ISSN 2047-3338

Factors Affecting the Successfulness of the Management of the Software Projects

Sayed AbdelGaber AbdelMawgoud[1], Reham Mohamed Younis Hagag[2], Mahmoud Hamdi Keshta[3]

[1,2,3]Department of Business Information System, Faculty of Commerce and Business Administration,
Helwan University, Egypt

***Abstract*– The purpose of this study is to figure out the main reasons that make software projects fail to sort out a new list of reasons that makes software projects fail and determine the most important factors that affect the failures. 270 factors that affect the software project failure have been studied and grouped into new categories, a list of questions has been asked for 4 types of actors "Developer, tester, project manager and End user" answers have used as raw da-ta for the statistical analysis that conducted with SPSS. The results show that the most influencing factors from end user perspective are Implementation actual time and Suitability of project documentation. From development team perspec-tive is suitability of project documentation. From quality assurance engineers is top management support. From project managers perspective is actual time for the implementation.**

***Index Terms*– Reasons of Software Failures, Factors of Software Failures, Business Information System, BIS and IT**

I. INTRODUCTION

THE impact of Corona virus was not only on the health field, but it was on all areas (Wajdi Alhakami, 2020), and it reduced productivity in some areas and greatly increased reliability on the software industry. This rapid and sudden development and dependence on working remotely posed several new challenges, which resulted in the failure of some projects with the new way of working.

One of the biggest challenges facing the development of software field is projects failure, as 31.1% of projects are cancelled before they ever get completed, and 52.7% of projects cost 189% of their original estimates (CHAOS, 2014). Only 16.2% of software projects are on time and on budget. The rest of the 52.7% are delivered with reduced functionality and 31.1% are cancelled before completion (Boehm, 1991).

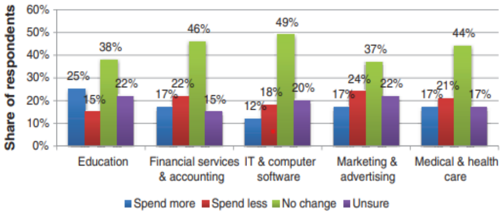

Fig. 1: The Impact of Covid 19

Software development teams and Decision Makers may face a variety of risks at any point during the development process, and an unanticipated risk could result in a significant loss. Furthermore, when new advanced technologies emerge, software development becomes much more complex than previously, necessitating more time and expense. As a result, a greater emphasis on risk factors is required. These factors will enable us to identify the risks that facing the teams while working on the projects, and to develop appropriate plans to overcome them and reduce their impact on the failure. Some measurements for the success or failure of software projects are user satisfaction, ability to meet budget targets, ability to meet schedule targets, product quality and staff productivity (Koru, 2008). In this paper the main categories of the reasons for the software failures are "People, planning, requirements, implementation, environment and communication, difficulty, complexity and delivery" perspective.

In this study, we highlight a list of factors that lead to the success or failure of the software projects and come up with new list of factors that affect the success or the failure and determine what is the fatal factors that make software projects fail.

II. LITERATURE REVIEW

This section examines the literature on the elements that contribute to software project failures. Nineteen scientific publications that have been published between 1991 and 2022 were examined, and the factors that contributed to software

- *Mahmoud Hamdi Keshta is currently pursuing Masters degree program in Business Information System in Helwan University, Egypt, PH +201272729056. E-mail: Mkeshta91@gmail.com*
- *Sayed AbdelGaber AbdelMawgoud Professor of Information Systems*, Faculty of Computers and Artificial Intelligence, Helwan University, Egypt, Mail: sgaber14@gmail.com, PH +20 100 698 8714
- *Reham Mohamed Younis Hagag, Dept. of Business Information System, Faculty of Commerce and Business Administration, Helwan University, Egypt*

project failures were identified and sorted out. Some of the previous research has focused on project failure determinants from the perspective of project managers and ignored other essential viewpoints of other actors. This study focuses on the project manager, the quality assurance team, the development team and the end user to give their opinion about the development cycle. Matching the plan, client satisfaction and increased productivity are all essential factors in determining whether a project succeeds or fails. Moreover, the customer happiness is essential for a company's long-term success as the project isn't finished until the customer is not satisfied and it could fail. So, before asking the actors who are involved in the software life cycle about the reasons behind the failures, the previous studies is a great reference to understand what happened before and budling the new study were the other end. The list of previous factor study exists.

A) Filtering

The collected factors consist of failure reasons are listed into 270 factors according to Appendix1. These factors are required to be filtered and merged. Factors with the same meaning are merged under a new name as Table 1 Filtered factors.

Factors	Renamed factors	No.
80,96,113,128,167,209,232,259,155	Frequent change to the project Structure	1
3,32,35,58,66,36,68,89,93,94,129,130,131,140,150,164,168,169,212,231,242,246,260,262,2,95	Misunderstand / unclear Requirements	2
56,45,70,207,208,213,214,230,237,157, 282	Uncertainty about project objectives/scope of work / inputs	3
49,53,59,61,62,67,77,88,92,111,138,139,162,206,217,220,222,227,238,247,249,252,65,156,191,218,219,84	Poor planning or inadequate planning for (cost / Time/resources)	4
46,69,74,86,102,160,154,240,233,229,109,181	Top management does not have enough experience or involvement	5
51,30,110,141,161,224,254,256,179	Poor Project Manager skills	6
31,33,40,41,57,104,124,126,127,151,163,243,251,261,9,194,47,24	User involvement during the project implementation	7
123,125,268	Resistance to change	
44,106,107,117,136,158,166,245,248,121,15,16,17,18,225	Lack of effective project management methodology	8
42,43,108,216,266,	Turnover	9
39,76,77,152,221,235,258, 265,223,185, 54,133,159, 193	Lack of experienced developers	10
52,37,78,132,134,135,234,239,270,269, 257	Applying new development method /technology during important project	11
55,137,165,210,250	Poor or nonexistent control / monitoring project execution	12
63,75,82,103,142,215,172	Ineffective communication	13
60,72,79,91,101,143,144,145,226,241,244,263,34,186,187,1,4,5,6,7,19,20,21,22,189,184	Poor project team skills or team harmony	15
38,71,85,118,153,236,10,8,99	Delivery and User satisfaction	16
50,83,97,98,112,192,180	Third-party dependency problems	17
73,120	Quality of project documentation and reports	19
90,114,148,149,211,146,147,183,182,177,253	Poor/unstable work environment	20
133, 267,190,119,64,87,105,115,112,173,100	High level of technical/project complexity	21
11,12,13,14,116,23,25,26,27,28,29,17481,195,196,197,198,199,200,201,202,203,204,205,171	Final product quality	23
175	Security failures	24

Table 1: Filtered factors

B) Categorization

Categorization is the process of dividing the world into groups of entities whose members are in some way similar to each other (Jacob, December 2004). The main categories have determined based on everything that should be involved in any Software development life cycle.

Main Category	Factors
People	Project manager skills
	Turnover
	User involvement during the project
	Resistance to change
	Top management experience and involvement
	Team harmony
	Project team skills
Planning	Cost estimation
	Time estimation
Requirements	Project Scope
	System requirement specifications
	Change requirements
Project Implementation	Monitoring and Control
	lack of effective project management methodology
Environment and communication	Communication effectiveness
	Work environment
Difficulty and complexity	Third-party dependency problems
	High level of technical/project complexity
	Development method
Delivery	Quality of project documentation
	User satisfaction
	Security features
	Final product quality

Table 2: Factors categories

III. RESEARCH METHODOLOGY

The Aim of this study is to develop, and authorities list of software project fail or success factors, and to determine which of those factors have the highest impact on the success or failure for software projects. If the project has the most effective methodology without data collection, it will be useless research, as the data collection is one of the most important stages in conducting research.

A) Data Collection

Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes (Sajjad, July 2016). Various techniques can be used to collect data. The research question guides the decision on which data collection technology to use. Data can be collected in a variety of ways such as interview, focus Groups, field Observation case Study, ethnography, oral History, projective techniques, questionnaire and Interview Schedule (Showkat, 2017).

The data has been gathered twice, the first time Linkedin has been used as the source of Project Managers, Developers, Quality assurance engineers and end users but the data wasn't accurate as the different actors wasn't work in the same project. The second time of data collection has been applied on different projects and all questioned actors were working in the same project.

B) Questioner

In the research we will use the questioner as it is the most commonly used method in social sciences, management, marketing and psychology to some extent (Sajjad, July 2016).

1) Sample area

To gather the samples of the data, 5 stages will be taken as showing in Fig. 2.

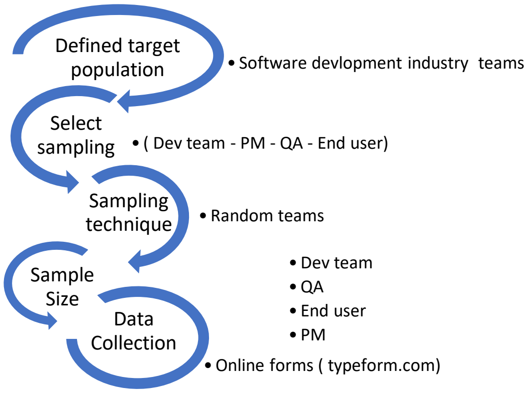
Defined target population
• Software devlopment industry teams
Select sampling
• (Dev team - PM - QA - End user)
Sampling technique
• Random teams
Sample Size
• Dev team
• QA
• End user
• PM
Data Collection
• Online forms (typeform.com)

Fig. 2: Sampling steps

Actors	Views	Starts	Submissions
Dev Team	413	300	189
PM	227	151	119
QA	392	238	157
End user	132	88	55

Table 3: Results of data collection

C) Questions

A set of questions were made for 4 types of actors associated with the projects (Development team – Quality Assurance engineers - Project managers - End users). The questions were prepared to have a quantitative data answer for all questions and the questioner was based on the selection for one choice from five answers. The quantitative data is numerical in nature and can be mathematically computed.

Quantitative data measure uses different scales, which can be classified as nominal scale, ordinal scale, interval scale and ratio scale (Sajjad, July 2016).

The questions that have been asked has implemented to make the understanding of the cycle clear like type of organization, number of team members, project size, implementation time, turnover rate, end user involvement, resistance of change, top management support, team members collaboration, capacity of errors, alignment with the project plan, commitment of the project methodology, satisfaction of team member on the environment, third parties' dependencies, project complexity, suitability of software methodology and suitability of project documentation. Each actor has different extra questions for example the end user has been asked about his satisfaction about the project and if the project become added value and increased the productivity. And the quality assurance engineers have been asked extra questions that focused on the quality of the product and finally the project manager has been asked extra questions from high level perspective and the questions of the developers has extra questions that was focusing on the challenges of development. The project manager questionnaire consists of 24 questions, The development questionnaire consists of 26 questions, The end user questionnaire consists of 15 questions and the quality assurance questionnaire consists of 30 with 10 minutes average for submission.

D) Data validation

After the data have been extracted from the survey system, the answers have been converted from string answers to integer answer for example (very high become 5 and very low become 1). The online form enabled us to make all questions mandatory so, there is no null answers have been entered to the system as well as the online form enabled the data collector to prevent the IP address to enter the answer twice.

E) Data Analysis

1) The Cronbach's Alpha

Cronbach's alpha is a popular method to measure reliability, in quantifying the reliability of a score to summarize the information of several items in questionnaires (Aelst, 2006).

$$\propto = \frac{j}{j-1}\left[1 - \frac{\sum_{j=1}^{n}\sigma_j^2}{\sigma_x^2}\right]$$

Let σ_j^2 denote the variance of item score X_j and $\sigma_{j\,k}$ the covariance between item scores X_j and X_k.

Result of the test

Project Manager	Quality assurance	Tester	End user
.975	.695	.982	.969

Table 4: The Cronbach' Alpha test result

2) Pearson correlation

Pearson's correlation coefficient (r) is a measure of two variables' linear relationship. A scatter diagram is used to graphically show the relationship between data pairs in correlation analysis. Correlation coefficient values range from –1 to +1. Positive correlation coefficient values suggest a propensity for one variable to rise or fall in tandem with another. Negative correlation coefficients imply that an increase in one variable's value is related with a reduction in the other variable's value, and vice versa. Correlation coefficients near zero suggest a weak linear relationship between two variables, whereas those near –1 or +1 indicate a strong linear relationship between two variables. The coefficient of determination is the square of the correlation coefficient, which represents the proportion of variation in one variable that can be explained by variation in the other variable. Pearson's correlation coefficient is based on the following assumptions: (a) a linear relationship between variables, (b) continuous random variables, (c) properly distributed variables, and (d) variables must be independent of one another (Samuels, 2014)

$$\mathbf{r} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Base on the results that have applied on the 4 for types of actors a strong correlation has been found between some of answers as mentioned in appendix 3, So the multicollinearity test has been performed to exclude the high correlation.

3) The Multicollinearity Diagnostics

Multicollinearity can be detected via VIF (Variable Inflation Factors)

$$\mathbf{VIF} = \frac{1}{1 - R^2}$$

Where (R2) is the coefficient of determination of a regression. The higher the value of VIF and the higher the multicollinearity with the particular independent variable. (Abdi, 2007). After figuring out the VIF, we have excluded the answers that have more than 10% VIF and sorted out the answers.

Result of the test

Actor	Measurement factor	Removed factor that have more than 10% VIF
End user	Alignment with project plan User requirement Usability End user happiness	• Type of organization • Frequently meetings • Suggested delivery time
Development team	Alignment with project plan	• Actual time for implementation • Project Manager Skills • Capacity of errors • Work Place • Suggested delivery time • Workplace and productivity • Third party dependencies • Understandability of software • Development methodology • Project documentation Readability
Quality assurance	Alignment with project plan	• Team Size • Project Size • Actual implementation time • Capacity of errors
Project Manager	Alignment with project plan	• Team Size • Project Size • Suitability of software development methodology • Project Documentation readability

Table 5: Execluded factors

IV. RESULTS

S	Factors	PM	Dev	QA	End User	Mean
1	Type of organization	40	42.1	40.1	NA	40.7
2	Project manager skills	NA	NA	35.2	46.7	40.9
3	Team turnover	35.7	58.1	44.3	NA	46.0
4	User involvement	38.3	42.7	45.9	NA	42.3
5	User Resistance	36.4	47.3	41.2	NA	41.6
6	Top managers support	48.0	50.3	54.0	NA	50.7
7	Cooperation between team members	50.7	48.6	40.6	NA	46.6
8	Requirements change	50.0	48.1	35.1	40.3	43.3
9	Frequently meetings	46.0	45.6	49.4	NA	47
10	Commitment with project management methodology	46.7	58.3	32.6	NA	45.8
11	Work owners satisfy team member with requirements of equipment	50.0	46.2	49.4	NA	48.5
12	Workplace	47.8	NA	48.2	NA	48
13	Suggested delivery time	NA	NA	39.7	NA	39.7
14	Productivity and workplace	NA	NA	35.7	NA	35.7
15	Third party dependency	36.1	NA	50.0	NA	43.0
16	Project Complexity	44.0	50.5	40.6	NA	45.0
17	Software Development methodology suitability	49.0	51.3	42.6	NA	47.6
18	Software development methodology understandability	NA	NA	45.0	NA	45
19	Readability of project documentation	NA	NA	45.9	48.2	47.0
20	Project documentation understandability	48.4	40.8	48.1	49.6	46.7
21	Project documentation suitability	50.4	64.1	46.8	47.5	52.2
22	Errors during the development	NA	NA	45.9	50.7	48.3
23	Same error repeated	NA	NA	39.8	NA	39.8
24	System Down	NA	NA	43.9	NA	43.9
25	System outage	NA	NA	43.3	NA	43.3
26	Actual time for the implementation	65.0	NA	NA	64.2	64.6
27	Development team size	NA	42.4	NA	NA	42.4
28	Project Size	NA	54.2	NA	NA	54.2

A) End user

Based on the results, it's concluded that the most influential factor in the success or failure of the project is the implementation actual time , and what affects the plan exceedance is the project management skills that give unrea- sonable estimation plan for the project as well as the capacity of errors during the development stage that is a reason to make the delivery time of each phase longer, and the understandability of documentation which makes the require- ments of the project differently understood by all parties in- volved in the project, and the change of requirements during the development phase is a reason to make the implementation time longer.

B) Development team

Based on the results, it's concluded that the most influential factor in success or failure is the suitability of project documentation, as it's the main core that the development team are count on to have the full view about the system idea and features. A Part of that development team turnover comes to the second factor that affects the failure or success for software project, Turnover rates indicate the number of employees who left a company in a certain period. The third high factor that affects the success or failure is commitment with project

C) Project Manager

Actual time of the implemntion comes in the first place in the reasons for the failure of projects from the point of view of the project manager, and the delay in delivering the project on time comes after the weak cooperation between the work team, and the weakness of the project's explanation documents. Also, the change of requirements for the project features during the development phase impact badly on the delivery.

D) Quality assurance team

Top management support cames in the first plae in the reasons for the failure from QA team, as the QA assurance team always have different persepective from development team wich always is not matching the development team needs. in case the top management support is not exisiting, a struggle will be occurs between QA team and development team that will lead the project to fail. Work Place and frequent meetings come in the second stage of the reasons that make software projects fail.

E) General

Actual time of implemtion on the top of list of the reasons that make software projects fail, as the stackholder always expecting the project to be deliverd on time, and delays may make them focus on other things or it takes into account that the work team is unprofessional and reduces the financial support required for the project or cancels it in order to invest in something else, which leads to the failure of the project. To prevent the failure, all of these reasons need to be considered during the planning of the project, starting from cost and time planning to the productivity that is increased by using the system also the customer satisfaction.

V. CONCLUSION

Ninteen papers have been studied to figure out the main reasons of software project failures, these factors have been used as the base of the research, filtering, sorting and merging have been performed on the factors to put them under main categories. List of questions have been questioned to types of actors who were involved in the same project. The results appeared that there are some factors affecting the failures It is more effective than other elements on the project failures, a new list of factors has been generated from the results reflecting the major factors affecting the software project failures.

REFERENCES

Wajdi Alhakami, A. B. (2020). Atrocious Impinging of COVID-19 Pandemic on Software Development Industries. *tech Since Press*, 16.

CHAOS. (2014). *The Standish Group Report.* Chaos.

Boehm, B. W. (1991). Software Risk Management: Principles and Practices. p. 41.

Koru, K. E. (2008). A Replicated Survey of IT. 7.

Jacob, E. K. (December 2004). Classification and Categorization: A Difference that Makes a Difference. *Library Trends*, 28.

Sajjad, S. M. (July 2016). METHODS OF DATA COLLECTION. *ResearchGate*, 77.

Showkat, H. P. (2017). *Data Collection.* Delhi: Pathshala.

Abdi, H. (2007). Multiple Correlation Coefficient. *8*.

Aelst, A. C. (2006). Robust estimation of Cronbach's alpha. *Journal of Multivariate Analysis*, 15.

Samuels, P. (2014). *Pearson Correlation.* Birmingham: Stats Tutor.

CIRTAUTIENĖ, A. E. (2021). TEAM EFFECTIVENESS IN SOFTWARE DEVELOPMENT. p. 14.

Kevin Crowston, H. A. (2003). Defining Open Source Software Project Success. p. 15.

ROY SCHMIDT, M. K. (2001). Identifing Software Project Risks : An International Delphi Study. *Journal of Management Information Systems*, 32.

May, L. J. (2008). *Major Causes of Software Project.* Crosstalk Associate.

Jarmo J. Ahonena, ∗. P. (2010). Software engineering projects may fail before they are started: Post-mortem analysis of five cancelled projects. *The juournal of systems and software* , 1.

Luís M. Alves, G. S. (2021). Longevity of risks in software development projects: a comparative analysis with an academic environment. *Procedia Computer Science*, 8.

Iskanius, P. (2009). Risk Management in ERP Project in the Context of SMEs. *Engineering Letters*, 8.

Sun-Jen Huang and, W.-M. H. (2008). Exploring the relationship between software project duration and risk exposure: A cluster analysis. *Information & Management*, 8.

Mulder, H. (1994). The Chaos report 1994. *THE STANDISH GROUP REPORT*, 9.

Germán Arias, D. V. (2012). The 7 key factors to get successful results in the IT Development projects. *Procedia Technology*, 9.

Chandan Kumar, D. K. (2015). A Probabilistic Software Risk Assessment and Estimation Model for Software Projects. *Procedia Computer Science*, 9.

Vahid Garousi, M. F. (2019). A survey on software testability. *Information and Software Technology*, 30.

Lang, M., Keaveney, S., & Conboy, K. (2011). Cost Estimation in Agile Software Development Projects. *NUI Galway*, 13.

Langley, M. A. (2017). Success Rates Rise. *PMI* , 32.

Zwikael, O. (2008). Top management involvement in project management. *Emeraldinsight*, 15.

ENFEI, L. (2015). Risk Factors of Software Development Projects in Chinese IT Small and Medium Sized Enterprises. *KTH Royal Institute of Technology School of Industrial Engineering And Management*, 54.

Slaughter, N. L. (2008). Project Managers' Skills and Project Success in IT Outsourcing. *43*.

Martín-Cruz, V. H. (2016). The role of top management involvement in firms performing projects: A dynamic capabilities approach. *12*.

N. Venkatraman, J. E. (1990). Environment-strategy coalignment: An empirical test of its performance implications. *Strategic Managment Journal*.

Tracy Hall, S. B. (2008). The Impact of Staff Turnover on Software Projects : The Importance of Understanding What Makes Software Practitioners tick. *DBLP*.

Muneera Bano, D. Z. (2018). User Involvement in Software Development: The Good, the Bad, and the Ugly. *EEE Software* , 5.

Dennis Pagano, B. B. (2013). User Involvement in Software Evolution Practice: A Case Study. *ICSE*, 10.

Iosif Alvertisa, S. K. (2016). User Involvement in Software Development Processes . *Procedia Computer Science*, 11.

Zafar Ahmed, T. R. (2006). Resistance to change and ERP implementation success: The moderating role of change management initiatives. *Asian Academy of Management Journal* , 18.

Ivana Šandrk Nukić, Z. D.-A. (2015). Impact of changes in a project team structure on the team performance. *Elektronički časopis građevinskog fakulteta Osijek* , 10.

Prasad, A. L. (1995). Causes if Inaccurate Software Development Cost Estimates. *Elsevier Science* , 10.

Albert L. Lederer, R. M. (1990). Information System cost Estimating : A Management Perspective. *MIS Quarterly*, 18.

Zakrani abdelali, H. M. (2018). Investigating the use of random forest in software effort estimation. *Procedia Computer Science*, 10.

Sarah Beecham, T. C. (2020). Do Scaling Agile Frameworks Address Risk in Global Software Development An Empirical Study. *Lero*, 88.

Hans, R. T. (2013). SOFTWARE PROJECT SCOPE VERIFICATION THROUGH DELIVERABLE-ORIENTED WORK BREAKDOWN STRUCTURE. *DOI*, 6.

Shalinka Jayatilleke, R. L. (2017). Managing software requirements changes through change specification and classification. *Computer Science and Information Systems*, 27.

Shalinka Jayatilleke, R. L. (2018). A systematic review of requirements change management. *Information and Software Technology*, 23.

Hassan Osman Ali, M. Z. (2013). Change Requirement Management Issues for a Large Software Development Projects. *INFORMATION SYSTEMS RESEARCH AND INNOVATION*, 7.

Alexander Herzfeldt, R. O. (2012). Developing a risk management process and risk taxonomy for medium-sized it solution providers. *AISeL*, 14.

Müller, R. J. (2014). The impact of project methodologies on project success in different contexts. *IPMA*, 18.

Pal, A. M. (2015). Identifying the Reasons for Software Project Failure and Some of their Proposed Remedial through BRIDGE Process Models. *InternationalJournalofComputerSciencesandEngineering*, 10.

Hassan Khalid, F.-u.-a. a. (2017). Root Causes for the Failure of Communication in GSD. *Information Technology & Software Engineering*, 8.

Brittany Johnson, T. Z. (2019). The Effect of Work Environments on Productivity and Satisfaction of Software Engineers. *Software Engineering*, 20.

Keshavarzsaleh, H. T. (2016). Crtitcal Factors that lead to projects Success Failure in Global Martketplace. *Procedia Technology*, 10.

Kanwal Yousaf, A. M. (2012). Improvement in Quality of Software Architecture via Enhanced-Pattern Driven Architecture (EPDA). *nternational Journal of Information Technology and Computer Science*, 10.

Young, D. (2013). Software Development Methodologies. *GDIT*, 11.

Richardson, N. C. (2015). *Software Quality Plan: A Review Report.* Ireland, Galway: DG INFSO.

Prakasam, C. S. (2014). Survey Of Customer Satisfaction In Software Development. *International Journal of Human Computer Interaction*, 7.

Sametinger, J. (2013). Software Security. *20th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems (ECBS)* (p. 1). Linz, Austria: Johannes Sametinger.

Veenendaal, E. v. (2000). Measuring software product quality during testing. *Conference proceedings European Software Quality Week* (p. 12). Erik van Veenendaal.

Appendix 1
Factors

Papers serial	Author	Factors	Factor serial
1	(CIRTAUTIENĖ, 2021)	Time zone differences	1
		Lake of information when working on a common project/product	2
		Lake of clear purpose	3
		Different cultures	4
		Conflicts between team members	5
		Failure to work on a virtual basis	6
		Do not speak foreign languages	7
2	(Kevin Crowston, 2003)	User Satisfaction	8
		User Involvement	9
		Product Meets requirements	10
		Product Code quality	11
		Product Portability	12
		Product Availability	13
		Process Activity	14
		Process Adherence to process	15
		Process Bug Fixing	16
		Process Time	17
		Process age	18
		Developers' involvement	19
		Varied developers	20
		Developers Satisfaction	21
		Developers Enjoyment	22
		Use Competition	23
		Number of users	24
		Downloads	25
		Recognition Referral	26
		Attention and recognition	27
		Recognition Spin off	28
		Influence	29
3	(ROY SCHMIDT, 2001)	lake of top management commitment to the project	30
		Failure to gain user commitment	31
		Misunderstanding the requirements	32
		lack of adequate user involvement	33
		lake of required knowledge / skills in the project personnel	34
		lack of frozen requirements	35
		Changing scope / objectives	36
		introduction of new technology	37
		Failure to manage end user expectations	38
		Insufficient/inappropriate staffing	39

		Conflict between user departments	40
		lack of cooperation from users	41
		change in ownership or senior management	42
		Staffing volatility	43
		lack of effective project management methodology	44
		unclear / misunderstood scope / objectives	45
		Improper definition of roles and responsibilities	46
		number of organizational units involved	47
		No planning or inadequate planning	48
		Artificial deadlines	49
		Multi-vendor projects complicate dependencies	50
		Lack of skills in project leadership	51
		trying new development method /technology during important project	52
		Bad estimation	53
		new and/or unfamiliar subject matter for both users and developers	54
		Poor or nonexistent control	55
4	(May, 2008)	Poor User Input	56
		Stakeholder Conflicts	57
		Vague Requirements	58
		Poor Cost and Schedule Estimation	59
		Skills that Do Not Match the Job	60
		Hidden Costs of Going "Lean and Mean"	61
		Failure to Plan	62
		Communication Breakdowns	63
		Poor Architecture	64
		Late Failure Warning Signals	65
5	(Jarmo J. Ahonena, 2010)	Making an unrealistic tender or agreement due to lack of understanding of the real needs of the customer	66
		Taking serious risks by agreeing to the customer's demands of a tight schedule	67
		Promising to extend the functionality of an existing product without deep understanding of the technical problem	68
		staffing decisions made due to the unavailability of experienced people	69
		the requirement engineering documents were reused in the new project	70
6	(Luís M. Alves, 2021)	Delay or non-fulfillment of dates on delivery of artifacts	71
		Lack of effort and commitment of the team members to the project	72
		Quality of project documentation and reports	73
		Workload/hours for some team members	74
		Communications difficulty between team members	75
		Loss of team members	76
		Shortage of time and resources	77
		Lack of knowledge of the tools being used	78
		Inexperience of team members	79

		Changes in requirements by the customer	80
		Complexity of the system functionalities used in the project	81
		Difficulty in communicating and gathering customer requirements	82
		Difficulty in managing subcontracting	83
		Difficulty in managing the evaluations of other unit courses	84
		Problems with software production	85
		Poor knowledge of the business area	86
		Poor quality of system architecture	87
		Failure in artifact planning	88
		Failure in modeling requested requirements	89
		Lack of adequate space for work and meetings	90
7	(Boehm, 1991)	Personnel shortfalls	91
		Unrealistic schedules and budgets	92
		Development the wrong functions and properties	93
		Development of wrong user interface	94
		Gold-plating (inclusion of functionalities not solicited by the client)	95
		Continuing stream of requirements changes	96
		Shortfalls in externally furnished components	97
		Shortfalls in externally performed tasks	98
		Real-time performance shortfalls	99
		Straining computer-science capabilities	100
8	(Iskanius, 2009)	Poor project team skills	101
		Low top management involvement	102
		Ineffective communication system	103
		Low key user involvement	104
		Complex architecture and high number of implementation modules	105
		Bad managerial conduction	106
		Ineffective project management techniques	107
		Inadequate change management	108
		Ineffective consulting services experiences	109
		Poor leadership	110
		Ineffective strategic thinking and planning strategic	111
9	(Luís M. Alves, 2021)	Problems with technical artifacts by third-parties	112
		Constant changing of the technical requirements	113
		Poor development environment acquaintance	114
		Technical issues with development	115
		System test failure	116
		Bad system development management	117
		Delivery failure	118
		Poor component conception	119
		Lack of documentation	120

		Incorrect interaction between organization and system processes	121
		Poor system mapping	122
10	(Sun-Jen Huang and, 2008)	Users resistant to change	123
		Conflict between users	124
		Users with negative attitudes toward the project	125
		Users not committed to the project	126
		Lack of cooperation from users	127
		Continually changing system requirements	128
		System requirements not adequately identified	129
		Unclear system requirements	130
		Incorrect system requirements	131
		Project involved the use of new technology	132
		High level of technical complexity	133
		Immature technology	134
		Project involves the use of technology that has not been used in prior projects	135
		Lack of an effective project management methodology	136
		Project progress not monitored closely enough	137
		Inadequate estimation of required resources	138
		Poor project planning	139
		Project milestones not clearly defined	140
		Inexperienced project manager	141
		Ineffective communication	142
		Inexperienced team members	143
		Inadequately trained development team members	144
		Team members lack specialized skills required by the project	145
		Change in organizational management during the project	146
		Corporate politics with negative effect on the project	147
		Unstable organizational environment	148
		Organization undergoing restructuring during the project	149
11	(Mulder, 1994)	Incomplete Requirements	150
		Lack of User Involvement	151
		Lack of Resources	152
		Unrealistic Expectations	153
		Lack of Executive Support	154
		Changing Requirements & Specifications	155
		Lack of Planning	156
		No longer required	157
		Lack of IT Management	158
		Technology Illiteracy	159
12	(Germán Arias, 2012)	Senior management support	160
		Qualified project managers	161

		Adequate Planning	162
		key users Involvement	163
		Requirement´s management	164
		Monitor and control the project execution	165
		Development team management	166
13	(Chandan Kumar, 2015)	Requirement stability	167
		Requirement Clarity	168
		Requirement dependences	169
		Requirement complexity	170
		Reuse level	171
		Interfacing level	172
		Nos. of programming languages	173
		Product Stability	174
		Difficult level to implement security	175
		Experience on the development process	176
		Development infrastructure availability	177
		Development software availability	178
		Project manager experience level	179
		Project Dependencies level	180
		Maturity level	181
		Motivation level	182
		Effective role of organization	183
		Team focus	184
		Turnover	185
		Team knowledge level	186
		Team experience level	187
		Team size	189
		Project size	190
		Financial feasibility	191
		External dependence level	192
		Client experience	193
		Client participation level	194
14	(Vahid Garousi, 2019)	Observability	195
		Controllability	196
		Complexity and simplicity	197
		dependency	198
		understandability	199
		inheritance	200
		Reliability	201
		Availability	202
		Flexibility	203

		Reusability of test suite	204
		Maintainability	205
15	(Lang, Keaveney, & Conboy, 2011)	Pressure to reduce estimates	206
		Poor or imprecise problem definition	207
		Users' lack of understanding of their own requirements	208
		Frequent requests for changes by users	209
		Management Control	210
16	(Langley, 2017)	Change in organization's priorities	211
		inaccurate requirements gathering	212
		Change in project objectives	213
		Inadequate vision or goal for the project	214
		Inadequate/poor communication	215
		Poor change management	216
		Inaccurate cost estimates	217
		Undefined opportunities and risks	218
		Inadequate sponsor support	219
		Inaccurate task time estimate	220
		Resource dependency	221
		Inadequate resource forecasting	222
		Limited/taxed resources	223
		Inexperienced project manager	224
		Task dependency	225
		Team member procrastination	226
17	(Zwikael, 2008)	poor project planning	227
		a weak business case	228
		lack of top management involvement and support	229
18	(CHAOS, 2014)	Lack of User Input	230
		Incomplete Requirements & Specifications	231
		Changing Requirements & Specifications	232
		Lack of Executive Support	233
		Technology Incompetence	234
		Lack of Resources	235
		Unrealistic Expectations	236
		Unclear Objectives	237
		Unrealistic Time Frames	238
		New Technology	239
19	(ENFEI, 2015)	Insufficient responsibilities	240
		Team members lack of cooperation	241
		Development teams misunderstand requirements	242
		Neglect user involvement and inadequate communication with users	243
		Lack of required technical knowledge and skills in the project personnel	244

		Lack of effective development process/methodology	245
		Requirements not adequately identified	246
		Lack long term plan	247
		Lack of effective project management methodology	248
		No planning or inadequate planning	249
		Project progress not monitored closely enough	250
		Conflicts with user representatives	251
		Inadequate estimation of required resources	252
		Lack of technical equipment’s	253
		Lack of top management commitment to the project	254
		Inexperienced project managers	256
		Technology has deficiency	257
		Staff turnover in an ongoing project	258
		Continually changing requirements	259
		Users are not clear about requirements	260
		Lack of cooperation and support from users	261
		Unreasonable requirements	262
		Lack of effective "people skills"	263
		Change in government policies	264
		Short of staff, need more people	265
		Change in organizational management	266
		Too high-level technical complexity	267
		Users’ resistance to change	268
		Change of development tools	269
		Introduction of new technology	270