



Software Failures: A Review of Causes and Solutions

¹Dauda I. A., ¹Nuhu B. K., ²Abubakar J., ¹Abdullahi I. M., ¹Maliki D.

¹Computer Engineering Department,
Federal University of Technology, Minna.

²Computer Science Department,
Federal University Wukari, Taraba.

ABSTRACT

Software failure occurs when the developed software swerves from the expected behaviours or could not execute the task it was developed to perform. Software failures could lead to different degrees of harm to organizations or individual businesses, which include but not limited to financial losses, embarrassments and damage to organizations' reputations. This study reviewed and analyzed several related works in this domain and put more lights on the factors that make software either fail or become inoperative. From the various analyses, it is discovered that failures occur due to schedule pressure, deficient requirements, lack of technical skillset, unrealistic requirements and lack of discrete allocation of tasks. It is therefore imperative to the new and existing organizations to understand these causes and devise a realistic measure to ensure their software perform adequately.

ARTICLE INFO

Article History

Received: August, 2020

Received in revised form: December, 2020

Accepted: January, 2021

Published online: April, 2021

KEYWORDS

Software, Review, Causes and Solutions

INTRODUCTION

The advancement in Information Communication Technology (ICT) has made software a significant tool for business transactions such as e-commerce, banking and stock exchange. It also plays vital roles in the daily activities of schools and offices. Software is a computer system instruction, otherwise known as programs or data (Jalote, 2012). It guides (instructs) the computer system on the steps to execute a certain task. The performance of the software is measured based on predetermined outputs and behaviours; hence, the software is considered to fail when it cannot effectively execute the tasks it was developed for (Chomal & Saini, 2012).

Several concept and models abound that can cause the software to be unsuccessful though, the most common ones are faults and failure –these models

sturdily look identical, in a way, software faults could exist and persists undetected without necessarily portray system failure (Putnam-Majarian, 2015). Whereas software faults occur due to various factors, lack of technical know-how to curtail numerous situational issues between concerned stakeholders is a major cause of software engineering failures (Clarke, 2012). On the other hand, software service provider and consumers are the principal entities that bore the consequences of software failure, though, colossal impacts are mostly felt by the service provider who suffers from a substantial loss of revenue and reputation (Eloff & Bella, 2018). While e various factors are responsible for software failures there are common reasons for software failure such as logical omission or execution error, computation and data operating errors, interface errors on software and hardware as well as data

*Corresponding author: Dauda, A. I. ✉ Idris.dauda@futminna.edu.ng ✉ Computer Engineering Department, Federal University of Technology, Minna. © 2019 Faculty of Technology Education, ATBU Bauchi. All rights reserved



and document problems (Zhu et al, 2010). In the same way, software project schedule pressure has been identified as one of the common factors or barriers that could lead to software failure due to the volume of tasks and pace at which developers or engineers are required to execute a software project (Khan & Malik, 2017).

As a result of schedule pressure, many of the software projects are executed behind schedule and are classified as failure even before the commencement of such projects (Stephens, 2015). Agreeably, computer software is categorized into applications software and system software. The latter administers system resources and also serves as a platform to run the former, which is developed to solve specific tasks. Furthermore, recent studies have shown that the two categories of software could fail due to certain factors, which include: insufficient software requirement, funding, lack of functional testing and the use of wrong methodologies and tools – however, most recent software failure is attributed to the security concerns that resulted from the quest for performance from such software (Chhillar & Sharma, 2019). Despite the evolution and maturity of software development processes, many organizations, have been negatively impacted by the performance and security issues of software, particularly the highly sensitive and enterprise software systems (Grottke et al., 2015). Software failures are a huge setback essentially for software-dependent businesses, more importantly, world economies that depend on software to perform business transactions and daily activities. It is therefore imperative for software developers to adopt a standard software development approach and clear methodologies to efficiently curtail such failures. Besides, software development processes are faced with many challenges, among which are: standardization, schedule pressure, software complexity

and the continuous change in requirements (Chhillar & Sharma, 2019). To minimize or eliminate software failures, standardization and a close check on the technical, human and management factors are key (Khan & Malik, (2017); Khan, et al., (2014)).

LITERATURE REVIEW/ RELATED WORKS

The work of Zahid et al 2013; Clarke & Connor 2012, presents a detailed and critical analysis of situational factors that cause software failure and those that hamper the software development process from attaining success. They examined various existing software development processes how they can be automated to minimize the causes of software failure. While the former pegs the cause of software failures on insufficient requirements provided by clients and misapprehension of software requirements offered by clients to the analyst, the latter proclaims lack of quality standards, understanding of software development processes and wrong use of software development approaches are the principal cause of software failure. However, none of the authors' talks about the technical skillset required to develop robust and quality software.

Similarly, Chen et al., 2018, presents an analysis of mechanism that engender software failure and the impact of failure inherent in the various mechanisms on software quality. It alights certain causes of software failure, however, it was silent on wrong application of engineering principles and requirement change. Accordingly, Dalal & Chhillar 2013, presents a root cause analysis of factors responsible for software failure such as improper conduct of software testing, wrong use of test tools and configuration and RCA (Root Cause Analysis) methods that will enhance and minimize the likelihoods of software failure, though, the paper was silent on certain factors such as

*Corresponding author: Dauda, A. I. ✉ Idris.dauda@futminna.edu.ng ✉ Computer Engineering Department, Federal University of Technology, Minna. © 2019 Faculty of Technology Education, ATBU Bauchi. All rights reserved



the delivery of software project on time and within budget and frequent change of software requirements. On the other hand, Song et al., 2019; Stadler & Seidl, 2013, presented a study on software failure analysis that identify factors that cause the software to fail or make it unsuccessful. While the formerly discussed software reliability as the utmost factor of software quality and the lack of it can make the software fail or unusable the latter identified poor software design as the major weakness that can lead to software failure, however, none take into account factors such as insufficient budget, poor implementation of engineering principles

and indiscrete distribution of software project tasks as factors responsible for software failure or unsuccessful. Also, Shende & Pawar 2016, presented a study that analyzes the factors that cause the software to fail wherein it mentioned various factors such as lack of user involvement, poorly defined scope, poor testing, requirements, managers and cost overrun. However, the study did not take into consideration these factors: market and competitive pressures that exist among technology companies, lack of proper planning and required skillset as well as misapplication of engineering principles.

Table 1.0: Factors that cause software failure

S/No	Authors	Factor and causes of Failures
1	(Chomal & Saini, 2012)	The authors attributed the factors responsible for software failure include but not limited to inadequate requirements, requirements alteration and specifications, absence of user participation, the disparity between the delivered software product and customer requirements.
2	(May 1998)	Factors that engender software failure include the following: poor communication amongst the stakeholders, conflicts between stakeholders, ambiguous requirements often proposed by clients and lack of quality assurance as well as a project underfunding.
3	(Henderson & Peter, 2006)	Delivery of final product behind schedule and proposed project cost exceeded.
4	(Khayyum et al, 2014)	It emphasizes the lack of proper planning, tight schedule and unsatisfactory resources are major cause of software failure.
5	(Khan et al, 2014)	The authors credited the cause of software failure as human, technical challenges and management characteristics.
6	(AcqNotes, 2020)	This author identified factors responsible for software failure as vague requirements and unarticulated software objectives. The use of antediluvian technology and inability to manage competitors and market pressures.
7	(Charette, 2005)	The author underscores the factors responsible for software failure are as following: lack of clear requirement definition reporting and robust communication amongst the concerned team members.

*Corresponding author: Dauda, A. I. ✉ ldris.dauda@futminna.edu.ng ✉ Computer Engineering Department, Federal University of Technology, Minna. © 2019 Faculty of Technology Education, ATBU Bauchi. All rights reserved



S/No	Authors	Factor and causes of Failures
8	(Verner, et al, 2008)	This paper describes major factors accountable for software failure as inadequate assessment of required resources, wrong use of software methodologies as well as incongruent defining of software, user and requirement specification.
9	(Khan & Malik, 2017)	It is established that lack of foreknowledge about software to be developed by customers, schedule and market pressures as well as change of requirements are the primary cause of software failure.
10	(Yusong et al, 2018)	The authors explain that lack of minimum required software expertise knowledge, continually change of software requirements, as well as inadequate software design, are major cause of software failure.

Table 1.1: Reported Software failures in recent years from various sectors

S/No	Authors	Failure Descriptions	The factor responsible for the problem
1	(Computerworld, 2017)	The operational disruption occurred prompted by technical failures which affected flight departure boards and check-in systems and thus denied passengers of information about their flights thereby preventing the customers the use of electronic tickets.	Software failure
2	(Smith, 2020)	Amazon data centres' servers went down due to technical difficulties thus impacted negatively various companies that stored their data at AWS Datacenters.	Software failure
	Google Plus Software security glitches occur which led to the exposure of private information of about 500,000 users using Google Plus across the globe. The reason for the security glitch was attributed to the use of certain API to get access to non-public information thus permitted unauthorized developers to see the personal details of the network's users.		
	Facebook user data leakage took place. A security vulnerability in the system was exploited by hackers to get undue access into accounts and personal information		

*Corresponding author: Dauda, A. I. ✉ Idris.dauda@futminna.edu.ng ✉ Computer Engineering Department, Federal University of Technology, Minna. © 2019 Faculty of Technology Education, ATBU Bauchi. All rights reserved



S/No	Authors	Failure Descriptions	The factor responsible for the problem
		of the user's on Facebook` and the account of the CEO himself. GitLab experienced data losses due to its dedicated server failure. It lost about 300 GB of customers' data in an 18-hour outage.	
3	(Nead, 2020)	Automatic Medical Infusion pumps experienced technical glitch occasioned by software error thus caused the pumps to delay infusion. The flaw got even worse than all distributed/sold pumps had to be recalled to fix the software glitches. Wales national health service experienced extensive hospital computer failure which led to disruption of general activities. The failure made it impossible for doctors to see patient files neither could they access test results thus cause a backlog of other activities even though the system did not allow for cancellations TSB Bank experienced a software glitch that locks customers out of their accounts after a routine system upgrade was performed. The glitch made it difficult for clients to login into their accounts while others are shown details of other peoples' accounts. At Bangladesh Bank, some malicious hackers took control of the bank system and made away with \$81 million in four transactions. Automatically dedicated printers that supposed to print read-outs of all successful transactions instantly refused to print transaction records at the moment of attack until several delays. This software glitch afforded the hackers enough time to getaway.	Software Failure

FINDINGS AND ANALYSIS

This section presents our findings of the research scope and limitation of factors that cause software failure from the review

of related works. Further down are the lists of major factors that can cause the software to fail or unsuccessful where software development is a concern. Thus, the under-

*Corresponding author: Dauda, A. I. ✉ Idris.dauda@futminna.edu.ng ✉ Computer Engineering Department, Federal University of Technology, Minna. © 2019 Faculty of Technology Education, ATBU Bauchi. All rights reserved



listed factors must be carefully put into consideration by the user, analyst and software developer to produce a credible and reliable software product.

Requirement Mismatch or Conflict

Requirements are critical to the success of every software product which must be explained explicitly incongruent to the product proposed functionalities. Consequently, unsubstantiated or ambiguous requirements would invariably lead to the failure of the software product. Arguably, one of the consequences for requirements conflict is that requirement interpreters arrive at a different understanding of what the requirement means or the client goals.

Discrete Allocation of Task

Software development personnel have various skills and individual area of specializations among each other. Tasks should always be discretely allocated amongst concerned persons to reflect their areas of competency as indiscrete distribution of tasks could lead to interpolation of undesirous errors that will either make the final software product fail or unsuccessful.

Schedule Pressure

Schedule pressure is another factor that often acts as a greater obstacle in a successful software development practice. Besides, tight schedules could hinder software personnel performance thus create an avenue to invent errors that will make software unusable. Accordingly, unrealistic goals, too tight and overly optimistic schedules are seen as big factors that engender error-ridden final software to be unproductive.

Incomplete Requirements

Insufficient and poorly incomplete requirements are a pointer to software

failure. Requirements must satisfactorily correspond to a proposed software functionality to avoid devastating failure. Moreover, requirement deficiency could delay the development of the overall product against the specified delivery date.

Insufficient Budget

Underfunding of product development is another challenge in the overall process of software development. Often funds are budgeted for development effort before the establishment of scope and requirements are defined or a complete disregard of former and the latter.

Frequent Change of Requirement

Consistently change software requirements could delay the overall process of development against the normal schedule. Besides, it could broaden the scope of the product leading to personnel's confusion and ambivalent of customer desires hence leads to software failure. This could inadvertently increase the overall budget of the proposed product.

Wrong Application of Engineering Principles

Software is principally developed with clear engineering concepts. Methods and steps must be thoroughly followed from beginning to end to achieve desired goals. However, misapplication of these concepts and methods comes with dire consequences not limited to software unproductivity, unusable and failure.

Lack of Technical Skillset

Advanced knowledge and skills are key to the development of software of any kind. However, lack of required skills and knowledge to develop standardize software leads to the designing of substandard software that is the cost to maintain, hence, bound to fail.

*Corresponding author: Dauda, A. I. ✉ Idris.dauda@futminna.edu.ng ✉ Computer Engineering Department, Federal University of Technology, Minna. © 2019 Faculty of Technology Education, ATBU Bauchi. All rights reserved



Poor Communication

Communication is crucial to the success of software development as lack of it could engender trouble. As the most significant instrument, the system analyst must be able to comprehend and digest users' requirements painstakingly so that he can articulately convey requirement specification to the developer for practical implementation. At any rate, inappropriate information to developers can harm the final software to be unsuccessful.

Lack of Proper Planning

Proper planning is specifically advantageous: it removes grey areas, clarifies ambiguities and doubts thus lack it habitually cause failure in software. At times software project fail due to negligence and lack of application of engineering principles.

Adopt New Technology into Legacy Systems

One of the key challenges for software developers is the adoption of new technologies into the existing systems especially in organizations that deals with a huge amount of sensitive data. Besides, new technologies come with inherent weaknesses that take a long to discover thus integrating them to scale the existing system without corresponding requirements will trigger software failure.

Lack of Proper Testing

Testing is an integral part of the overall process of the software development life cycle. As the name implies, the software is often tested against its requirements to know if the final developed product meets all requirements. Besides, testing is conducted in virtually all stages of software development include unit, integration as well as user acceptance testing. Accordingly, testing must be carried out by a skilled software testing engineer without skipping

any stage as improper or lack of quality testing will invariably lead to software failure.

Market and Competitive Pressure

Many techy giants are never in hesitation to launch new products in the market to assuage their customers while competing against their counterparts. Due to these reasons, some requirements are often not consistent or not well specified which leads to software failure.

Software Development Outsourcing

Companies' associates through partnership to jointly develop a product and share accrue profits based on established principles. Companies' often outsource project or contract that's beyond their organization's manpower, knowledge and skillset required. However, some of the companies that accept outsource projects do not possess the experience, knowledge and skills needed to successfully execute the project which often fails.

CONCLUSION AND FUTURE WORK

Software development is a detailed and methodic process that begins from the requirement analysis phase through to maintenance of the software product. Based on the related works reviewed in this research, it was found that there are a plethora of factors accountable for software failure. These causes can be categorized into four namely: human, technical, user and management factors. Further researches can be done in this direction to provide an alternative method that will minimize or possibly eliminate the causes of failures highlighted in this report.

REFERENCES

- AcqNotes (2020). Software management. Available at:<https://acqnotes.com/acqnote/careerfields/common-software->

*Corresponding author: Dauda, A. I. ✉ ldris.dauda@futminna.edu.ng ✉ Computer Engineering Department, Federal University of Technology, Minna. © 2019 Faculty of Technology Education, ATBU Bauchi. All rights reserved



- failure-causes. Accessed (26-12-202).
- Andrew Smith (2020). The biggest software failures in recent years. While in most cases the programmers' mistakes are not too serious, some IT failures can have truly horrific consequences. Available at: <https://dzone.com/articles/the-biggest-software-failures-in-recent-years>. Accessed (27-12-2020).
- Computerworld (2017). Top software failures in recent history. The biggest software failures in recent history including ransomware attacks, IT outages and data leakages that have affected some of the biggest companies and millions of customers around the world. Available at: <https://www.computerworld.com/article/3412197/top-software-failures-in-recent-history.html>. Accessed (27-12-2020).
- Chhillar, D. and Sharma, K., 2019. Proposed T-Model to cover 4S quality metrics based on an empirical study of the root cause of software failures. *International Journal of Electrical & Computer Engineering (2088-8708)*, 9(2). DOI: 10.11591/ijece.v9i2.pp1122-1130
- Clarke, P. and O'Connor, R.V., 2012. The situational factors that affect the software development process: Towards a comprehensive reference framework. *Information and Software Technology*, 54(5), pp.433-447.
- Chen, Y., Chen, J., Gao, Y., Chen, D. and Tang, Y., 2018, July. Research on software failure analysis and quality management model. In *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 94-99). IEEE.
- Chen Yusong, Chen Jianguang, Gao Yang, Chen Dongfeng & Tang Yuming (2018). Research on Software failure analysis and quality management model. 2018 IEEE International Conference on Software Quality, Reliability and Security Companion. DOI 10.1109/QRS-C.2018.00030
- Chomal, V.S. and Saini, D.J.R., 2012. Identification and Analysis of Causes for Software Failures. *NATIONAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY*, 4(02).
- Dalal, S. and Chhillar, R.S., 2013. An empirical study of root cause analysis of software failure. *ACM SIGSOFT Software Engineering Notes*, 38(4), pp.1-7.
- Grottke, M., Kim, D.S., Mansharamani, R., Nambiar, M., Natella, R. and Trivedi, K.S., 2015. Recovery from software failures caused by mandelbugs. *IEEE Transactions on Reliability*, 65(1), pp.70-87.
- H. H. Khan, M. Naz'ri B. Mahrin, and S. Bt Chuprat, 2014. "Review of support to situational requirement engineering from standards and models," *Int. J. Digit. Inf. Wireless Commun.*, vol. 4, no. 1, pp. 79–83, 2014.
- Henderson, P. (2006). Why large projects fail. *ACM Trans. Program. Lang. Syst*, 15(5), 795-825.
- J. Eloff & M. Bihina Bella (2018). *Software Failure Investigation*. Springer International Publishing AG 2018 7, DOI 10.1007/978-3-319-61334-5_2.
- Jalote, P. (2012). *An integrated approach to software engineering*. Springer Science & Business Media.



- June Verner, Jennifer Sampson and Narciso Cerpa (2008). What factors lead to software project failure?. Conference: Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference. DOI: 10.1109/RCIS.2008.4632095.
- Khan, H. H., & Malik, M. N. (2017). Software standards and software failures: a review with the perspective of varying situational contexts. *IEEE Access*, 5, 17501-17513. 10.1109/ACCESS.2017.2738622.
- May, L.J., 1998. Major causes of software project failures. *CrossTalk: The Journal of Defense Software Engineering*, 11(6), pp.9-12.
- Nate Nead (2020) 10 Examples of software development failure. Available at: <https://readwrite.com/2020/08/20/software-development-failure/>. Accessed (27-12-2020).
- P. Clarke and R. V. O'Connor, "that situational factors that affect the software development process: Towards a comprehensive reference framework," *Inf. Softw. Technol.*, vol. 54, no. 5, pp. 433–447, 2012.
- Robert N. Charette (2005). Why Software Fails. Available at: <https://spectrum.ieee.org/computing/software/why-software-fails>. Accessed (26-12-2020).
- R. Stephens, (2015). *Beginning Software Engineering*, B. Hochgurtel, Ed. Hoboken, NJ, USA: Wiley, 2015.
- Shende, A. and Pawar, J., 2016. Study and Analysis of Factors Responsible for Failures of Software Project. *International Journal of Computer Trends and Technology (IJCTT)*, Volume 41 Number 2, November 2016.
- Samitha Khaiyum, Dr. Y S Kumaraswamy & Dr. K Karibasappa (2014). Significance of failure avoidance in the software development process. 2014 International Conference on Intelligent Computing Applications. 2014 IEEE: DOI 10.1109/ICICA.2014.76.
- Stadler, J.J. and Seidl, N.J., 2013, January. Software failure modes and effects analysis. In 2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS) (pp. 1-5). IEEE.
- Song, J., Zhao, H., Li, X., Yang, Y., Liu, C. and Li, H., 2019, May. A new software failure analysis method based on the system reliability modelling. In 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC) (pp. 1143-1149). IEEE.
- T. Putnam-Majarian and D. Putnam, "the most common reasons why software projects fail," in *Facilitating the Spread of Knowledge and Innovation in Professional Software Development*. San Francisco, CA, USA: Info Que (Electronic Article), 2015.
- Vikas Sitaram Chomal & Dr. Jatinderkumar R. Saini (2012). Identification and Analysis of Causes for Software Failures. *National Journal of computer science and technology*. Volume: 04 | Issue: 02 | July – December – 2012
- Zahid, A.H.A., Haider, M.W., Farooq, M.S., Abid, A. and Ali, A., 2018. A critical analysis of software failure causes from project management perspectives. *VFAST Transactions on Software Engineering*, 6(1), pp.62-68.