

# Software Project Planning with Tracking and Oversight

Aregbesola, Moses Kehinde

Department of Mathematics and Computer Science, Elizade University, Ilara-Mokin, Ondo State, Nigeria

**Abstract-** Software project management has been identified as a major determinant factor of software project's success or failure. Software Project Planning (SPP) and Software Project Tracking and Oversight (SPTO) have equally been identified as two key process areas in software project management at maturity level 2 of the SEI CMMI. Following the need for consistent software process improvement in software industries world-over, especially as it concerns software project management, this study focused on evaluating the current level of performance of the two identified KPAs, SPP and SPTO in the software industry. Besides a broad review of literature, a study was conducted covering 30 software companies in the Nigerian geographical area. The study employed both the survey and the action research approaches with some of the companies selected as case studies for further appraisal. The study revealed some level of strength in the performance of the practices associated with SPP and SPTO in the Nigerian software industry. The results of the study were consistent with findings of earlier studies in enterprises whose maturity levels at the time of study were similar to that of the Nigerian software industry. Several recommendations were given for improving the performance of the practices associated with SPP and SPTO key process areas in particular, and software project management in general.

**Index Terms-** Software Project Planning, SPP, Software Project Tracking and Oversight, SPTO, Nigeria Software Companies, Software Industry, CMMI, Key Practice Area, KPA, Capability Maturity Model Integration, Software Process.

## I. INTRODUCTION

Nan and Harter (2009) described software project management as the art and science of planning, coordinating and providing leadership for software projects. In order to be successful, organizations usually need effective and efficient project plans to help manage and possibly bring down the overall cost of software development. Typically, as a project increases in size, so thus the accompanying complexity, thereby further intensifying the need for thorough planning. A number of studies have shown that a large percentage of project failures were as a result of poor project planning. For instance, Ding and Jing (2003) in their study reported that in China, more than 40 percent of failed software projects were unsuccessful because of ineffective planning of human resources and project tasks. Chang et al. (2008) explained that the main reason for this was that unlike other projects, software project activities were people-intensive and that the related resources were mostly human resources. Meanwhile, the human resource allocation had already been described as a complex task. Kurien and Nair (2014) therefore puts forward that since cost and time are the significant

dynamics in software projects, effective tools for managing these factors could create more room for success. An effective planning tool was considered important because an effective plan maximizes the overall cost and time required for a software project. Some of the project management techniques that have been applied in software project planning include the critical path method (CPM), the program evaluation and review technique (PERT), and the resource-constrained project scheduling problem (RCPS) model (Shtub, Bard and Globerson, 2005). For calculating project workload and cost, as well as deciding project schedule and resource allocation, the COCOMO model (Chen and Zhang, 2013) is the model commonly employed.

Software Project Planning (SPP) and Software Project Tracking and Oversight (SPTO) also known as Project Monitoring and Control are two closely knit key process areas (KPA) at maturity level 2 of the Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI). The SEI CMMI maturity level 2 (Managed) is the maturity level that lay emphasis on software project management. As has already been described, SPP which is closely knit with SPTO play very pivotal roles in the software project management process. A software development plan, the output from the SPP KPA, is a prerequisite for SPTO (Paulk *et al.*, 1993). The established plans from the SPP KPA are the necessary foundation for managing the software project, as described in the SPTO KPA.

The Software Engineering Institute upgraded their capability maturity model (CMM) with the CMMI in 2001 (CMMI Product Team, 2006). The CMM Integration was designed to address the problem of using multiple CMMs. CMMI is an outstanding process maturity framework, made up of 5 maturity levels, for practical organizations that desire to accomplish higher levels of performance in their activities (Hurst, 2017; Glover and Dennie, 2017; O'Neill, 2017). Although software is excluded from the titles and from the topics within the five levels of the CMMI because the CMMI encompasses a broader process-orientation, the current study is however focused on software and therefore includes software in its titles.

It is expected that findings from the current study will provide insight to software organizations across the globe about the current state of their software development process with regards SPP and SPTO. It is equally expected that these findings would endear software enterprises towards higher levels of performance in trying to achieve significant improvement in their software development processes. In addition, although a number of studies have examined software development processes from different perspectives, none of them have focused on examining the SPP and SPTO KPAs in the Nigerian context. The current study examined the level of performance of the SPP and SPTO practices in the Nigerian software industry based on results

obtained from some randomly selected software companies within its geographical space.

## II. RELATED WORKS

Although a number of studies have focused on software development companies in Nigeria including Soriyan, Mursu and Korpela (2000), Soriyan and Heeks (2004), Akinola, Osofisan and Akinkunmi (2009), Aregbesola and Akinkunmi (2010a; 2010b), Aregbesola *et al.* (2011), Aregbesola and Onwudebelu (2011), Akinola and Osofisan (2011), Aregbesola and Oluwade (2014), and Aregbesola (2017a; 2017b; 2017c; 2017d), none of them have actually examined the Software Project Planning (SPP) and Software Project Tracking and Oversight (SPTO) key process areas (KPA). This section of the study reviewed similar studies related to SPP and SPTO KPA. This section has been divided into two subsections, one for each of the discussed KPAs, SPP and SPTO.

### 1.1. Software Project Planning

Software Project Planning (SPP) is aimed at establishing reasonable plans for accomplishing the software engineering goals and for managing of the software project. It involves coming up with estimates for the work to be carried out, establishing the necessary commitments, and outlining the plan to get the work done (Paulk *et al.*, 1993). Typically beginning with the statement of work to be performed, SPP is usually guided by the practices of Requirements Management KPA (Aregbesola, 2017a). SPP involves iterating through a number of activities for estimating the size of the software work products as well as the needed resources. It also involves producing a schedule, identifying possible risks, assessing the risks, and negotiating commitments. The established plan forms the basis for executing and managing the events of the software project's activities and addressing the commitments to the software project's customer on the basis of constraints, resources, and software project capabilities. Paulk *et al.* (1993) further explained that the established plans are the necessary foundation for managing the software project, as described in Software Project Tracking and Oversight.

Software Project Management (SPM) which is one of the primary factors to software success or failure cannot be effectively implemented without realistic plans. Since SPM has been a bottleneck in software engineering, with software project planning being one of its most critical activities, a number of approaches have been developed and applied with the purpose of improving its project planning practices. For instance, Han *et al.* (2015) applied an improved Max–Min Ant System algorithm to Software Project Planning to develop an appropriate worker-task assignment in a software project with the aim of maximizing cost and minimizing duration. The resulting model for software project planning made use of three important resources, namely tasks, employees, and skills. While the tasks were the jobs needed to complete the project, the employees were the workers with the requisite set of skills for performing the tasks. The developed model could make a suitable allocation of tasks to employees based on their skill sets. The experimental results from the study with regards selected projects yielded feasible solutions for optimum cost and duration as well as generating the

appropriate PERT Graph and Gantt chart of the software project, thus improving the software project management process.

Rosso-Llopart (2005) discussed the goal of SPP to involve the establishment of a pragmatic strategy for controlling, tracking, and monitoring a complex technical project. The established strategy must deal with the project's complexity which is heavily influenced by past experience of the practitioner and has a strong effect on the project's overall outcome. It must also address the project's size which typically increases in parallel with the interdependence of the different project elements. Finally, the established strategy must equally deal with the degree of structural uncertainty, which is the degree to which requirements are solidified, as well as the ease of functional decomposition. The study equally highlighted the need to watch out for "scope creep" which is a concept that occurs when customers change requirements mid-cycle. Putting all these together, the overarching aim of project planning is to guarantee that the final result is finished on time, within budget, and exudes quality.

The software engineers, software managers, and other stakeholders involved in the software project planning are usually trained in the software estimating and planning procedures which are applicable to their assigned tasks. Paulk *et al.* (1993) identified the three goals of software project planning to include: documenting software estimates used in planning and tracking the software project; planning and documenting software project activities and commitments; and stakeholders agree to their commitments related to the software project. Some of the commitments made in software project planning include: Designation of a project manager to be responsible for negotiating commitments and developing the project's software development plan and; The software project follows a written organizational policy for planning a software project. For software project planning to be most effective, it must be initiated at the early stages of the overall project planning process and in parallel with it.

Han *et al.* (2015) explained that project plan development typically involves activities with tasks; cost estimation and; schedule with a completion finish date. In addition, SPP typically involves Risk Management, which comprises the anticipation of potential problems, mitigating or avoiding the problems, and tracking existing and potential problems. SPP also involves incremental release process model providing periodic demonstrations, reaching short-term goals, and checking progress towards long-term goals. Three general software project planning approaches were equally identified to include past experience, standard guidelines, and support tools. It was observed that experienced managers depended on their past experiences as well as the experiences of successful managers in creating plans. Documentations of past completed projects are usually adopted as models for project plans. A number of studies including those of requirements Alba and Chicano (2007), Chang *et al.* (2001; 2008), and Xiao, Ao and Tang (2013) reiterated that software project planning among other things consists of establishing a worker-task schedule for a software project. It was emphasized that employee skills and remunerations should be considered in assigning them to project tasks on the basis of task requirements.

### 1.2. Software Project Tracking and Oversight:

Software Project Tracking and Oversight (SPTO) is aimed at establishing adequate visibility into the projects progress, enabling management to respond effectively when the performance of the software project diverges considerably from the software project plans. A software development plan, the output from the activities of the Software Project Planning KPA, is a prerequisite for Software Project Tracking and Oversight (Paulk *et al.*, 1993). The SPTO at the CMMI maturity Level 2 emphasizes tracking the project and taking necessary corrective actions. This is more or less of a reactionary approach on the part of management to the actual problems. This reactionary approach is much unlike the proactive approach of the Integrated Software Management KPA (Aregbesola, 2017b) at maturity level 3, a higher maturity level of the SEI CMMI maturity scale. This is because at maturity level 3, the software process of the project is completely defined with well-established relationship among the different activities, tasks and work products of the software project. SPTO involves tracking and reviewing the software achievements and outcomes against documented plans, estimations, and commitments, and making necessary adjustments based on the projects current accomplishments (Nalbant, 2004; Olson, Reizer and Over, 1994; Nair and Annamalai, 2013). Progress is typically determined by making comparisons between the values in the documented plan and the actual software cost, schedule, size, and effort at selected milestones and when selected software work products are completed (Paulk *et al.*, 1993; Hjalmarsson, 2013; Futrell, Shafer and Safer, 2002). Corrective actions are taken when it becomes obvious that the software project's plans are not being met. Such corrective actions may typically include adjusting the software development plan to reflect the actual accomplishments or taking steps to improve the project performance or making adjustments to the remaining tasks (Amid and Moradi, 2013; Alwin, 2001; Dreon, 2000; Paulk *et al.*, 1993; Olson *et al.* 1993).

The General Accounting Office (1999) in studying their organization observed that no software quality assurance group existed, and that there were therefore no reviews and/or audits of the activities and work products for SPTO. A weak performance was equally observed for a large portion of the KPAs associated with SPTO. As such, it was further explained that effective SPTO among other things should include: Designation of a project software manager who will be responsible for the project's software activities and outcomes; Establishment of a documented software development plan on the basis of which status monitoring and tracking of software activities are performed; Establishment and adherence to written organizational policy for managing projects; Periodic reviews of current project status and accomplishments against the software development plan; Tracking all risks associated with the software project; Explicit assignment of responsibility for software activities and work products; Keeping track of the sizes of the software work products or accompanying changes and making

necessary modifications as required and; Regular review of the SPTO activities with senior management.

As a result of its high level of relevance, SPTO as a process area is common to a majority of the software process improvement models and plays a significant role aimed at delivering quality products within budget and schedule. The work of Baruah, Ashima and Barbaruah (2013) discussed various software process models developed to help different software organization compete favorably in the changing market environment and produce quality product. The focus was on exploring the methods commonly employed in SPTO in different small and medium scale enterprises in India. The different software models that were studied included the Capability Maturity Model CMM (Paulk *et al.*, 1993), CMM Integration (CMMI Product Team, 2006), Personal Software Process PSP (Humphrey, 2005), Team Software Process TSP (Davis and McHale, 2003), Six Sigma (Brue and Launsby, 2003; Eckes, 2001), People Capability Maturity Model PCMM (Vakaslahti, 1997), ISO 9000 (Stelzer, Mellis and Herzwurm, 1997), ISO 9001:1994 (TickIT, 1995), ISO/IEC 12207 (Singh, 1996), SPICE (Rout and Terence, 1995), BOOTSTRAP (Kuvaja, 1994), IEEE 1220 (Doran, 2006), IDEAL model (Gremba and Myers, 1997), Process Improvement for Small to Medium Enterprises PRISMS (Allen, Ramachandran and Abushama, 2003), K-model (Hwang, 2010) and TRISO (Li, 2007).

Although the study of General Accounting Office (1999) revealed that the projects evaluated exhibited some level of SPTO practice strengths (such as the documentation of software development plans and the respective designation of responsibilities to a project software manager), the projects collectively had many weaknesses. One of the significant weaknesses identified was that none of the projects studied followed a written organizational policy for managing the software projects. This increases the risk of key tracking and oversight activities not being performed as effectively as required. As observed in the study, the increased risk was evident in the project manager's failing to: periodically review project status and accomplishments against the software development plan; track all risks associated with the software project; explicitly assign responsibilities to individuals for software activities and work products; keep track of the sizes of the software work products or accompanying changes and making necessary modifications as required, and regularly review the SPTO activities with senior management. To further buttress the points raised, Table 1 which is from the study of Lowe and Cox (1996) shows the assessment results from the SPTO KPA at some point in Hewlett-Packard's journey towards its attainment of the SEI CMM maturity Level 2, a level which it has however since surpassed. The results in Table 1 show a combined average of 54.2% for both full and partial performance of the listed practices.

**Table 1: Assessment results for the software project tracking and oversight key process area.**

Survey Questions	Percent of Survey Respondents		
	Fully	Partial	Not
Are the project’s planned activities and deliverables tracked (e.g., schedule, effort, and tests)?	43	24	33
Are the actual results compared to your estimates throughout the project?	10	19	71
If there is a variance, does someone take corrective action or explain why the variance has occurred?	14	38	48
Are changes to the activities, deliverables, and schedule discussed with the people who will be affected?	19	58	23
Does someone review the project results regularly with your section and lab managers?	18	18	64

Source: Lowe and Cox (1996)

**III. RESEARCH METHODOLOGY**

A survey on the level of performance of the different practices associated with the Software Project Planning (SPP) and the Software Project Tracking and Oversight (SPTO) key process areas was conducted. The research was performed on 30 software companies within the federal republic of Nigeria using an abridged version of the verified SEI Maturity Questionnaire (Zubrow *et al.*, 1994). The abridged version of the SEI Maturity Questionnaire was adopted as the research instrument for eliciting required information for the study. The questionnaire consisted of two major sections. The first sections comprised of questions regarding software process key practices within the organisation. The second section which was the response section consisted of four response options namely “Yes”, “No”, “NA” for Not Applicable and “DK” for Don’t Know. These four options were the possible responses available to each respondent with regards to the organizations performance of the respective key practices in the given section. A total of twenty six (26) (i.e. 86.67%) of the 30 selected companies eventually participated in the study.

Further studies, using the action research approach, was carried out on some of the selected companies to ascertain the veracity of the collected data. A direct observation and actual participation in the organizational software development activities were adopted as a means of getting firsthand information about the practices of some of the organizations, and thereafter reconciling such information with the collected data. Measurement of process-related phenomena was also performed. Print and electronic documentation were equally explored as sources of useful details about the companies and their operations. Both structured and unstructured interviews were also employed in the information elicitation process.

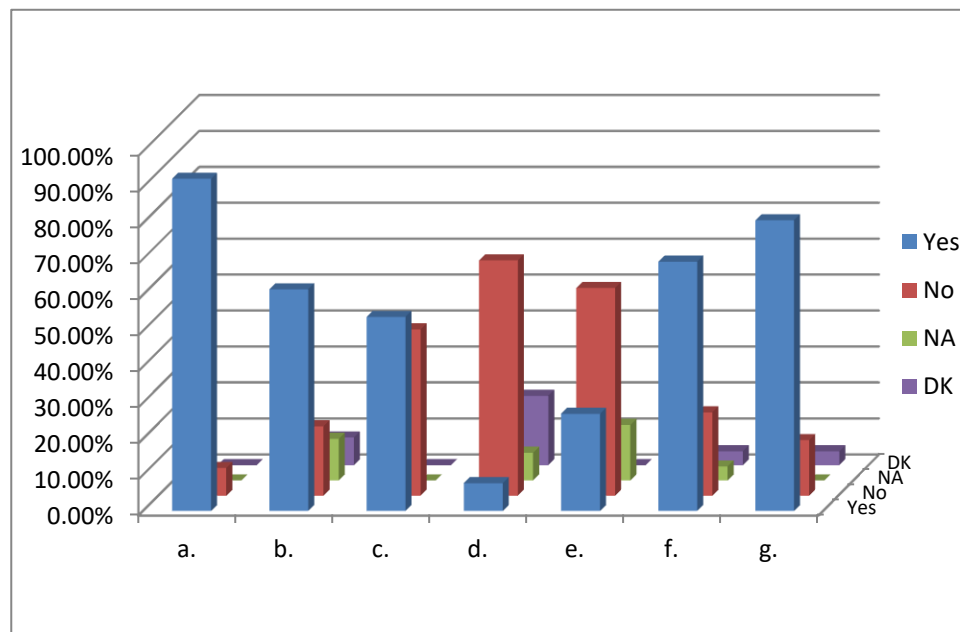
**IV. RESULTS**

The results of the current study are as shown in Tables 2 and 3, and equally graphically represented as depicted by Figures 1 and 2. The results are presented in percentages of actual responses. The averages for each response option are shown in bold at the last row of each table. Discussions and resultant conclusions from these results are presented in the subsequent sections.

**Table 2: Key Practices of the Software Project Planning (SPP) KPA**

Key Practices	Responses			
	Yes %	No %	NA %	DK %
a. Are estimates (e.g., size, cost, and schedule) documented for use in planning and tracking the software project?	92.31	7.69	0.00	0.00
b. Do the software plans document the activities to be performed and the commitments made for the software project?	61.54	19.23	11.54	7.69

c.	Do all affected groups and individuals agree to their commitments related to the software project?	53.85	46.15	0.00	0.00
d.	Does the project follow a written organizational policy for planning a software project?	7.69%	65.38	7.69	19.23
e.	Are adequate resources provided for planning the software project (e.g., funding and experienced individuals)?	26.92	57.69	15.38	0.00
f.	Are measurements used to determine the status of the activities for planning the software project (e.g., completion of milestones for the project planning activities as compared to the plan)?	69.23	23.08	3.85	3.85
g.	Does the project manager review the activities for planning the software project on both a periodic and event-driven basis?	80.77	15.38	0.00	3.85
<b>Average:</b>		<b>56.04</b>	<b>33.52</b>	<b>5.49</b>	<b>4.95</b>

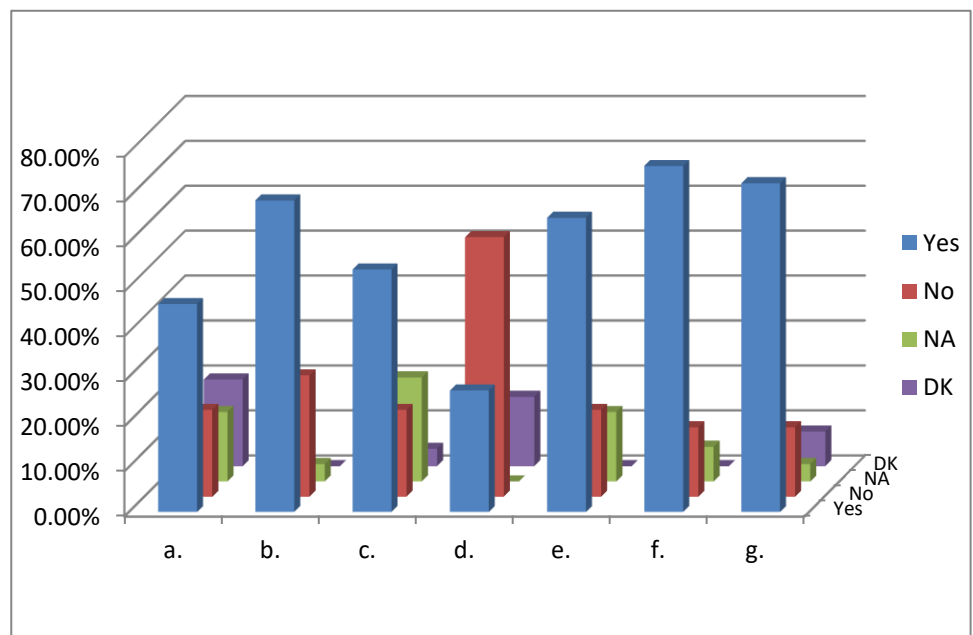


**Figure 1: Performance against Key Practices of the Software Project Planning (SPP) KPA**

**Table 3: Key Practices of the Software Project Tracking and Oversight (SPTO) KPA**

Key Practices	Responses			
	Yes %	No %	NA %	DK %
a. Are the project's actual results (e.g., schedule, size, and cost) compared with estimates in the software plans?	46.15	19.23	15.38	19.23
b. Is corrective action taken when actual results deviate significantly from the project's software plans?	69.23	26.92	3.85	0.00
c. Are changes in the software commitments agreed to by all affected groups and individuals?	53.85	19.23	23.08	3.85
d. Does the project follow a written organizational policy for both	26.92	57.69	0.00	15.38

tracking and controlling its software development activities?					
e.	Is someone on the project assigned specific responsibilities for tracking software work products and activities (e.g., effort, schedule, and budget)?	65.38	19.23	15.38	0.00
f.	Are measurements used to determine the status of the activities for software tracking and oversight (e.g., total effort expended in performing tracking and oversight activities)?	76.92	15.38	7.69	0.00
g.	Are the activities for software project tracking and oversight reviewed with senior management on a periodic basis (e.g., project performance, open issues, risks, and action items)?	73.08	15.38	3.85	7.69
<b>Average:</b>		<b>58.79</b>	<b>24.73</b>	<b>9.89</b>	<b>6.59</b>



**Figure 2: Performance against Key Practices of the Software Project Tracking and Oversight (SPTO) KPA**

V. DISCUSSION

The results shown in Tables 2 and 3, which are graphically depicted by the charts in Figures 1 and 2 show relatively high degrees of performance of the practices associated with the Software Project Planning (SPP) and Software Project Tracking and Oversight (SPTO) KPAs, with the performance of the practices associated with SPTO being somewhat better than the performance of the practices associated with SPP. These performances are quite remarkable considering that both KPAs are associated with software process CMMI maturity level 2 (Managed) while the Nigerian software industry has been said to be at maturity level 1 according to the studies of Aregbesola and Akinkunmi (2010a; 2010b), Aregbesola *et al.* (2011), Aregbesola and Onwudebelu (2011), and Aregbesola and Oluwade (2014). Of course, results from the studies never implied that there were no software companies within the country with higher maturity levels. Therefore, software companies within the country with higher maturity levels could have accounted for the high

performances experienced in the practices associated with both KPAs despite the industry average. In a similar study carried out by General Accounting Office (1999), it was equally revealed that the projects evaluated exhibited some level of strengths in their SPP and SPTO practices, despite the fact that the organizations were obvious yet to attain the CMMI maturity level 2.

The results obtained in the current study were equally somewhat similar to those of Lowe and Cox (1996) especially with regards to the performance of the practices associated with SPTO. While Lowe and Cox (1996) recorded a 54.2% performance of these practices, the current study recorded 58.8% performance of similar set of practices. The similarities in the results of both studies were quite consistent with expectations because the enterprise under study in Lowe and Cox (1996) were equally on their journey towards the SEI CMMI maturity level 2 at the time.

Establishing and adhering to written organizational policy for planning, tracking and controlling software development activities and projects were major weaknesses for most of the

organizations. Formal documentation of an organizational policy for SPP and SPTO was a major weakness across board. This observation was equally consistent with that of General Accounting Office (1999) which stated that one of the significant weaknesses identified in its study was that none of the projects studied followed a written organizational policy for managing the software projects. It was equally stated that this increased the risk of key tracking and oversight activities not being performed as effectively as required. It is therefore easy to see that although the performance of both KPA considered in this study can be said to be relative okay, the need still exists for significant improvement.

## VI. CONCLUSION

Considering that more than 40 percent of failed software projects have been said to be unsuccessful because of ineffective planning of human resources and project tasks, and that the reason for this was that unlike other projects, software project activities were people-intensive and that the related resources were mostly human resources; it therefore becomes imperative that training and retraining of the human resources is given utmost diligence. Also, considering that human resource allocation has been identified as a complex task, it behooves senior management to support the project manager in acquiring effective methodologies and software tools for managing resource allocation along SPP and SPTO, thereby maximizing cost and time.

Proper documentation of the entire software process and securing stakeholder's commitment have equally been identified as key factors that must be given kin consideration when it comes to SPP and SPTO. Besides the software project following a written organizational policy for SPP and SPTO, another commitment that is an absolute necessity for success in these KPAs is the designation of a project manager to be responsible for negotiating commitments and establishing the project's software development plan. The appointment of a project manager is an absolute necessity because he would be responsible for the majority of all the other commitments. For software project planning to be most effective, it must be initiated at the early stages of the overall project planning process and in parallel with it.

The current study focused on Software Project Planning (SPP) and Software Project Tracking and Oversight (SPTO). Existing works on the two KPAs were reviewed and research result on the level of performance of both KPAs in the Nigerian software industry was equally presented. By using survey and action research methods, it was shown that the performance of the practices associated with the two KPAs was relatively strong in the Nigerian software industry. There however still exists plenty of room for improvement in the performance of the practices in the KPAs. The industry should therefore endeavor not just to maintain the current performance level, but strive to bring about significant improvements. Considering the pivotal role SPP and SPTO KPAs play in the software development process, it is quite certain that an improvement in the performance of the practices associated with them will result in significant productivity enhancements and improved software process maturity.

## REFERENCES

- [1] Alba E. and Chicano F. (2007). "Software project management with gas", *Information Sciences*, vol. 177 no. 1, pp. 2380-2401.
- [2] Allen P., Ramachandran M. and Abushama H. (2003). "PRISMS: an Approach to Software Process Improvement for Small to Medium Enterprises", in *Proc. of the Third International Conference On Quality Software (QSIC'03)*, pp. 1-4.
- [3] Alwin L. F. (2001). *Texas Can Benefit From Using a Standard Framework to Manage Software Development. A Pilot Study Using the Capability Maturity Model for Software*. November. Report No. 02-008.
- [4] Amid A., Moradi S. (2013). *A Framework for Software Quality Assessment*. *Technical Journal of Engineering and Applied Sciences*.
- [5] Aregbesola M. K. (2017a). *Exploring Requirements Management with Software Subcontract and Configuration Management*. Manuscript submitted for publication.
- [6] Aregbesola M. K. (2017b). *Integrated Software Management and Product Engineering*. Manuscript submitted for publication. *International Journal of Scientific & Engineering Research (IJSER)*. In press.
- [7] Aregbesola M. K. (2017c). *Experiential Appraisal of Organizational Process Focus and Process Definition in Nigerian Software Companies*. *Journal of Scientific and Engineering Research*. In press.
- [8] Aregbesola M. K. (2017d). *Investigating Training Program and Intergroup Coordination in relation to Peer Review in Nigerian Software Companies*. *Journal of Scientific and Engineering Research*. In press.
- [9] Aregbesola M. K. and Oluwade B. A. *An Experimental Evaluation of Defect Prevention and Change Management in Software Process Optimization in the Nigerian Software Industry*. *ARNP Journal of Systems and Software Vol.4, No.1, 2014*, pp. 5-11.
- [10] Aregbesola M. Kehinde and B.O. Akinkunmi. (2010a). *Software Process Implementation – A focus on the Nigerian Software Industry*. *Journal of Research in Physical Sciences*, Vol. 6, No. 2, 2010, pp. 9 – 14.
- [11] Aregbesola M. Kehinde, Babatunde O. Akinkunmi and Olalekan S. Akinola. *Process Maturity Assessment of the Nigerian Software Industry*. *International Journal of Advances in Engineering and Technology (IAET)*, Vol.1, Issue 4, 2011, pp. 10-25.
- [12] Aregbesola M.K. and B.O. Akinkunmi. (2010b). *Software Process Implementation – A focus on the Nigerian Software Industry*. *International Research and Development Institute (IRDI), World Congress on Research and Development, Conference Center, University of Ibadan, 5th - 8th October 2010*. Vol. 5, No. 6, pg.111-116.
- [13] Aregbesola M.K. and Onwudebelu U.: *Typical Software Quality Assurance and Quality Management Issues in the Nigerian Software Industry*. *National Association for Science, Humanities & Education Research, 8th National Conference, University of Ado Ekiti, Ado Ekiti, September 14-17, 2011*.
- [14] Baruah N., Ashima, and Barbaruah K. (2013). *Software Project Tracking and Oversight and Its Different Measures*. *International Journal of Scientific and Research Publications*, Volume 3, Issue 9.
- [15] Brue G. and Launsby R. G. (2003). *Design for Six Sigma*, McGraw-Hill Professional.
- [16] Chang C. K., Hsin Yi J., Yu D., Dan Z. and Yujia G.(2008). "Time-line based model for software project scheduling with genetic algorithms", *Information and Software Technology*, vol. 50 no. 11, pp. 1142-1154.
- [17] Chang C.K., Christensen M.J., and Zhang T. (2001). "Genetic Algorithms for Project Management," *Annals of Software Eng.*, vol. 11, pp. 107- 139.
- [18] Chen W. N. and Zhang. J. (2013). "Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler," *IEEE Trans. Software Engineering*, Jan. vol. 39, no. 1, pp.1-17.
- [19] CMMI Product Team (2006): 'CMMI for Development, Version 1.2 - CMMI-DEV, V1.2', *Software Engineering Institute, Carnegie Mellon University*.
- [20] Davis N. and J. McHale,(2003). "Relating the Team Software ProcessSM(TSP SM) to the Capability Maturity Model ® for Software(SW-CMM®) With Strategy and Overview by Watts S. Humphrey ",*Carnegie Mellon Software Engineering Institute, Technical Report CMU/SEI-2002-TR-008 ESC-TR-2002-008*.

- [21] Ding R. G. and Jing X. H. (2003). "Five Principles of Project Management in Software Companies," Project Management Technology (in Chinese). vol. 1.
- [22] Doran T. (2006). "IEEE 1220: For Practical Systems Engineering", Computer, pp. 92-94. [ieeexplore.ieee.org/iel5/2/34216/01631953.pdf](http://ieeexplore.ieee.org/iel5/2/34216/01631953.pdf)
- [23] Dreon B. E. (2000). CMM Level 3: How Do I Get There from Here? Litton PRC. version 1.7.
- [24] Eckes G. (2001). The Six Sigma Revolution, How General Electric and Others Turned Process into Profits. USA: John Wiley & Sons, Inc.
- [25] Futrell R. T., Shafer D. F. and Safer L. I. (2002). Quality Software Project Management. Prentice Hall PTR. January 24, 2002.
- [26] General Accounting Office (1999). Customs Service Modernization: Ineffective Software Development Processes Increase Customs System Development Risks. United States General Accounting Office: Report to Congressional Requesters. GAO/AIMD-99-35. pp 1- 75.
- [27] Glazer, Hillel. (2001). Dispelling the Process Myth: Having a Process Does Not Mean Sacrificing Agility or Creativity. CROSS TALK: The Journal of Defense Software Engineering.
- [28] Glover, Margaret Tanner and Dennie, Debra. (2017). CMMI–Agile Process Combo: How to be Agile with CMMI. Excellence in Measurement Technology.
- [29] Gremba J. and Myers C. (1997). The IDEAL(SM) Model: A Practical Guide for Improvement. US:SEI, Carnegie Mellon University.
- [30] Han W., Jiang H., Lu T., Zhang X. and Li W. (2015). An Optimized Resolution for Software Project Planning with Improved Max–Min Ant System Algorithm. International Journal of Multimedia and Ubiquitous Engineering. Vol.10, No.6.
- [31] Hjalmarsson A. (2013). Software Development Cost Estimation Using COCOMO II Based Meta Model. Master thesis, ICS, KTH Electrical Engineering, Stockholm, Sweden.
- [32] Humphrey S. W. (2005). "The Personal Software Process: Status and Trends", IEEE Software, vol.17, no.6, pp.71-75.
- [33] Hurst, Jim. (2017). The Capability Maturity Model and Its Applications. SANS Software Security with Frank Kim.
- [34] Hwang S. M. (2010). "Quality Metrics for Software Process Certification based on K-Model" in Proc. IEEE 24th International Conference on Advanced Information. Networking and Applications Workshops, pp. 827-830.
- [35] Kurien V., Nair R. S. (2014). Software Project Planning and Resource Allocation Using Ant Colony Optimization with Uncertainty Handling. International Journal of Innovative Research in Science, Engineering and Technology. July, Volume 3, Special Issue 5.
- [36] Kuvaja P. (1994). Software Process Assessment and Improvement: the BOOTSTRAP Approach, UK: Blackwell Publishers, Oxford.
- [37] Li M. (2007). "TRISO-Model: A New Approach to Integrated Software Process Assessment and Improvement", Software Process Improvement and Practice, vol.12, issue.5, pp.387-398.
- [38] Lowe D. E. and Cox G. M. (1996). Implementing the Capability Maturity Model for Software Development. Hewlett-Packard Journal. August. pp 1-11.
- [39] Mogre, Anjali, Salunkhe, Sujata. (2014). Effective CMMi Implementation in Agile environment with fresh team. Atos, Mumbai, India.
- [40] Nair P. R. and Annamalai P. (2013). Comparison between ISO 9000 and CMM Software Quality Standards. International Journal of Engineering, Business and Enterprise Applications (IJEBA). pp 10 - 14.
- [41] Nalbant S. (2004). An Information System for Streamlining Software Development Process. MİSOFT Yazılım Teknolojileri A.S. ODTU-Teknokent İkizleri 06531, Ankara-TURKEY. pp 1 - 10.
- [42] Nan N. and Harter D. E. (2009). "Impact of Budget and Schedule Pressure on Software Development Cycle Time and Effort," IEEE Trans. Software Eng. Sept./Oct. vol. 35, no. 5, pp. 624-637.
- [43] O'Neill Don. (2017). In Search of a Modern Software Life Cycle: Secure DevOps Foundations for Large-Scale Software Systems. CrossTalk March/April 2017: Modern Process Trends.
- [44] Olson T. G., Linda Parker Gates, Julia L. Mullaney, James W. Over, Neal R. Reizer, Mark I. Kellner, Richard W. Phillips, Salvatore DiGennaro J. (1993). A Software Process Framework for the SEI Capability Maturity Model: Repeatable Level. Software Engineering Institutes, Carnegie Mellon University. Special Report CMU/SEI-93-SR-007. Report. Pittsburgh, Pennsylvania 15213-3890
- [45] Olson T. G., Reizer N. R. and Over J. W. (1994). A Software Process Framework for the SEI Capability Maturity Model. CMU/SEI-94-HB-01 Handbook, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213.
- [46] Paulk M. C., Charles V. Weber, Bill Curtis, & Mary Beth Chrissis (1995): The Capability Maturity Model: Guidelines for Improving the Software Process. Addison – Wesley, Boston, 1995
- [47] Paulk M. C., Weber C. V., Garcia S. M., Chrissis M. B., and Bush M. (1993). "Key Practices of the Capability Maturity ModelSM, Version 1.1". Technical Report CMU/SEI-93-TR-025 ESC-TR-93-178, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania. 1993.
- [48] Paulk M.C., B. Curtis, M.B. Chrissis, and C.V. Weber. (1993). "Capability Maturity Model, Version 1.1," IEEE Software, vol. 10, no. 4, pp. 18-27.
- [49] Pikkariainen, Minna. (2008). Towards a Framework for Improving Software Development Process Mediated with CMMI Goals and Agile Practices. Academic dissertation at the Department of Information Processing Science, University of Oulu, Ostrobothnia, Finland.
- [50] Rosso-Llopert M. (2005). Project planning and scheduling. Carnegie Mellon.
- [51] Rout and Terence P. (1995). "SPICE: A Framework for Software Process Assessment," Software Process Improvement and Practice, vol. 1, no. 1, pp. 57-66.
- [52] Shtub A., Bard J. F., and Globerson S. (2005). Project Management: Processes, Methodologies, and Economics, second ed. Prentice Hall.
- [53] Singh R. (1996). "International Standard ISO/IEC 12207 Software Life Cycle Processes", Software Process Improvement and Practice, vol. 2, no. 1, pp.35-50.
- [54] Soriyan H. A. and Heeks R. (2004): A Profile of Nigeria's Software Industry. Development Informatics Working Paper No 21, Institute for Development Policy and Management, University of Manchester. 2004.
- [55] Soriyan H. A., Mursu A. and Korpela M. (2000): 'Information system development methodologies: gender issues in a developing economy', In: Women, Work and Computerization, E. Balka & R. Smith (eds.), Kluwer Academic, Boston, MA, 146-154.
- [56] Stelzer D., Mellis W. and Herzwurm G. (1997). "A critical look at ISO 9000 for software quality management," Software Quality Journal, vol. 6, no. 2, pp. 65–79.
- [57] TickIT (1995). TickIT Guide to Software Quality Management System Construction and Certification using ISO 9001:1994. Issue 1.0, DISC TickIT Office.
- [58] Vakaslahti P., (1997). "Process Improvement Frameworks- a Small Case Study with People Capability Maturity Model", Software Process Improvement and Practice, vol. 3, no.4, December, pp.225-234.
- [59] Veldman, J. (2011). Process improvement for engineering & maintenance contractors. University of Groningen, SOM research school, Groningen.
- [60] Xiao J., Ao X. T., and Tang Y. (2013). "Solving software project scheduling problems with ant colony optimization", Computers and Operations Research, vol. 40 no. 1, pp. 33–46.
- [61] Zubrow D., William H., Jane S. and Dennis G. (1994): 'Maturity Questionnaire'. Special Report CMU/SEI-94-SR-7, June 1994.
- [62] Akinola S. O., Osofisan A. O., Akinkunmi B. O. (2009). "Industry Perception of the Software Inspection Process: Nigeria Software Industry as a Case Study", African Journal of Computing & ICT (Journal of the IEEE Nigeria Computer Chapter), Vol. 2, No. 2, pp. 3-11.
- [63] Akinola S. O., Osofisan A. O. (2011). "Evaluating Defect Detection Techniques in a PaperBased Environment – A Multi-Trial Experiment Report", African Journal of Computing & ICT (Journal of the IEEE Nigeria Computer Chapter), Vol. 2, No. 2, pp. 16-30.

#### AUTHORS

**First Author** – Aregbesola, Moses Kehinde; Department of Mathematics and Computer Science, Elizade University, Ilara-Mokin, Ondo State, Nigeria, [kehinde.aregbesola@elizadeuniversity.edu.ng](mailto:kehinde.aregbesola@elizadeuniversity.edu.ng)