UNIVERSITY OF WATERLOO
Faculty of Mathematics

AN EVALUATION OF DESIGN DECISIONS FOR FRESH PRINTS ERP SYSTEM

Fresh Prints
New York, NY, USA

Prepared by
Heli Wang
3A Computer Science
ID 20552080
April 15, 2017

MEMORANDUM

To: Adam Pan (Lead Developer at Fresh Prints)

From: Heli Wang

Date: April 15, 2017

Re: Work Report: An Evaluation of Design Decisions for Fresh Prints ERP System

As we agreed, I have prepared the enclosed report, "An Evaluation of Design Decisions for Fresh Prints ERP System" as my 3A work report. This report, the third of four work reports that the Co-operative Education Program requires that I successfully complete as part of my Computer Science Co-op degree requirements, has not received academic credit.

My job in Fresh Prints is as a software developer. I was responsible for both front-end and back-end development of a web application. This report is an in-depth evaluation of design decisions for Fresh Prints ERP System. This report was written entirely by me with no assistance.

The Faculty of Mathematics requests that you evaluate this report for command of the topic and technical content/analysis. Following your assessment, the report, together with your evaluation, will be submitted to the Math Undergrad Office for evaluation on campus by qualified work report markers. The combined marks determine whether the report will receive credit and whether it will be considered for an award.

Thank you for your assistance in preparing this report.

Heli Wang

Heli Wang

Contents

List of Figures

List of Tables

Executive Summary

Fresh Prints, a custom apparel start-up company, has decided to develop an Enterprise Resource Planning (ERP) web application to manage market sales and order deliveries. It is a fundamental part of the project planning phase to select a programming paradigm among two alternatives—procedure-oriented programming (POP) and object-oriented programming (OOP). This report compares and evaluates these two paradigms according to the Analytic Hierarchy Process (AHP) methodology.

The following set of criteria is specified in the report: scalability, portability, time and cost, reliability, maintainability, and security. Quantitative weightings of these criteria are clarified to reflect their priorities.

These two programming paradigms—POP and OOP—are very different when evaluated according to the above-listed criteria. OOP has considerable advantages regarding scalability, reliability, and maintainability. Its security is also slightly preferred over POP. Conversely, high-portability and low-cost are two significant POP pros.

Ultimately, with a numerical advantage of just 9.2% when calculated according to AHP, OOP is the superior option for Fresh Prints to build its ERP system, which the report suggests.

Introduction

1.1 Purpose of the Report

Fresh Prints is a custom apparel company in the United States that acts as an intermediary between university students, T-shirt designers, and clothing manufacturers. The start-up company is growing lightning-fast, hiring more than one hundred campus managers to expand its business and receiving thousands of purchase orders every month. As it grows, its current IT infrastructure, which is a website built in 2011, can no longer handle more complex operations. A new technology platform that enables the young business to keep track of sales and deliveries is urgently needed.

Enterprise Resource Planning (ERP) is “a business process management ecosystem that automates back-office functions” [NevproBusiness, 2013]. An ERP system usually integrates the most important parts of business, such as product planning, material purchasing, inventory control, finance, and human resources. ERP is used by top apparel companies in America, such as TJ Maxx, H&M, Zara, and Gap, as a key component in their retail services.

Fresh Prints has decided to build its ERP system as a web application. The developers have suggested two separate programming paradigms for the ERP application—procedure-oriented programming (POP) and object-oriented programming (OOP). The purpose of this report is to evaluate these two programming paradigms based on a set of evaluation criteria in order to select the best ERP solution for Fresh Prints. Due to the complexity of the design decision problem, the Analytic Hierarchy Process (AHP) methodology is used to compare the programming paradigms.

1.2 Requirement Specifications

The following criteria have been specified by Fresh Prints for use in evaluating a programming paradigm:

- Reliability: The quality of a product being trustworthy.
- Maintainability: Easiness of fixing a system failure.
- Scalability: Flexibility for potential changes.
- Security: Ability to resist unauthorized visits.
- Portability: Compatibility to run on numerous platforms.
- Time and Cost

The company has ranked the above criteria as follows according to their priorities:
Scalability > Portability > Time and Cost > Reliability > Maintainability > Security

Also, as a requirement, the ERP system must be implemented in JavaScript, a popular web programming language.

1.3 Programming paradigms

1.3.1 Procedure-oriented programming (POP)

A procedure is a set of "instructions to be carried out by a computer" [TechTarget, 2017], which is a countable noun in Computer Science. The computer will follow instructions step-by-step according to a procedure. For instance, the user login page for a website could be a procedure, which might work like this:

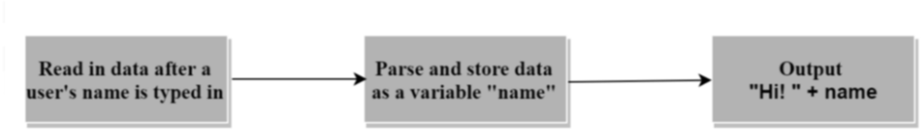

Read in data after a user's name is typed in
Parse and store data as a variable "name"
Output "Hi! " + name

Figure 1.1: The user Login procedure for a website

There are three steps in the procedure example. When the procedure is performed, the user login page will first prompt the user to enter a name. Then, the name data gets stored as a variable. Lastly, a welcome message that includes the username will be displayed.

Variables and statements are essential components of a procedure. A variable represents an area of memory that stores one item of data. Statements are the set of instructions that manipulate variables.

A procedure just performs a single task. A system, like a website, consists of a set of different procedures to execute multiple tasks. POP is a methodology to break down a system into a collection of procedures.

1.3.2 Object-oriented programming (OOP)

POP is an approach to solving a problem by a series of procedures, whereas OOP relies heavily on objects for problem-solving. Objects are building blocks of a program with some properties (fields) and desired functionalities (methods).

For example, a user of a website could be an object. Logically, a user is expected to have a name. In POP, the name is recorded as a variable in a procedure. In contrast, in OOP the name would be a field inside a user object. Users are allowed to have some actions, such as a login action. All of those actions are considered to be methods in an object. OOP is a methodology "to organize a system into different objects, and then assign those objects some fields and methods" [DeNero, 2009].

Evaluation Methodology

Multiple conflicting criteria are involved when choosing the best solution between two programming paradigms for ERP application. In section 1.2, the following criteria have been listed for the evaluation: reliability, maintainability, scalability, security, portability, time and cost. Time and cost are obviously in conflict with the other criteria; after all, it is unusual that the most reliable and the most secure system development solution is the one that costs the least.

This report leverages the Analytic Hierarchy Process (AHP), "a multi-criteria decision-making methodology" [Majumde, 2002], to analyze the complex comparison problem in order to find the solution that best suits the Fresh Prints design goal.

The decision-making problem will be first deconstructed as an AHP hierarchy tree by listing evaluation criteria and all alternatives. Based on the introduction to the decision scenario provided in section 1, the AHP hierarchy can be described as follows:

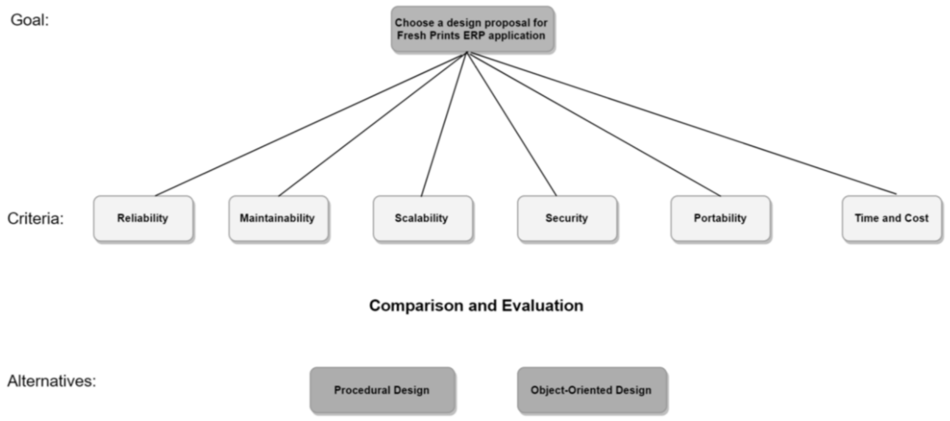
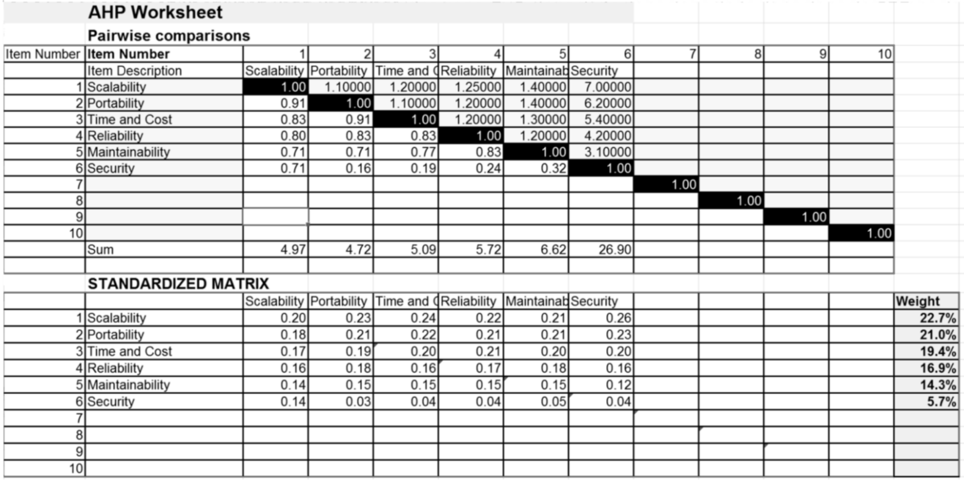


Figure 2.1: The AHP hierarchy tree for this report

The second step for AHP analysis is to transform the qualitative comparisons of criteria into a set of weighted percentages that can be measured quantitatively. Based on discussions with Fresh Prints, the relative importance of the criteria is rated as Scalability > Portability > Time and Cost > Reliability > Maintainability > Security. After conducting pairwise comparisons between these criteria, a set of weightings for each criteria is calculated according to the AHP worksheet, as shown in the tables below.



AHP Worksheet

Pairwise comparisons

Item Number	**Item Number**	1	2	3	4	5	6	7	8	9	10
	Item Description	Scalability	Portability	Time and (	Reliability	Maintainab	Security				
1	Scalability	1.00	1.10000	1.20000	1.25000	1.40000	7.00000				
2	Portability	0.91	1.00	1.10000	1.20000	1.40000	6.20000				
3	Time and Cost	0.83	0.91	1.00	1.20000	1.30000	5.40000				
4	Reliability	0.80	0.83	0.83	1.00	1.20000	4.20000				
5	Maintainability	0.71	0.71	0.77	0.83	1.00	3.10000				
6	Security	0.71	0.16	0.19	0.24	0.32	1.00				
7								1.00			
8									1.00		
9										1.00	
10											1.00
	Sum	4.97	4.72	5.09	5.72	6.62	26.90				

STANDARDIZED MATRIX

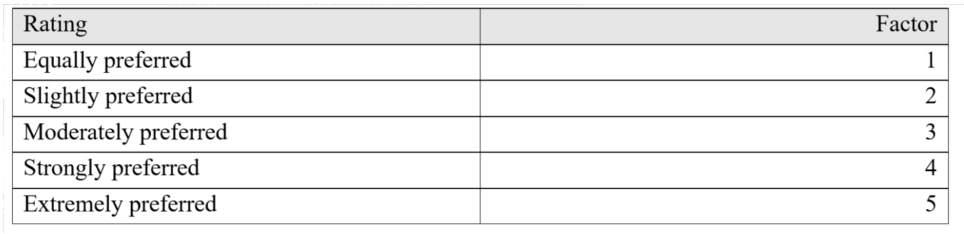
		Scalability	Portability	Time and (	Reliability	Maintainab	Security					**Weight**
1	Scalability	0.20	0.23	0.24	0.22	0.21	0.26					**22.7%**
2	Portability	0.18	0.21	0.22	0.21	0.21	0.23					**21.0%**
3	Time and Cost	0.17	0.19	0.20	0.21	0.20	0.20					**19.4%**
4	Reliability	0.16	0.18	0.16	0.17	0.18	0.16					**16.9%**
5	Maintainability	0.14	0.15	0.15	0.15	0.15	0.12					**14.3%**
6	Security	0.14	0.03	0.04	0.04	0.05	0.04					**5.7%**
7												
8												
9												
10												

Table 2.1: The AHP worksheet for this report

Criterion	Weighting
Scalability	22.7%
Portability	21.0%
Time and Cost	19.4%
Reliability	16.9%
Maintainability	14.3%
Security	5.7%

Table 2.2: Weightings for each criteria

A rating scale that will convert subjective ratings into numerical factor values as defined in the table below will be used to compare POP and OOP.



Rating	Factor
Equally preferred	1
Slightly preferred	2
Moderately preferred	3
Strongly preferred	4
Extremely preferred	5

Table 2.3: Evaluation rating scale

Analytical Comparisons

3.1 Reliability

Reliability is defined as “the quality of being trustworthy or of performing consistently well” [Schenkelberg, 2016]. In the context of the web application that is Fresh Prints ERP system, reliability is measured by the frequency of system failures.

A system failure can either be caused by human errors or be the result of running environment issues. According to a survey from the IDC Enterprise Data Center, human errors contribute to almost half of web application system failures. Thus, an outstanding programming paradigm should keep human errors, such as bugs in source code, well under control.

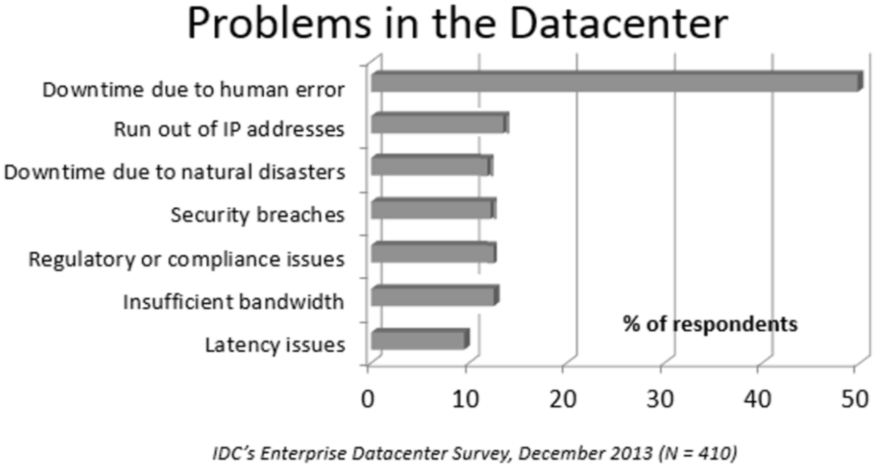


Figure 3.1: A survey of common system failures [Quinn, 2016]

As a tradition in software development, unit testing must be conducted to reduce human error. Unit testing is a process “where the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation" [Mumford, 2017]. The size of each independent unit of an application will determine the difficulty of unit testing.

A unit of code in POP is typically larger than in OOP. In POP, a procedure that performs a particular task, which consists of a set of instructions, is the smallest unit of code. A procedure will build up thousands of lines of code to implement a complex task, making it a challenge to be thoroughly unit-tested.

In contrast, the smallest unit of code in OOP, an object, is just a building block of a program with a few properties and limited functionalities. It is not necessary for an object to cover all the details of a task. Thus, regarding the size of a code unit, an object is smaller than a procedure.

Compared to POP, the complexity of one unit in OOP is significantly less. It is easier to conduct unit testing for OOP in order to correct human errors, making an object-oriented system more reliable. For this reason, OOP is moderately preferred over POP.

3.2 Maintainability

Maintainability is "the ease and speed with which a system can be restored to operational status after a failure occurs" [Fleming, 2017]. The difference between maintainability and reliability is that reliability measures the consistency of systems ability to perform well, whereas maintainability measures the time and effort incurred for a successful repair action to take place after a system is down.

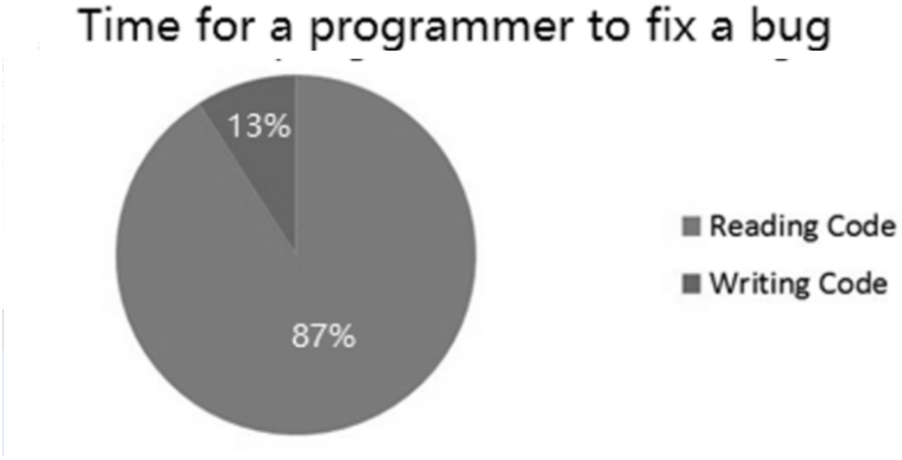


Figure 3.2: Time for a programmer to fix a bug [Kateron, 2012]

When programmers repair a broken system, 87% of their time is spent in reading code, because it is hard to remember all project details as time passes. A clean and reader-friendly code will significantly reduce the debug time and improve a products maintainability.

POP privileges the sequence of actions to be done. A complicated procedure could consist of hundreds of variables and statements, making the code hard to read and modify. Large-scale changes to a procedure are also needed in case of issues.

Unlike POP, OOP focus on an object, which packs data, as well as possible actions, together in each object. In other words, all relevant information about an object is encapsulated within in it. For this reason, only a part of the system needs to be read and be updated to perform maintenance in OOP. Hence, OOP is strongly preferred over POP in term of maintainability.

3.3 Scalability

Scalability is described as “the ability of a system to be readily enlarged as demands increase” [Wicks, 2006]. Code reusability matters for system scalability, since the development process for new features can be accelerated if pieces of code written before can be reused.

In POP, procedures can call each other to share code. However, it is challenging to extend the fixed content in the callee procedure to be more specific for the caller procedure.

OOP provides a data structure “class” to handle code reuse. A class is a template for creating duplicate objects. Nevertheless, a class could also extend the properties and behaviors of another class and override existing functionalities through a “class inheritance” feature. This feature makes OOP more flexible than POP in terms of code reusability. Therefore, OOP will improve software-development productivity, which makes it extremely preferred over POP in regards to scalability.

3.4 Security

When comparing programming paradigms, security is defined as “limiting the visibility of variables inside each independent unit of a system, and only granting needed access privileges to the outside world.” The concept of security here is not about how to protect a website from hackers. Instead, it is about hiding internal data inside one independent unit of source code and preventing illegal access from outside of that unit.

In POP, the current context of a procedure is a “scope”. There are two types of scopes: “global scope,” which is the root scope of a procedure, and “local scope,” which is defined by each statement inside a procedure. The variables defined in a global scope can be accessed everywhere, while the variables inside a local scope can be visited only by the statement that owns the scope. Thus, in POP, the internal data of a procedure are hidden and secured by scopes.

Data hiding is more explicit in OOP than in POP. As stated in section 1.3.2, variables are “fields” in an object. A field must be declared with a modifier, which determines the desired type of access control for the field. There are three types of modifiers: public, protected, and private. They are defined as follows:

- Public is the least restrictive type, indicating the field is accessible anywhere, similar to a global scope in POP.
- Private is the most restrictive type. A private field can only be accessed by the methods inside the same class.
- Protected is less restrictive than private access but more restrictive than public. Methods in the same class, as well as in the inherited classes, can visit a protected field.

Since OOP offers clearer options for data hiding, it is slightly preferred over POP.

3.5 Portability

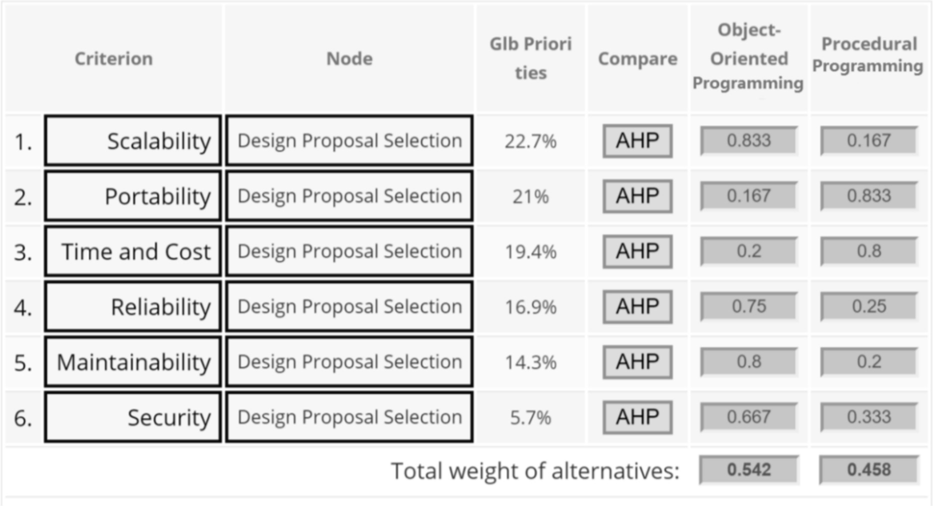
Portability is the ability for a system to run on numerous platforms, which is highly related to the adopted programming language. As required, the ERP system must be implemented using JavaScript, a programming language with two popular versions: ES5 and ES6. According to "ECMA JavaScript standard," only JavaScript ES6 fully supports OOP, while both ES5 and ES6 are compatible with POP. Hence, POP is extremely preferred over OOP for portability.

3.6 Time and Cost

The plan is for the ERP system to be completed by three Fresh Prints co-op students in one year, including training time for the interns. The thought process involved in POP is quite natural for students, while OOP is relatively abstract, with a steep learning curve, meaning that the training process incurs higher costs. Therefore, considering time and cost, POP is strongly preferred over OOP.

Evaluation Results

Weighted scores of the two programming paradigms are calculated by an AHP Online System (bpmsg.com), using the criteria weightings in section 2 and the comparison factors in section 3 as parameters. Table 4.1 provides a summary of the weighted scores of the programming paradigms: OOP weighs 54.2%, which is slightly preferred over POP with a weight 45.8%.



	Criterion	Node	Glb Priorities	Compare	Object-Oriented Programming	Procedural Programming
1.	Scalability	Design Proposal Selection	22.7%	AHP	0.833	0.167
2.	Portability	Design Proposal Selection	21%	AHP	0.167	0.833
3.	Time and Cost	Design Proposal Selection	19.4%	AHP	0.2	0.8
4.	Reliability	Design Proposal Selection	16.9%	AHP	0.75	0.25
5.	Maintainability	Design Proposal Selection	14.3%	AHP	0.8	0.2
6.	Security	Design Proposal Selection	5.7%	AHP	0.667	0.333
				Total weight of alternatives:	0.542	0.458

Table 4.4: Evaluation results

Conclusions

The results of the evaluation show that OOP and POP are very different in some aspects. OOP resulted in considerable advantages regarding scalability, reliability, and maintainability. Its security is also slightly preferred over POP. Conversely, high-portability and low-cost are two significant pros of POP. Overall, as indicated by the AHP calculations, with a numerical advantage of just 9.2%, OOP is the superior option for Fresh Prints to build its ERP system.

Recommendations

All in all, based on the previous analysis, this report recommends that the Fresh Prints ERP system be built on OOP due to its incredible scalability, reliability, maintainability. This section offers some suggestions to compensate for the drawbacks of the recommended paradigm.

Portability is the principal concern stated in the previous analysis. Members of the development team should be careful when selecting JavaScript frameworks for the ERP application. Node.js v8, a server-side JavaScript-based framework with high OOP compatibility, could be considered.

To reduce the training cost, free online courses about object-oriented system design might be excellent supplementary materials. However, the project manager should take potential copyright issues into consideration when developing a training plan with open online courses.

Bibliography

[DeNero, 2009] DeNero, J. (2009). Object-oriented programming. retrieved from http://composingprograms.com/pages/25-object-oriented-programming.html.

[Fleming, 2017] Fleming, N. (2017). Reliability and maintainability. retrieved from http://www.reliasoft.com/blocksim/maintainability$_{a}nalysis.htm$.

[Kateron, 2012] Kateron, J. (2012). Building real software: Fixing a bug is like catching a fish. retrieved from http://swreflections.blogspot.com/2012/07/fixing-bug-is-like-catching-fish.html.

[Majumde, 2002] Majumde, M. (2002). Multi criteria decision making. retrieved from http://www.springer.com/cda/content/document/cda$_{d}ownloaddocument/$9789814560726$-c2.pdf?sgwid = 0 - 0 - 45 - 1504478 - p$176263854.

[Mumford, 2017] Mumford, W. (2017). Unit testing. retrieved from http://softwaretestingfundamentals.com/unit-testing/.

[NevproBusiness, 2013] NevproBusiness (2013). An introduction to enterprise resource planning. retrieved from http://www.nevprobusinesssolutions.com/enterprise-resource-planning/.

[Quinn, 2016] Quinn, K. (2016). Idc survey spotlight: Top problems in the enterprise datacenter. retrieved from http://www.idc.com/getdoc.jsp?containerid=us41291416.

[Schenkelberg, 2016] Schenkelberg, F. (2016). Credibility vs reliability. how is it different? retrieved from https://www.quora.com/credibility-vs-reliability-how-is-it-different.

[TechTarget, 2017] TechTarget (2017). What is instruction? retrieved from http://whatis.techtarget.com/definition/instruction.

[Wicks, 2006] Wicks, T. (2006). ”what is scalability? retrieved from https://dx.doi.org/10.1145