

## The Most Neglected ‘P’ in Software Development Projects

F.A. Egbokhare and E.A. Onibere

Department of Computer Science, University of Benin, Benin City, Nigeria

**Abstract:** Every software development project utilizes three major P’s: problem, process and people. A major challenge faced by software project managers in Nigeria centers on proper coordination and utilization of these three P’s to achieve the set objectives of quality software systems and satisfied customers. This study discusses the role of the three P’s, in the overall activity of software development in Nigeria. Information was gathered from 20 randomly selected software development organizations and the results provide insights to the impact of neglect of software developers on software projects.

**Key words:** Problem, process, people, interactions, software development project

### INTRODUCTION

Baetjer (1998) described the activity of software development as a social learning process in which the knowledge that must become the software is brought together and embodied in the software system. Chiemeke *et al.* (2005) opined that the success of information system development projects is usually affected by the interaction of various factors. These factors, collectively referred to as the three P’s: problem, process and people are represented on Fig. 1.

A software project manager’s greatest challenge anchors on proper coordination and management of these three P’s for competitive advantage and to realize the set objectives of quality software systems and customer satisfaction. Bach (1995) observed that each of the 3 P’s contributes substantially to the final project outcome and partial consideration of any could lead to chaos as reported in Oz (1994), Drummond (1996), Flowers (1997) and Heeks (2002). Paller and Laska (1990) observed that the success or failure of software development projects to some extent depends on the way the development process is managed. This study adopts these definitions for the following groups of people in a software development project:

- A software project manager is an experienced professional who accepts responsibility for planning, controlling and monitoring all activities regarding the development of a particular software system.
- We use the term ‘developer’ to denote a team consisting of systems analysts, system designers, programmers, testers and all those directly involved in the transformation of a customer problem into software systems.

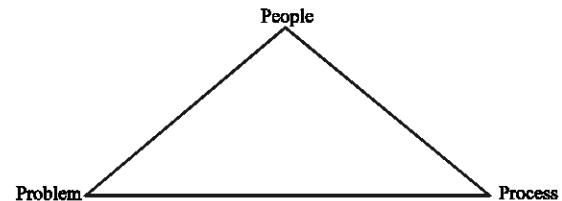


Fig. 1: The software development triangle. Adopted from CMU/SEI-95-MM-01, September, 1995

- The customer is a general term used to denote the end user (user organization), any individual or other stakeholders interested in the outcome of the final software system. The customer may purchase the resulting software system or pay for software development services.

A problem (usually initiated by the customer) is an undesirable situation that prevents an organization from fully achieving its mission or set objectives (Bentley and Dittman, 2004). It contains the business requirements that a customer desires in a software system. These business requirements describe the value that the users, developing organization, or other stakeholders want to receive from the proposed software system (Sommerville, 1997; Pressman, 2001). The main aim of the problem definition phase is to thoroughly study and understand the problem domain and to analyze its opportunities. Because in most cases the customer experiences difficulty, uncertainty and confusion in expressing the exact problem in a manner that yields logical design the systems analyst through series of interactions, which involve different

details, opinions and perceptions carefully studies and understands the user's problem domain, I.

A process according to Webster's dictionary (Webster, 1994), is a system of operations in producing something, or a series of actions, changes, or functions that achieve an end or result. Pfleeger (2001) described a process more comprehensively as a set of ordered tasks involving activities, constraints and resources that produce output. A process involves activities (actions), resources (human and material) and constraints (cost and time). A software system is an output of a process or a set of processes, which encompasses the total set of engineering activities needed to transform users' requirements into complete software systems. A software process leads to the design, development and implementation of a software system. Humphrey (1990) opined that the quality of a software system depends on the quality of the process used to produce and maintain it and that as an organization matures, the software process becomes better defined and more consistently implemented throughout the organization. Today, there is a paradigm shift from methodological approaches to software development using software life cycle models to software process improvement models, which help an organization to carry out an honest introspection and careful analysis of its software development process to identify reasons for previous projects successes and shortcomings. With software process improvement models, any identified success factors are consistently applied, while process improvement frameworks are used to continuously improve any factors that caused problems/failures in past projects.

Passova (2005), viewed the software development process as a multi-level human communication channel. The problem originates from people (customer), creative and knowledgeable people (developers) who use available tools and technology (process) to build the required software systems. Therefore because, the human resource is the most valuable, most volatile and most potentially unpredictable resource, which any organization utilizes, Beardwell and Holder (1997), the central task of a software project manager's job is on how to manage people. If an organization fails to place and direct the human resource in the right areas of business at the right time and cost, serious inefficiencies may arise, which can lead to operational difficulties and possible business failures. How software tasks are performed, the tools/technology used and the precise way and speed with which software development jobs are performed lies primarily on the activities performed by people in a software development project. Weinberg (2004) attributed reasons for most

software project failures to people-related factors such as unrealistic budgets due to inability of people to accept their limitations and 'speak truth to power', loss of emotional control by people when contingencies arise in software projects and lack of progress honestly during software development projects. Brooks (1987), opined that the central question in how to improve the software art centers as it always has on proper management of people.

## **MATERIALS AND METHODS**

Development of quality software systems within estimated cost and schedule and to customer satisfaction has been a major challenge in Nigerian software development organizations. To elicit the necessary information and determine the reasons for most software development projects in Nigeria, 20 software development organizations were visited using questionnaires and key informant interviews. A stratified random sampling method was used. To minimize the possibility of sample bias, the sample population consisted of companies involved strictly with software development selected from 5 major cities in Nigeria, where there is high level software development activity. Three data sets were obtained and analyzed. The results are presented in Onibere and Egbokhare (2007) and human related problem factors observed include:

- Change in essential project personnel due to poor incentives.
- Low skill level.
- High developers mobility due to high market demand for skilled. Information Technology (IT) personnel
- Poor communication among team members.
- Isolation of developers from users.
- Low level of user involvement in software development projects.
- Unrealistic budgets and timelines.

## **RESULTS AND DISCUSSION**

The results from the study of software development organizations in Nigeria revealed certain factors that are responsible for the high rates of failures in software development projects. Most of the failure factors can be traced to improper management of the interactions and roles of the human resource in software development projects.

The venn diagram in Fig. 2 depicts a model for problem-people-process interactions in a software development project.

The areas marked A, B, C and D in Fig. 2 shows specific points of interactions between the 3 Ps.

Area A represent a problem-process interaction. The results of the study of Nigerian software development organizations showed that there is no direct interaction between problem and process without people in a software development project.

Area B represents the peak of activities in a software development project (implementation). The three Ps actually interact at this point with each contributing substantially to the final product outcome. Developers transform the customer problem using a set of processes and tools to a software system under the supervision of a competent project manager. Customer presence is required at this stage of software development because the developers may require certain information which can best be provided by the customer. The study of Nigerian software development organizations however, showed that the customer interacts with the developer for the first time in most software development projects during the testing stage. This isolation of developers from customers was attributed to lack of confidence and trust on the ability of developers to relate well with customers. This must have contributed substantially to most software project failures in Nigeria as represented in Table 1.

All respondents were promised protection of privacy and confidentiality. Company names in this study are denoted with  $C_i$  ( $i = 1...6$ ). Only 6 of the 20 software development organizations studied provided project information and consented that the results can be published as long as the issues of confidentiality and secrecy are maintained. Only one of the 6 companies ( $C_4$ ) was able to complete all projects. Neglect of developers,

which resulted in frustration, de-motivation resulted in high developers mobility which eventually lead to abandoned projects.

Area C shows a problem-people interaction in a software development project. Most software engineering literature refer to this activity as requirement definition/analysis. Ideally, the task of problem definition and analysis is led by the project manager but facilitated by the systems analyst. At this point, a specific subset of people, the developers interact with the customer to understand and transform the problem into a set of software requirements that are implementable.

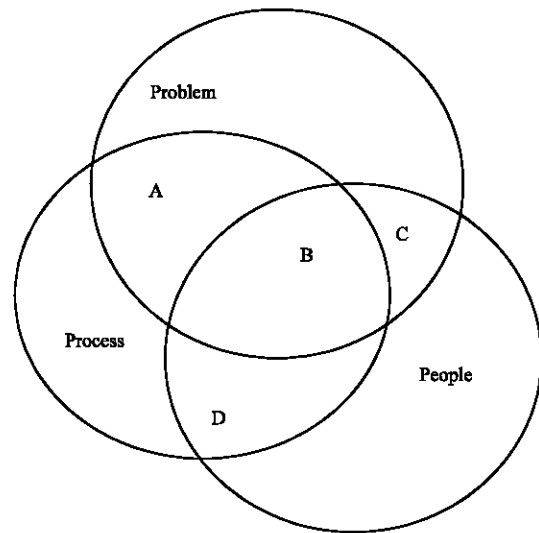


Fig. 2: Interactions between problem, process and people during software development projects

Table 1: Project statistics from 6 software development organizations in Nigeria

Company No.	No. of software projects from 2002-2007	Successful projects	Abandoned/failed projects	Failure factors
$C_1$	3	0	3	Change in essential project personnel Lack of user involvement Lack of developer motivation Incomplete requirements
$C_2$	2	1	1	Lack of user involvement Developers isolation from customers Insufficient communication among team members lack of incentive which resulted in high developers mobility
$C_3$	5	1	4	Low skill level Unrealistic timelines due to non-involvement of developers in requirement definition phase Insufficient communication among team members
$C_4$	2	2	0	Nil
$C_5$	3	1	2	Isolation of developers from users Incomplete requirements Insufficient communication among team members
$C_6$	4	1	3	Isolation of developers from users Low skill level Unclear statement of requirements

This requires series of interactions between the developer (s) the customer. The study of Nigerian software development organizations however, revealed that most software project managers act as 'middlemen' between the customer and software developers, thus, usurping the role of developers and passing incomplete requirements to developers. The resultant systems are unusable and in most cases abandoned.

Area D represents people-process interaction. Humphrey (1990) specified the need for an organization wide process, which should be consistently applied to projects. This processes are defined and used by people in a software development organization and as an organization matures, it evolves from one maturity level to another. The study of Nigerian software development organizations showed that most of them are at maturity level 1 (Onibere and Egbokhare, 2006).

### CONCLUSION

The activity of software development is highly human-centric and management of the human resource is one of the key challenges in a software development project. This study showed that software developers like all other people in software development projects have traditional roles and responsibilities and when these roles are usurped or swapped, problems are bound to arise, which can have drastic effects on job performance. Also, the level of skill in a project has effect on the final product. To retain developers, software project managers must provide good incentives in the form of career advancement, skill expansion, bonuses and raises in salary since the booming technology market now offers wide range of opportunities.

### REFERENCES

Baetjer, H., 1998. Software as capital. IEEE Computer Society Press, pp: 85.  
Bach, J., 1995. Enough about process: What we need are heroes? IEEE Software, 12 (2): 96-98.  
Beardwell, I. and L. Holder, 1997. Human resource management: A contemporary perspective. Pitman Publishing, England, pp: 816.  
Bentley, J.L. and K.C. Dittman, 2004. Systems analysis and design methods. Irwin McGraw Hill.

Brooks, F.P. Jr., 1987. No silver bullet: Essence and accidents of software engineering. IEEE. Comput., 20 (4): 10-19.  
Chiemekwe, S.C., F.A. Egbokhare and F.O. Ikpotokin, 2005. A framework for the development and use of information technology systems. J. Inst. Math. Comput. Sci. (Comp. Sci. Series), 16 (2): 193-206.  
Drummond, H., 1996. Escalation in decision making: The tragedy Taurus. Oxford University Press, Great Britain.  
Flowers, S., 1997. Information system failure: Identifying the critical failure factors. Failure and Lessons Learned in Information Technology Management, pp: 19-29.  
Heeks, R., 2002. Information systems and developing countries: Failure, success and local improvisations. Inform. Soc. Taylor and Francis Publisher, 18 (2): 101-112.  
Humphrey, W.S., 1990. Managing the Software Process. Addison-Wesley.  
Onibere, E.A. and F.A. Egbokhare, 2006. Process assessment: A mirror for software project managers. Nig. J. Applied Sci., 24: 74-81.  
Onibere, E.A. and F.A. Egbokhare, 2007. Software development process: A case of Nigerian software development organizations. Inform. Technol., 4 (1): 1-13.  
Oz, E., 1994. When professional standards are lax: The confirm failure and its lessons. Commun. ACM, 37 (10): 29-36.  
Paller, A. and R. Laska, 1990. The Elsbook, Dow Jones-Irwin, Homewood, IL.  
Passova, S., 2005. The need for customer-oriented software development. Sofea, Inc. e-Biz Development tools. [www.ebiz.net/topics/devtools/features/5577.html](http://www.ebiz.net/topics/devtools/features/5577.html).  
Pfleeger, S.L., 2001. Software engineering: Theory and practice. Pearson Education, North Asia.  
Pressman, R.S., 2001. Software Engineering: A Practitioner's Approach. 5th Edn. McGraw Hill.  
Sommerville, I., 1997. Software Engineering. 5th Edn. Addison-Wesley.  
Webster, M., 1994. Webster's Dictionary of English. Merriam Webster Inc.  
Weinberg, G.M., 2004. Advice for software managers. [www.geraldweinberg.com](http://www.geraldweinberg.com).