1	0.95	Iterations
-0.962	1	Days
0.937	0.998	Fail ratio
-1	0.992	Chassis
-1	1	Claims
-0.524	1	Cost
0.773	1	Cost ratio

Table D.4. Bootstrap table: Part III

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}
X_1	1														
X_2	0.961	1													
X_3	0.199	-0.078	1												
X_4	0.604	0.591	0.085	1											
X_5	-0.017	-0.178	0.573	0.101	1										
X_6	0.532	0.523	0.066	0.442	0.229	1									
X_7	0.615	0.666	-0.145	0.287	0.263	0.534	1								
X_8	0.831	0.924	-0.281	0.372	-0.325	0.338	0.623	1							
X_9	0.608	0.714	-0.34	0.347	-0.57	0.011	0.296	0.825	1						
X_{10}	0.381	0.206	0.649	0.195	0.656	0.683	0.401	-0.041	-0.419	1					
X_{11}	0.452	0.301	0.566	0.352	0.559	-0.102	0.285	0.263	0.215	0.243	1				
X_{12}	0.538	0.424	0.439	0.51	-0.193	-0.082	-0.027	0.34	0.541	-0.011	0.514	1			
X_{13}	0.472	0.324	0.556	0.379	-0.18	-0.037	-0.152	0.242	0.442	0.066	0.455	0.956	1		
X_{14}	0.13	0.211	-0.28	0.247	-0.336	0.485	-0.099	0.118	0.134	-0.062	-0.469	-0.155	-0.035	1	
X_{15}	0.133	0.213	-0.277	0.257	-0.422	0.44	-0.149	0.112	0.179	-0.11	-0.501	-0.075	0.038	0.989	1

Table D.5. Correlation table: Part II

Bibliography

- [1] Scania Inline *Electrical System* 2009
- [2] Akoo Hematbolland *Resource optimization in embedded systems based on data mining* MSc at KTH, 2008
- [3] Christof Ebert, Capers Jones *Embedded Software: Facts, Figures, and Future* IEEE Computer Society Press, Volume 42 Issue 4, Pages 42-52
- [4] Scania Inline *Electronic control units* 2010
- [5] Scania Inline *Bug tracking systems* 2011
- [6] <http://www.bugheaven.com>
- [7] Scania Inline *SWAT* 2009
- [8] Mark Bentham *Acceptance criteria for vehicle test* MSc at KTH, 2010
- [9] Nasir Awan, Adnan Alvi *Predicting software test effort in iterative development using a dynamic Bayesian network* MSc at BTH, 2010
- [10] Boehm, B.W. *Software Risk Management* IEEE Computer Society Press.
- [11] Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli *Fundamentals of Software Engineering* Prentice Hall,
- [12] Boris Beizer, *Software Testing Techniques* Software Testing Techniques. International Thomson Computer Press, 1990
- [13] Cem Kaner *Association for Software Testing* 2004
- [14] Gunnar Blom *Sannolikhets teori och statistik teori med tillämpningar* Lund, 2005
- [15] Kendall MG *Rank correlation methods (4th ed.)* London: Griffin 1970
- [16] Hollander M, Wolfe DA *Nonparametric statistical methods* New York: Wiley
- [17] Marta Monika Wawrzyniak *Dependence Concepts* MSc at Delft, 2006

BIBLIOGRAPHY

- [18] Nancy R. Tague *Seven Basic Quality Tools* <http://www.asq.org/learn-about-quality/sevenbasic-quality-tools/overview/overview.html> 2004
- [19] Gunnar Englund *Datorintensiva metoder i matematisk statistik*
- [20] Bradley Efron, R.J. Tibshirani *An Introduction to the Bootstrap* Chapman & Hall, 1993
- [21] Ulf Olsson *Generalized Linear Models: An applied Approach* Lund 2002
- [22] Donovan.J, Murphy,.E *Reliability growth — a new graphical model* Quality and Reliability Engineering International Volume 15, Issue 3, pages 167–174, May/June 1999
- [23] Ledoux J *Software reliability modelling* Springer 2003
- [24] NIST/SEMATECH *Handbook of Statistical Methods* <http://www.itl.nist.gov/div898/handbook/> 2012
- [25] Menard. S *Applied Logistic Regression Analysis* Sage, Thousand Oaks, 2002

TRITA-MAT-E 2013:08
ISRN-KTH/MAT/E--13/08-SE