PROCESS MATURITY ASSESSMENT OF THE NIGERIAN SOFTWARE INDUSTRY

Kehinde Aregbesola[1], Babatunde O. Akinkunmi[2], Olalekan S. Akinola[3]
[1]Salem University, Lokoja, Kogi State, Nigeria.
[2&3]Department of Computer Science, University of Ibadan, Ibadan, Nigeria.

ABSTRACT

Capability Maturity Model Integration (CMMI) is a recognized tool for performing software process maturity and capability evaluation in software organizations. Experience with software companies in Nigeria shows that most project management activities do not follow the conventional practices. The study considered the extent to which companies make use of organizational software process in performing their software development activities. The extent to which software products are developed and documented as well as level of adherence to existing organizational software process were studied among Twenty-six (26) selected software companies in Nigeria. The selection criteria were based on: availability of personnel to provide adequate information; size of the development team; how established the companies are; and geographical distribution. Our study revealed that the software companies do not have adequate documentation of their organizational software process, and that most of the companies carry out their software development process by means of implicit in-house methods.

KEYWORDS: Software Process, Software Industry, CMMI, Nigeria

I. INTRODUCTION

Success in software development is expected to be repeatable if the team involved is to be described as dependable. Dependability in software development can only be achieved through rigorous software development processes and project management practices. Understanding organizational goals and aspirations, is always the first step in making progress of any kind.

This study focuses on knowing the current state of software process maturity level of the Nigerian software industry. Nigeria is a strategic market for application software in the African continent. The Nigerian software industry has a strategic influence in West Africa. The bulk of the Nigerian software industry is located in the commercial capital of Lagos. According to the 2004 study by Soriyan and Heeks [13, 14], Lagos, which is widely regarded as Nigeria's "economic capital", accounts for 52 software companies representing about 49 percent of the software companies in Nigeria.

The study was conducted to determine the Capability and Maturity levels of the Nigerian software industry using the CMMI Model. The specific objectives of the study are listed below:

- Survey the software practices adopted by a good number of software companies;

- Apply the SEI Maturity Questionnaire to further gather data;

- Properly summarize and document the data collected;

- Evaluate the practices in the industry based on key process areas.

- Apply CMMI methods to determine the maturity and capability levels of the industry.

The rest of the paper is organized as follows. Section 2 reviews some literatures related to this work. Section 3 discusses the approach applied in performing the study. Section 4 discusses the findings of the study. Section five summarizes the conclusions drawn from the study.

II. LITERATURE REVIEW

Heyworth [5] described the characteristics of projects to include bringing about a change of state in entities of concern within well planned time frames. This indicates a strong relationship between projects and processes.

A prior study comparing CMMI appraisals for different countries have been reported by Urtans [6]. The study revealed the observed trends in CMM to include the following:

- Higher maturity levels seen mostly outside the USA
- India is the leader in CMM
- China and Korea are emerging as outsourcing centers
  - Increasing number of high maturity companies
- Canada, Ireland, Australia considered for outsourcing due to native English
  - Starting to report lower levels of CMM
- The number of companies each year using CMM to assess their software management practices more than doubles every five years

According to Heeks [7, 8], production of software provides many potential benefits for developing countries, including creation of jobs, skills and income. According to him also, selling software services to the domestic market is the choice of most developing countries software enterprises, but it typically represents a survival strategy more than a development strategy. He further iterated that most information systems - including current ICT projects - in developing countries fail either totally or partially due to the notion he described as design-reality gaps.

Soriyan and Heeks [13] gave a very descriptive view of the Nigerian software industry. According to them, 43.7% of the companies had 1-5 IT professionals, 27.2% had 6-15, 23.3% had 16-50, and only 5.8% of firms had more than 50 IT professionals. Also, 51% of the companies were involved with servicing imported applications, 25% were involved with Developing and servicing local applications, while 24% were involved with servicing and developing local and imported applications. This basically reveals that most of the software companies in the industry are small, and not as much attention as expected is given to developing and servicing local applications. Virtually no attention is given to the development of software tool. Also, their work revealed that Nigerian software industry showed significant use of formal methods but with a strong tendency to rely on in-house-developed methods rather than industry standards.

The work of Paulk et al [9, 10] produced the Maturity Questionnaire (MQ) which formed the major instrument of information elicitation during the course of the study discussed in this paper. According to Ahern et al [1], Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisals can help organizations identify the strengths and weaknesses of their current processes, reveal crucial development and acquisition risks, set priorities for improvement plans, derive capability and maturity level ratings, and even perform realistic benchmarking.

For this study we used the maturity questionnaire for eliciting information from surveyed companies, while

2.1. The Capability Maturity Model Integration (CMMI)

CMMI is a model for evaluating the maturity of software development process. It was developed from CMM**.** CMMI stands for Capability Maturity Model Integration. It is a method to evaluate and measure the maturity of the software development process of an organization. It measures the maturity of the software development process on a scale of 1 to 5. It was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University in Pittsburgh, USA [3, 12].

2.2. Maturity Level

A maturity level can be said to be a well-defined evolutionary plateau toward achieving a mature software process. Each maturity level provides a layer in the foundation for continuous process improvement. In CMMI models, there are five maturity levels designated by the numbers 1 through 5.
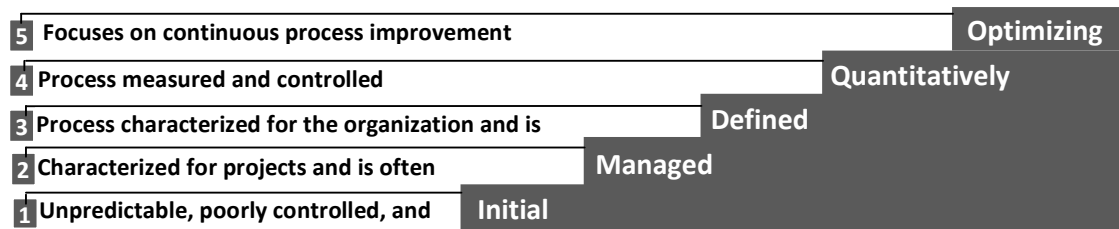
5 Focuses on continuous process improvement			**Optimizing**
4 Process measured and controlled		**Quantitatively**	
3 Process characterized for the organization and is	**Defined**		
2 Characterized for projects and is often	**Managed**		
1 Unpredictable, poorly controlled, and	**Initial**		

Fig. 1: The Five Levels of CMMI [3, 12]

Maturity levels consist of a predefined set of process areas. The maturity levels are measured by the achievement of the specific and generic goals that apply to each predefined set of process areas. The following sections describe the characteristics of organizations at each maturity level.

Maturity Level 1 – Initial: Processes are usually ad hoc and chaotic. They do not provide stable work environment. Success depends on the competence and heroics of the people in the organization and not on the use of proven processes.

Maturity Level 2 – Managed: The projects of the organization have ensured that requirements are managed and that processes are planned performed, measured, and controlled. They ensure that existing practices are retained during times of stress.

Maturity Level 3 – Defined: Processes are well characterized and understood, and are described in standards, procedures, tools, and methods.

Maturity Level 4 - Quantitatively Managed: At maturity level 4 sub-processes are selected that significantly contribute to overall process performance. These selected sub-processes are controlled using statistical and other quantitative techniques.

Maturity Level 5 – Optimizing: Processes are continually improved based on a quantitative understanding of the common causes of variation inherent in processes. Maturity level 5 focuses on continually improving process performance.
Maturity levels should not be skipped. Each maturity level provides a necessary foundation for effective implementation of processes at the next level.
- Higher level processes have less chance of success without the discipline provided by lower levels.
- The effect of innovation can be obscured in a noisy process.

Higher maturity level processes may be performed by organizations at lower maturity levels, with the risk of not being consistently applied in a crisis [3].

2.3. Capability Level

A capability level is a well-defined evolutionary plateau describing the organization's capability relative to a process area. Capability levels are cumulative, i.e., a higher capability level includes the attributes of the lower levels. In CMMI models with a continuous representation, there are six capability levels designated by the numbers 0 through 5.

Capability Level 0 – Incomplete: An "incomplete process" is a process that is either not performed or partially performed. One or more of the specific goals of the process area are not satisfied and no generic goals exist for this level.

Capability Level 1 – Performed: This is a process that is expected to perform all of the Capability Level 1 specific and generic practices. Performance may not be stable and may not meet specific

objectives such as quality, and cost, but useful work can be done. It means that you are doing something but you cannot prove that it really works for you.

Capability Level 2 – Managed: A managed process is planned, performed, monitored, and controlled for individual projects, groups, or stand-alone processes to achieve a given purpose. Managing the process achieves both the model objectives for the process as well as other objectives, such as cost, schedule, and quality.

Capability Level 3 – Defined: A defined process is a managed (capability level 2) process that is tailored from the organization's set of standard processes according to the organization's tailoring guidelines, and contributes work products, measures, and other process-improvement information to the organizational process assets.

Capability Level 4 – Quantitatively Managed: A quantitatively managed process is a defined (capability level 3) process that is controlled using statistical and other quantitative techniques. Quantitative objectives for quality and process performance are established and used as criteria in managing the process.

Capability Level 5 – Optimizing: An optimizing process is a quantitatively managed process that is improved, based on an understanding of the common causes of process variation inherent in the process. It focuses on continually improving process performance through both incremental and innovative improvements [3].

Fusaro et al [11] did some work on the reliability test of the SEI MQ. According to them, the Spearman-Brown formula was used to make all of the reliability estimates applicable to instruments of equal lengths. During their study, a point was noted where all of the internal consistency values for full length instruments were above the 0.9 minimal threshold. For this reason, the full length instrument was therefore considered to be internally consistent for practical purposes.

III.   RESEARCH DESIGN, METHODOLOGY AND APPROACH

This study was aimed at assessing software process maturity in the Nigerian software industry. In this section, the methodology and approach we took in carrying out this study is outlined.

The purpose of this section is to:
- Discuss the research philosophy used in this work;
- Expound the research strategy adopted in this work, including the research methodologies adopted;
- Introduce the research instruments we adopted in the carrying out the research.

Two major research methodologies were applied in performing this study. These methodologies are survey research and case study research methodologies.

Survey Research: According to our research objectives, we surveyed the software practices adopted by many of the Nigerian software companies. For this study 30 Nigerian software companies were studied. 27 of those companies were based in Lagos southwestern Nigeria, while three were based in Asaba, south-southern Nigeria. The sampling method is stratified in the sense that the majority of Nigeria's software companies were based in Lagos.

An instrument – the SEI Maturity Questionnaire (MQ) – was used to gather information about software process implementation within the companies covered. This instrument was administered to solutions developers and software project managers in the industry. This instrument served as the key data collection tool for the survey.

Case Study Research: Some of the companies were taken as case studies for more detailed investigation. A direct observation of their activities and environment was carried out. Indirect observation and measurement of process related phenomena was also performed. The companies involved were visited and observed over a period of time to see how they actually implement their software development process. Both structured and unstructured interviews were also used to solicit information. Documentation, such as written, printed and electronic information about the company and its operations were another method by which information was gathered.

In order to analyze the current situation in the Nigerian software industry, it is essential to have a validated and reliable instrument for the collection of the information required. For this reason, the SEI Maturity Questionnaire was adopted.

3.1 The Software Process SEI Maturity Questionnaire (MQ)

The software process maturity questionnaire (MQ) replaces the 1987 version of the maturity questionnaire, CMU/SEI-87-TR-23, in the 1994 set of SEI appraisal products. This version of the questionnaire is based on the capability maturity model (CMM) v1.1. It has been designed for use in the new CMM-based software process appraisal methods: the CMM-based appraisal for internal process improvement (CBA IPI) which is the update of the original software process assessment (SPA) method, CMM-based software capability evaluations (SCEs), and the interim profile method. The questionnaire focuses solely on process issues, specifically those derived from the CMM. The questionnaire is organized by CMM key process areas (KPAs) and covers all 18 KPAs of the CMM. It addresses each KPA goal in the CMM but not all of the key practices. By keeping the questions to only 6 to 8 per KPA, the questionnaire can usually be completed in one hour [4].

IV.   RESEARCH FINDINGS AND INTERPRETATION

In as much as the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) is an appraisal method that meets all of the Appraisal Requirements for CMMI (ARC), and currently the only SEI approved Class A appraisal method, it was used in appraising the industry.

4.1 Evaluation of Research Findings

Out of the 30 companies surveyed, only responses from 26 companies were found useful. Responses from four companies were either inconsistent or could not be verified. As such, the evaluation of the companies was based on responses from 26 companies. 23 of these were based in Lagos, while three were based in Asaba.
In order to meet the objective of this study, the key practices were organized according to key process areas (labeled in Roman numerals). The key process areas were organized according to maturity level. Only the result for maturity level 2 is discussed this section. This is because an evaluation of the key practices at maturity level 2 suffices to arrive at a conclusion as to which maturity level the Nigerian software industry belongs.
To appraise an organization using the Standard CMMI Appraisal Method for Process Improvement (SCAMPI), the organization (Industry) is considered to have reached a particular level of maturity when it has met with all of the objectives/practices within each of the key process areas from maturity level 2 to the maturity level in question. This work shall therefore progress in that order, starting with the appraisal of the key process areas and practices found within maturity level2, until a point is reached where all the objectives/practices associated with a particular KPA are not met.
In the instrument that was administered, "Yes" connotes that the organizations perform the specified practice, while "No" means that the organization does not perform the specified practice. In the summary tables found in this section of the work:
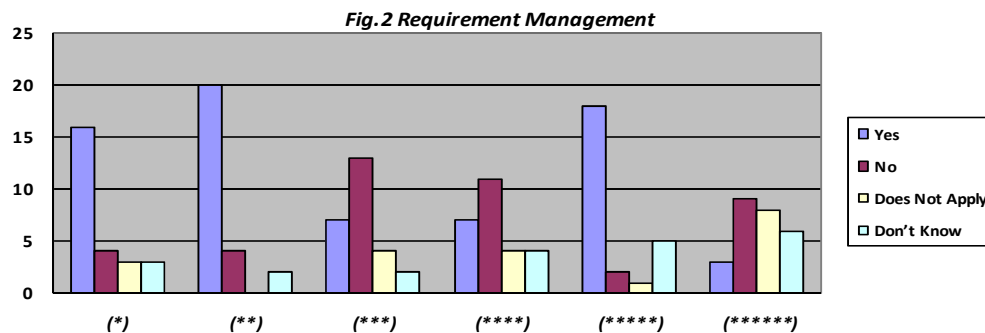
> ➢ The "Yes" column indicates the number of companies that perform the specified practice;

> ➢ The "No" column indicates the number of companies that do not perform the specified practice;

> ➢ Both the "Does Not Apply" and the "Don't Know" column values are used in the appraisal to indicate the amount of organizational unawareness in the industry;

> ➢ Percentage values are recomputed for the number of explicit ("yes" or "no") responses gathered, and would be used as a major appraisal factor.

4.2 Evaluation of the Results Obtained for Maturity Level 2 (Managed)

4.2.1. Requirement Management

Table 1: Requirement Management

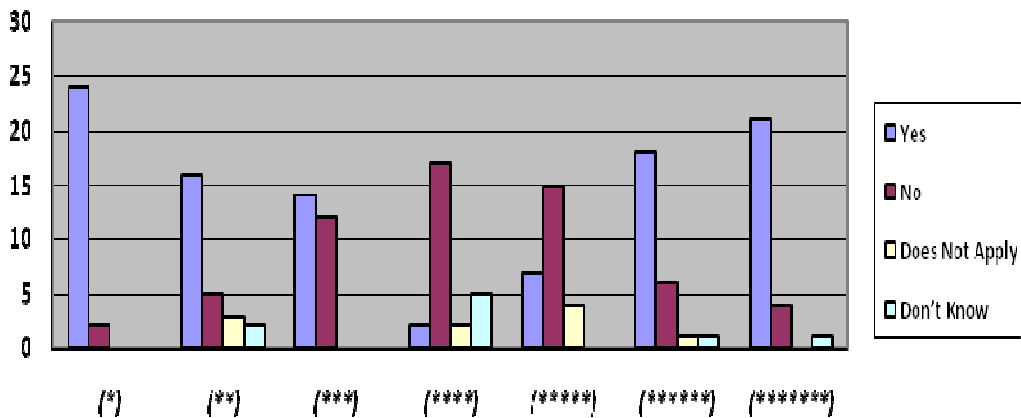|	**I** **Requirement Management**				
	QUESTIONS (Key Practices)	Yes	No	Does Not Apply	Don't Know
1	Are system requirements allocated to software used to establish a baseline for software engineering and management use? – *(*)*	16	4	3	3
2	As the systems requirements allocated to software change, are the necessary adjustments to software plans, work products, and activities made? – *(**)*	20	4	0	2
3	Does the project follow a written organizational policy for managing the system requirements allocated to software? – *(***)*	7	13	4	2
4	Are the people in the project that are charged with managing the allocated requirements trained in the procedures for managing allocated requirements? – *(****)*	7	11	4	4
5	Are measurements used to determine the status of the activities performed for managing the allocated requirements (e.g., total number of requirements changes that are proposed, open, approved, and incorporated into the baseline). – *(*****)*	18	2	1	5
6	Are the activities for managing allocated requirements on the project subjected to SQA review? – *(******)*	3	9	8	6
		45.5%	27.6%	12.8%	14.1%

Fig.2 Requirement Management

From the table above, out of the total number of people who responded explicitly as either "Yes" or "No", there was a 62.3% bias for the performance of requirement management associated practices, while 37.7% bias holds for non performance of requirement management associated practices. Basically, since industry wide, the "Yes" column contains values greater than zero; it means that at least one company performs one or more of the practices associated with the requirement management key process area.

4.2.2. Software Project Planning

Table 2: Software Project Planning

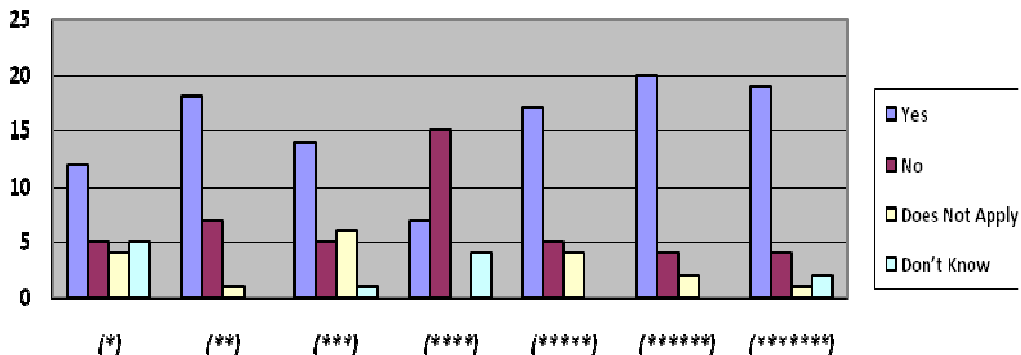|	II **Software Project Planning**				
	QUESTIONS (Key Practices)	Yes	No	Does Not Apply	Don't Know
1	Are estimates (e.g., size, cost, and schedule) documented for use in planning and tracking the software project? – *(*)*	24	2	0	0
2	Do the software plans document the activities to be performed and the commitments made for the software project? – *(**)*	16	5	3	2
3	Do all affected groups and individuals agree to their commitments related to the software project? – *(***)*	14	12	0	0
4	Does the project follow a written organizational policy for planning a software project? – *(****)*	2	17	2	5
5	Are adequate resources provided for planning the software project (e.g., funding and experienced individuals)? – *(*****)*	7	15	4	0
6	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)? – *(******)*	18	6	1	1
7	Does the project manager review the activities for planning the software project on both a periodic and event-driven basis? – *(*******)*	21	4	0	1
		56.0%	33.5%	5.5%	4.9%

Fig.3 Software Project Planning

From the table above, out of the total number of people who responded explicitly as either "Yes" or "No", there was a 62.6% bias for the software project planning associated practices, while a 37.4% bias holds for non performance of software project planning associated practices. Basically, since industry wide, the "Yes" column contains values greater than zero; it means that at least one company performs one or more of the practices associated with the software project planning key process area.

4.2.3. Software Project Tracking and Oversight

Table 3: Software Project Tracking and Oversight

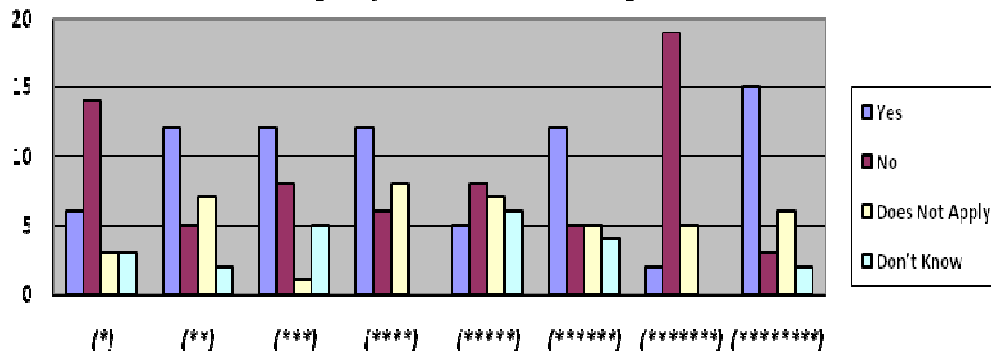|	III Software Project Tracking and Oversight				
	QUESTIONS (Key Practices)	Yes	No	Does Not Apply	Don't Know
1	Are the project's actual results (e.g., schedule, size, and cost) compared with estimates in the software plans? – *(*)*	12	5	4	5
2	Is corrective action taken when actual results deviate significantly from the project's software plans? – *(**)*	18	7	1	0
3	Are changes in the software commitments agreed to by all affected groups and individuals? – *(***)*	14	5	6	1
4	Does the project follow a written organizational policy for both tracking and controlling its software development activities? – *(****)*	7	15	0	4
5	Is someone on the project assigned specific responsibilities for tracking software work products and activities (e.g., effort, schedule, and budget)? – *(*****)*	17	5	4	0
6	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)? – *(******)*	20	4	2	0
7	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)? – *(*******)*	19	4	1	2
		58.8%	24.7%	9.9%	6.6%

Fig.4 Software Project Tracking and Oversight

From the table above, out of the total number of people who responded explicitly as either "Yes" or "No", there was a 70.4% bias for the software project tracking and oversight associated practices, while a 29.6% bias holds for non performance of software project tracking and oversight associated practices. Basically, since industry wide, the "Yes" column contains values greater than zero; it means that at least one company performs one or more of the practices associated with the software project tracking and oversight key process area.

4.2.4. Software Subcontract Management

Table 4: Software Subcontract Management

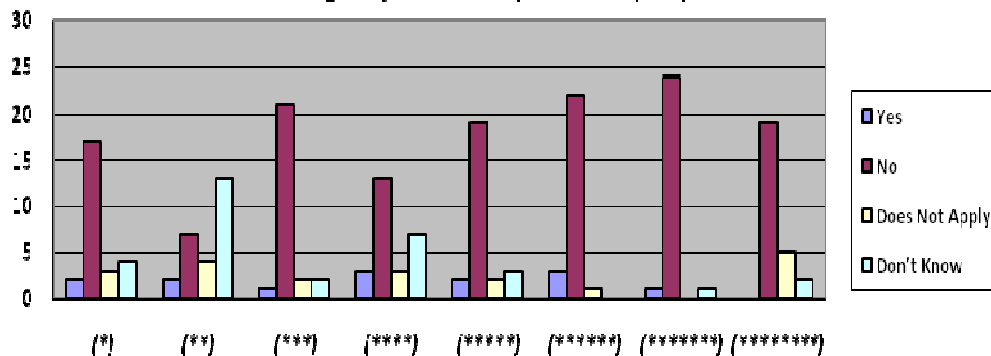|	IV Software Subcontract Management				
	QUESTIONS (Key Practices)	Yes	No	Does Not Apply	Don't Know
1	Is a documented procedure used for selecting subcontractors based on their ability to perform the work? – *(*)*	6	14	3	3
2	Are changes to subcontracts made with the agreement of both the prime contractor and the subcontractor? – *(**)*	12	5	7	2
3	Are periodic technical interchanges held with subcontractors? – *(***)*	12	8	1	5
4	Are the results and performance of the software subcontractor tracked against their commitments? – *(****)*	12	6	8	0
5	Does the project follow a written organizational policy for managing software subcontracts? – *(*****)*	5	8	7	6
6	Are the people responsible for managing software subcontracts trained in managing software subcontracts? – *(******)*	12	5	5	4
7	Are measurements used to determine the status of the activities for managing software subcontracts (e.g., schedule status with respect to planned delivery dates and effort expended for managing the subcontract)? – *(*******)*	2	19	5	0
8	Are the software subcontract activities reviewed with the project manager on both a periodic and event-driven basis? – *(********)*	15	3	6	2
		36.5%	32.7%	20.2%	10.6%

Fig.5 Software Subcontract Management

From the table above, out of the total number of people who responded explicitly as either "Yes" or "No", there was a **52.8%** bias for the software subcontract management associated practices, while a **47.2%** bias holds for non performance of software subcontract management associated practices. Basically, since industry wide, the "Yes" column contains values greater than zero; it means that at least one company performs one or more of the practices associated with the software subcontract management key process area.

4.2.5. Software Quality Assurance (SQA)

Table 5: Software Quality Assurance (SQA)

	V Software Quality Assurance (SQA)				
	QUESTIONS (Key Practices)	Yes	No	Does Not Apply	Don't Know
1	Are SQA activities planned? – *(*)*	2	17	3	4
2	Does SQA provide objective verification that software products and activities adhere to applicable standards, procedures, and requirements? – *(**)*	2	7	4	13
3	Are the results of SQA reviews and audits provided to affected groups and individuals (e.g., those who performed the work and those who are responsible for the work)? – *(***)*	1	21	2	2
4	Are issues of noncompliance that are not resolved within the software project addressed by senior management (e.g., deviations from applicable standards)? – *(****)*	3	13	3	7
5	Does the project follow a written organizational policy for implementing SQA? – *(*****)*	2	19	2	3
6	Are adequate resources provided for performing SQA activities (e.g., funding and a designated manager who will receive and act on software noncompliance items)? – *(******)*	3	22	1	0
7	Are measurements used to determine the cost and schedule status of the activities performed for SQA (e.g., work completed, effort and funds expended compared to the plan)? – *(*******)*	1	24	0	1
8	Are activities for SQA reviewed with senior management on a periodic basis? – *(********)*	0	19	5	2
		6.7%	68.3%	9.6%	15.4%

Fig.6 Software Quality Assurance (SQA)

From the table above, out of the total number of people who responded explicitly as either "Yes" or "No", there was a 9.0% bias for the software quality assurance associated practices, while a 91.0% bias holds for non performance of software quality assurance associated practices. Basically, since industry wide, the "Yes" column contains a zero value at some point; it means that no company performs one or more of the practices associated with the software quality assurance key process area.
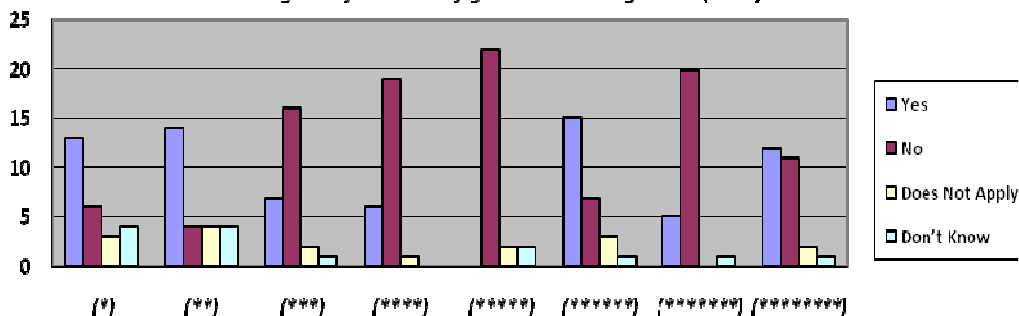
Industry wide, this is an explicit violation of the requirement for an industry to be at this current maturity level (2) under consideration.

4.2.6. Software Configuration Management (SCM)

Table 6: Software Configuration Management (SCM)

	QUESTIONS (Key Practices)	Yes	No	Does Not Apply	Don't Know
	VI Software Configuration Management (SCM)				
1	Are software configuration management activities planned for the project? – *(*)*	13	6	3	4
2	Has the project identified, controlled, and made available the software work products through the use of configuration management? – *(**)*	14	4	4	4
3	Does the project follow a documented procedure to control changes to configuration items/units? – *(***)*	7	16	2	1
4	Are standard reports on software baselines (e.g., software configuration control board minutes and change request summary and status reports) distributed to affected groups and individuals? – *(****)*	6	19	1	0
5	Does the project follow a written organizational policy for implementing software configuration management activities? – *(*****)*	0	22	2	2
6	Are project personnel trained to perform the software configuration management activities for which they are responsible? – *(******)*	15	7	3	1
7	Are measurements used to determine the status of activities for software configuration management (e.g., effort and funds expended for software configuration management activities)? – *(*******)*	5	20	0	1
8	Are periodic audits performed to verify that software baselines conform to the documentation that defines them (e.g., by the SCM group)? – *(********)*	12	11	2	1
		34.6%	50.5%	8.2%	6.7%

Fig.7 Software Configuration Management (SCM)

From the table above, out of the total number of people who responded explicitly as either "Yes" or "No", there was a 40.7% bias for the software configuration management associated practices, while a 59.3% bias holds for non performance of software configuration management associated practices. Basically, since industry wide, the "Yes" column contains a zero value at some point; it means that no company performs one or more of the practices associated with the software configuration management key process area. Industry wide, this is an explicit violation of the requirement for an industry to be at this current maturity level (2) under consideration.

V. RESULT AND DISCUSSION

The result of the study is expressed in terms of Software Process Maturity Assessment and Capability Assessment of the industry. The capability assessment is done based on individual KPAs while the maturity assessment is based on a specific collection of KPAs for each maturity level.

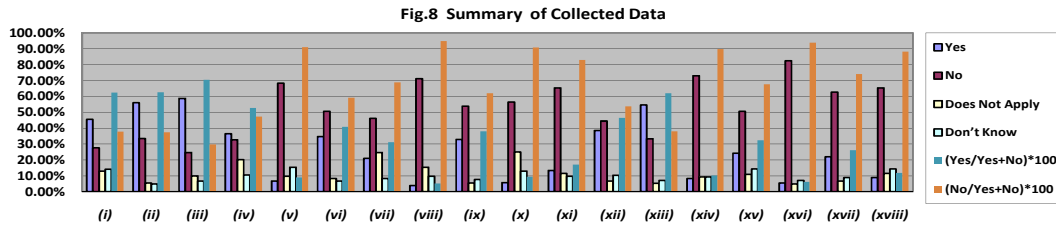### 5.1. Software Process Maturity Assessment

From the foregoing data in section 4, it can be deduced that due to the explicit violation of the requirement that at maturity level 2, an organization/industry has achieved all the specific and generic goals of the maturity level 2 process areas, it suffices to conclude that the Nigerian software industry does not belong to the SEI CMMI Maturity level 2. Hence, it suffices to conclude that the Nigerian software industry is at the SEI CMMI Maturity Level 1.

5.2. Key Process Area Capability Assessment

The project management practice in the Nigerian software industry was evaluated based on the key process areas identified by the adopted SEI Maturity Questionnaire. Table 7 below gives a high level summary of the data collected from the research. The percentage values for the number of explicit "yes" or explicit "no" responses gathered are shown in the columns "(Yes/Yes+No)*100" and "(No/Yes+ No)*100" respectively.

Table 7: Summary of Collected Data

S/N	Key Process Area (KPA)	Yes	No	Does Not Apply	Don't Know		(Yes/Yes+No)*100	(No/Yes+No)*100
1	Requirements Management – *(i)*	45.51%	27.56%	12.82%	14.10%		62.28%	37.72%
2	Software Project Planning – *(ii)*	56.04%	33.52%	5.49%	4.95%		62.58%	37.42%
3	Software Project Tracking and Oversight – *(iii)*	58.79%	24.73%	9.89%	6.59%		70.39%	29.61%
4	Software Subcontract Management – *(iv)*	36.54%	32.69%	20.19%	10.58%		52.78%	47.22%
5	Software Quality Assurance – *(v)*	6.73%	68.27%	9.62%	15.38%		8.97%	91.03%
6	Software Configuration Management – *(vi)*	34.62%	50.48%	8.17%	6.73%		40.68%	59.32%
7	Organization Process Focus – *(vii)*	20.88%	46.15%	24.73%	8.24%		31.15%	68.85%
8	Organization Process Definition – *(viii)*	3.85%	71.15%	15.38%	9.62%		5.13%	94.87%
9	Training Program – *(ix)*	32.97%	53.85%	5.49%	7.69%		37.97%	62.03%
10	Integrated Software Management – *(x)*	5.77%	56.41%	25.00%	12.82%		9.28%	90.72%
11	Software Product Engineering – *(xi)*	13.46%	65.38%	11.54%	9.62%		17.07%	82.93%
12	Intergroup Coordination – *(xii)*	38.46%	44.51%	6.59%	10.44%		46.36%	53.64%
13	Peer Reviews – *(xiii)*	54.49%	33.33%	5.13%	7.05%		62.04%	37.96%
14	Quantitative Process Management – *(xiv)*	8.24%	73.08%	9.34%	9.34%		10.14%	89.86%
15	Software Quality Management – *(xv)*	24.18%	50.55%	10.99%	14.29%		32.35%	67.65%
16	Defect Prevention – *(xvi)*	5.49%	82.42%	4.95%	7.14%		6.25%	93.75%
17	Technology Change Management – *(xvii)*	21.98%	62.64%	6.59%	8.79%		25.97%	74.03%
18	Process Change Management – *(xviii)*	8.79%	65.38%	11.54%	14.29%		11.85%	88.15%

Fig.8 Summary of Collected Data

The conclusions arrived at in the succeeding subsections are based on the data drawn from table 7 above.

5.2.1 Requirements Management (RM)
The Nigerian software industry performs requirement management practices to a good degree. The rudiments for basic requirement management are well carried out, even though it is nowhere near perfection at this point in time. The industry can still do with a whole lot of improvement, especially with requirement management quality assurance. The Requirement Management KPA can be said to be at the SEI CMMI Capability Level 1.

5.2.2 Software Project Planning (SPP)
The software project planning KPA is performed in almost the same degree as the Requirement Management KPA. There however seem to be very little organizational policy for planning software projects. The Software Project Planning KPA can also be said to at the SEI CMMI Capability Level 1.

5.2.3 Software Project Tracking and Oversight (SPTO)
Projects are actively tracked in the Nigerian software industry. The reason for this has been identified to be mainly due to cost management. SPTO can be said to be at the SEI CMMI Capability Level 1

5.2.4 Software Subcontract Management (SSM)
The Nigerian software industry does not involve so much in subcontracting activities. Most subcontracting activities performed are usually on a small scale. Not so much of written organizational policy exists for managing software subcontract, and the measures for managing software subcontracts are not well developed. The SSM KPA can be said to be at the SEI CMMI Capability Level 1.

5.2.5 Software Quality Assurance (SQA)
The performance of SQA activities are at the very minimum in the Nigerian software industry. Findings revealed that for most of the time, SQA activities are not planned, verified, reviewed, nor resolved. They do not follow written organizational policy, lack adequate funding, and lack adequate basis for measurement. SQA KPA can be said to be at the SEI CMMI Capability Level0.

5.2.6 Software Configuration Management (SCM)
The performance of SCM practices in the Nigerian software industry seems to be rather low. Organizational policies supporting SCM practices were difficult to come by. SCM KPA can be said to be at the SEI CMMI Capability Level 0.

5.2.7 Organization Process Focus (OPF)
Most software companies in Nigeria seem to focus too much on the product to be developed. They don't have time to work on the process required to build the product. The SPF KPA can be said to be at the SEI CMMI Capability Level0

5.2.8 Organization Process Definition (OPD)

Most software organizations in Nigeria have very poorly defined software process structure. Some don't even have at all. As expected, this would be at Capability Level 0.

5.2.9 Training Program (TP)

Even though some software organizations are intensive about staff training, the trend does not cut across board. Most pressing is the issue of most software organizations not having any written organizational policy to meet the training needs of its members of staff. This KPA is also at Capability Level 0.

5.2.10 Integrated Software Management (ISM)

Most software organizations do not have well defined organizational software process and therefore do not have a structure to pattern after. This KPA is also at the SEI CMMI Capability Level 0.

5.2.11 Software Product Engineering (SPE)

Most software companies in Nigeria do not involve in SPE practices. This KPA is at Capability Level 0.

5.2.12 Intergroup Coordination (IC)

Even though intergroup coordination seems to be relatively high in the industry, it is not even nearly as high and integrated into the system as it should be. IC KPA is at Capability Level 0.

5.2.13 Peer Reviews (PR)

Peer review practices seem to be actively carried out in software organizations in Nigeria. There is however still much gap to be filled. This KPA is at Capability Level 0.

5.2.14 Quantitative Process Management (SPM)

Quantitative process management seems to be unpopular with the software industry. This is mainly due to the total absence or lack of adequate organizational software process. It is at Capability Level 0.

5.2.15 Software Quality Management (SQM)

The practice of SQM practices in the Nigerian software industry does not seem to be so much on the high side. The seeming lack of written organizational policy calls for a lot of concern and craves for attention. This also falls under the SEI CMMI Capability Level 0.

5.2.16 Defect Prevention (DP)

As important as this KPA is, its practices are not more popular than a few others thus far mentioned. Adequate quality assurance and written organizational policies to support this KPA seem to be wanting. This KPA also falls under the SEI CMMI Capability Level 0.

5.2.17 Technology Change Management (TCM)

This KPA does not seem to be getting much attention. Most software organizations in Nigeria do not have any plan for managing technology changes. This KPA falls under the SEI CMMI Capability Level 0.

5.2.18 Process Change Management (PCM)

Just like most of the other process oriented KPA, the practices associated with PCM are not much favored by the lack of or inadequate organizational software process. Neither documented procedures nor written organizational policies seem to exist for supporting the PCM practices. Its capability level falls in the SEI CMMI Capability Level 0.

5.3. Discussion

Results from this study are in consonance with results from studies by other scholars. The study of Soriyan and Heeks [13, 14] shows that the Nigerian software industry is not so inclined to formal, well documented and standardized methodologies. The formalized methods used when there is any are usually developed in-house. According to Urtans [6], India, China, Japan, Korea, Australia, and Canada reported the highest number of appraisals and seem to have the highest maturity rankings. Besides these countries, most other countries are either on or fall below maturity level 3. Virtually all developing countries (to which Nigeria belongs) are in software maturity levels between 1 and 2. India happens to be one of the highest exporter of software and hence have software as one of its major sources of revenue [2, 6]. The Indian software industry attributed their success to strict adherence to the CMMI. The Nigerian software industry can experience the same monumental development following the same route other successful industries have been through.

VI.   CONCLUSION

To achieve the objective of this work, the Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI) for software process improvement was employed. The SEI Maturity Questionnaire (MQ) was the primary instrument used for eliciting data from respondents. Survey (using the MQ), and Case Study combined research methodologies were applied across thirty software organizations in Nigeria. The required data was successfully collected, verified, collated and evaluated. The Standard CMMI Appraisal Method for Process Improvement (SCAMPI) appraisal method was applied in the appraisal of the industry. The result of the appraisal was then summarized, indicating maturity level, capability levels, and project management practices based on the CMMI Key Process Areas (KPA).

The result revealed that the Nigerian software industry is very deficient in so many areas. This includes virtually all the Key Process Areas (KPA) in the SEI Maturity Questionnaire. The appraisal also revealed that the software process of the Nigerian software industry is at the maturity level 1, which is the very base level. While clamoring for a drastic improvement, this result should however not be so alarming as many industries in the world (even in developed countries) have not yet exceeded maturity level 2. The capability level for the identified key process areas were also identified to toggle between 0 and 1.

The scalability of the SEI CMMI model makes it adaptable to any kind and size of software development organization or industry. All that is required is the identification of a need to develop, grow, or mature the organizational software process. Once this need has truly been identified, the discipline required for climbing up the ladder of software process maturity will be imbibed.

ACKNOWLEDGEMENT

REFERENCES

[1]. Ahern, Dennis M.; Armstrong, Jim; Clouse, Aaron; Ferguson, Jack; Hayes, Will; Nidiffer, Kenneth (2005). 'CMMI SCAMPI Distilled: Appraisal for Process Improvement'.
[2]. Ajay Batra (2000), What Makes Indian Software Companies Thick? (CMM Practices in India)
[3]. CMMI Product Team (2006), 'CMMI for Development, Version 1.2 - CMMI-DEV, V1.2', Software Engineering Institute, Carnegie Mellon University.
[4]. David Zubrow, William Hayes, Jane Siegel, & Dennis Goldenson (1994) 'Maturity Questionnaire'.
[5]. Frank Heyworth (2002), 'A Guide to Project Management'. European Centre for Modern Languages, Council of European Publishing.
[6]. Guntis Urtans (2004) 'SW-CMM Implementation: Mandatory or Best Practice?', GM Eastern Europe, Exigen Group.
[7]. Heeks, R.B. (1999) Software strategies in developing countries, *Communications of the ACM*, 42(6), 15-20

[8]. Heeks, R.B. (2002) i-Development not e-development, *Journal of International Development*, 14(1): 1-12

[9]. Mark C. Paulk, Charles V. Weber, Bill Curtis, & Mary Beth Chrissis (1995) *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison – Wesley, Boston,1995

[10]. Mark C. Paulk, Charles V. Weber, Suzanne M. Garcia, Mary Beth Chrissis, Marilyn Bush, (1993), 'Key Practices of the Capability Maturity Model', Software Engineering Institute, Carnegie Mellon University CMU/SEI-93- TR-25, Pittsburgh, 1993.

[11]. Pierfrancesco Fusaro, Khaled El Emam, & Bob Smith (1997) 'The Internal Consistencies of the 1987 SEI Maturity Questionnaire and the SPICE Capability Dimension'. *Empirical Software Engineering: An International Journal*, 3(2): 179 -201.

[12]. SCAMPI Upgrade Team (2006), 'Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.2: Method Definition Document', CMU-SEI-2006-HB-002 Software Engineering Institute, Carnegie Mellon University, 2006.

[13]. Soriyan Abimbola & Richard Heeks (2004), *A Profile of Nigeria's Software Industry*. Development Informatics Working Paper No 21, Institute for Development Policy and Management, University of Manchester, 2004.

[14]. Soriyan, H.A., Mursu, A. & 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

Biography

Kehinde Aregbesola had his secondary education at Lagelu Grammar School, Agugu, Ibadan, Nigeria, where he was the Senior Prefect. He obtained his first and second degrees in Computer Science from the prestigious University of Ibadan (a former college of the University of London). He is an experienced solutions developer with several years in the industry. He has been involved in the development of diverse kinds of applications currently in use in different organizations, as well as a few tools currently in use by other software developers. He has implemented projects with a few prominent ICT companies including LITTC, Microsolutions Technology, Farsight Consultancy Services, Chrome Technologies, infoworks, etc. His focus is to be a pure blend of academic excellence and industrial resourcefulness. He is a member of the Computer Professionals of Nigeria (CPN), Nigeria Computer Society (NCS), and Nigerian Institute of Management (NIM), a certified manager of both human and material resources. He is currently a Lecturer at Salem University, Lokoja, Kogi State, Nigeria.

Babatunde Opeoluwa Akinkunmi is a member of the academic staff at the Dept of Computer Science University of Ibadan. He has authored over twenty five research articles in computer science. His research interests include Knowledge Representation, Formal Ontologies and Software Engineering.

Olalekan S. Akinola is currently a lecturer of Computer Science at the University of Ibadan, Nigeria. He had his PhD Degree in Software Engineering from the same University in Nigeria. He is currently working on Software Process Improvement models for the Nigeria software industry.