Causes of Software/Information Technology Project Failures In Nigerian Software Development Organizations

F.A. Egbokhare
Department of Computer Science
University of Benin
Benin City, Nigeria
Email: fegbokhare@yahoo.com

ABSTRACT
Managing the software development process has become one of the major areas for research especially in today’s information driven economy. A literature search shows that most software development projects are considered as partial failures due to technical and nontechnical factors. There is however dearth of information on studies of the state of software development in Nigeria which forms the focus of this paper. Using descriptive research methods, an introspective assessment of 20 randomly selected software development organizations in Nigeria was performed. The findings show that most software development organizations do not adopt structured software development methodologies. Other factors contributing to failures/chaos in software development in Nigeria are also reported.

Keywords: Software, Information Technology, Project Failures, Nigeria, Software Development and Organizations

African Journal of Computing & ICT Reference Format:

1. INTRODUCTION
The influx of Information and Communications Technology (ICT) and its widespread application in most aspects of human endeavours has increased the demand for computer hardware and software. While the trend in the hardware industry shows remarkable improvement in performance and reduced costs, crisis has loomed the software world which has attracted several research efforts. A literature review of trends in software development show that most software projects are considered as partial failures due to problems arising from the software development process [5][6]and strategies were provided for improving the software process irrespective of the actual life cycle model used. Today, most studies tend to determine the success and failure factors in software projects. Any factors that give good results are consistently applied while process improvement frameworks are proposed to improve any factor(s) that cause problems and affect the achievement of its business goals.

2. RELATED WORKS
Geethalakhmi and Shanmugam [4] studied success and failure factors in software projects from a practitioners’ perspective. This study considered the level of user involvement as one of the main contributors to software project success and failure. The study suggested that user involvement in all phases of software development may enhance the confidence of practitioners and thus improve the software development process. There a re three ‘Ps’ in software development: Problem, Process and People. When these three ‘Ps’ are not properly managed in software projects, crisis can arise. While Abbas and Sanavullah [1] identified flawed software engineering principles resulting from improper management of the three ‘Ps’ as the root cause of software failures, Onibere and Egbokhare [3] identified neglect of the people in software development projects as the main cause of project failures. Al-Ahmad et al.[2] after a comprehensive literature review to investigate and understand the root causes of IT project failures in most domains grouped IT/software project failures into six broad categories namely: Project management factors, Top management factors, Technology Factors, organizational factors, complexity/size factors and process factors. Despite the global trend in the efforts study to improve the software development process, there is no substantive evidence of studies in software project management in Nigeria. The main objective of this research is to study the present state of software development in Nigeria.
3. METHODOLOGY

To gain an insight into the present state of software development in Nigeria, questionnaires and interviews were the major tools for data collection. The questionnaire was adapted from the key practices of the Capability Maturity Model [7], to elicit data on the key practices adopted by Nigerian software development organizations. The sample population a homogenous sample consisting of 20 software project managers randomly selected from four states in Nigeria where there is high level of software development activities. Because the questionnaire was bulky (contained over 50 items) the researcher personally administered it to software managers and software project team leaders in the form of oral interviews and the responses were recorded into the questionnaire. The aim was to elicit information on the base practices adopted during software development projects. Focus group discussion was also used to gather data in this research. This involved an informal interaction with software users and developers to identify the human-level interactions during software development projects.

4. DATA ANALYSIS

To determine the level of adoption of basic software engineering practices, we needed to compute the maturity levels of the software organizations studied. Humphrey [5] Defined 5-steps of process maturity levels that through an introspective self assessment, an organization can know its level on the maturity plateau. The lowest is level 1, where software development practice is chaotic, without sound software engineering practices. The data collected was therefore analyzed using a modified form of the process maturity formula [9]:

\[
\text{Process Maturity Level} = \text{Sat}_{KP}[i] = \# \{KP[i,j] / Passed\}, i = 1,2,\ldots,5 \quad \ldots\ldots(1)
\]

\[
= \# \{ KP[i,j] / r_{KP}[i,j] = 1 \text{ or } r_{KP}[i,j] = 0\} \quad \ldots\ldots(2)
\]

Where:

- \( r_{KP}[i,j] \) is the rating of the \( j^{th} \) key practice at the \( i^{th} \) capability level.

- \# counts the number of KPs (Key Practices) that are satisfied at level \( i \).

A key practice (i) is a cluster of base practices that are expected to exist at a particular maturity level. A pass threshold for a capability level 1, \( P_{KP}[i] \) is defined as:

\[
P_{KP}[i] = N_{KP}[i] * 80\% \quad \text{(i.e. 80\% of the KPs defined for that level should be satisfied).}
\]

\( N_{KP}[i] \) is the number of defined key KPs at maturity level \( i \).

5. FINDINGS/DISCUSSION

Table one show the maturity levels of the organizations studied. From the survey, we discovered that most Nigerian software development organizations (70%) are at level 1 (Initial level). This shows that most software organizations in Nigeria do not define and apply sound software engineering practices in their software development activities. Humphrey [5] observed that most level 1 organizations succeed only through the heroic effort of a competent manager. This may account for the unpredictable and unstable nature of software development in the country and the preference of packaged software systems to custom made software systems by most users.

Table 1: Frequency Distribution of organizations by maturity level

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>No of Organizations</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>20</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 2 show the profile of users/Developers interviewed during the focus group discussion. The questions were open and close-ended with frequent probing to elaborate and clarify meaning. Responses were precise in identification of the factors required. A total of 40 respondents drawn from five sectors of the Nigerian economy as represented in Table 4 participated in the Focus Group Discussions.

Table 2 : Users/Developers Profile

<table>
<thead>
<tr>
<th>S/NO</th>
<th>Sector</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Education</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Banking and Finance</td>
<td>08</td>
</tr>
<tr>
<td>3</td>
<td>Oil</td>
<td>07</td>
</tr>
<tr>
<td>4</td>
<td>Health</td>
<td>05</td>
</tr>
<tr>
<td>5</td>
<td>Manufacturing</td>
<td>03</td>
</tr>
<tr>
<td>6</td>
<td>Government</td>
<td>05</td>
</tr>
<tr>
<td>7</td>
<td>Total</td>
<td>40</td>
</tr>
</tbody>
</table>
To obtain the necessary factors and make valid inferences from the Informal interviews, content analysis was used to analyze interview reports. Content analysis is a method of analysis used in a qualitative research in which text (notes) are systematically examined by developing and grouping themes and coding, classifying and identifying categories. Holsti in Stemler [8] offers a broad definition of content analysis as “any technique for making inferences by objectively and systematically identifying specific characteristics of messages”. It is a research technique used for making replicative and valid inferences from data to their context and a procedure for the categorization of verbal behavioural data, for purposes of classification, summarization and tabulation. The content can be analyzed on two levels. The basic level of analysis is a descriptive account of the data, that is what actually was said with nothing read and assumed into it. The second level of the analysis (sometimes called the latent level) is interpretative: it is concerned with what was meant by the response, what was inferred or implied.

All respondents were promised protection of privacy and confidentiality. The interviewees also gave their consent that the results could be published as long as the issues of confidentiality and secrecy are maintained. The interview sessions demanded considerable skills which included avoiding putting one’s own constructions into the interview, rather, allowing the interviewees to speak. The interview results revealed the following about the perceptions, interactions and experiences of the users and software developers:

- Isolation of actual software users from developers throughout the software development process
- The customers find it difficult to express what they really want and since the developers are not available to aid the customer in this process, requirement definition is usually vague and in some cases, incomplete
- Lack of experienced software development personnel due to poor reward schemes that eventually result in young programmers preferring other highly paid jobs.
- Lack of proper documentation from past software projects
- Poor maintenance culture since stakeholders are not willing to commit resources to preventive maintenance and when they are ready (corrective maintenance), it may be too late
- Slow response to changing technology.
- Organizational politics

6. CONCLUSION

The findings from this study show that software development projects in Nigeria are faced with same global problems reported from the literature review. Because software development is still at the rudimentary stage in Nigeria like most developing countries, software project failures are caused by both technical and non-technical factors. Lack of adoption of software engineering practices (project management factor) and poor human level interactions and organizational politics (organizational factor) are the major causes of software development failures in Nigerian software development organizations. Also, the study discovered that there is lack of definition of roles and responsibilities in software projects, thus most projects are doomed for failure from inception due to poor understanding of user requirements. These factors can be considered in future research efforts to improve product quality and increase software project success.
REFERENCES


