Hybrid Spiral Model to Improve Software Quality Using Knowledge Management

BRIJENDRA SINGH AND SHIKHA GAUTAM

ICT Research Lab, Department of Computer Science
University of Lucknow, Lucknow, 226007, India

(Received on June 08, 2016, revised on June 17, 2016)

Abstract: In a fast and dynamic competitive environment it is not easy to survive and maintain a credit in market. It can be possible when user/customer trust on product quality and its performance. This can be possible when process is well defined because quality of product directly related to the quality of process. To develop a mature and effective process lot of resources is required, every organization want to achieve that at a minimum cost and time. That can be achieved when they effectively utilize the knowledge and experience. At present knowledge management is an area where quality is derived. In this paper we show how knowledge management is used to achieve quality in finished product and proposed a hybrid spiral model which is integration of spiral model and knowledge management. We proposed a hybrid spiral model to improve software quality using knowledge management which is based on knowledge flow during process. Proposed hybrid spiral model has been illustrated with example

Keywords: Software Process, Software Quality, Knowledge Management, Hybrid Spiral Model, Knowledge Flow.

1. Introduction

Software engineering is knowledge management based activity which requires creativity and experience. When experience and knowledge is created, evaluated, engineered, maintained, and disseminated systematically to solve a problem, then this is called knowledge management [1]. Using software development process software is created. To achieve software quality effective process is required. Knowledge helps organization to perform better, faster and develop quality product within time and budget. Software quality totally based upon the quality of process, which is based on the knowledge and experience of whole team of software development. Well define process is required to achieve quality in finished product.

The knowledge related to a company is known as corporate knowledge. Corporate knowledge is the explicit and tacit knowledge; these are two types of knowledge. Explicit knowledge is tangible knowledge which expressed formally, can be seen, stored and updated. Explicit knowledge found in files, databases, documents, research papers, books, source code, e-mail, messages, any written processes and procedures. Tacit knowledge is richer and more valuable than explicit knowledge, which cannot be easily expressed and communicated. It is about human experience, creativity, feelings, and interpretation. Tacit knowledge is based on mental models and perspectives so that people take them for granted, so making it difficult to articulate. Tacit knowledge is found in the heads of individual person (employees and experts), in the collective sharing of employees,
individual person (employees and experts), in the collective sharing of employees, conversations, unwritten processes and procedures and organizational culture [2-5].

There is various quality models [6] are proposed, to improve and evaluate the quality of software products such as: McCall’s Model [7], Boehm Model [8], FURPS Model [9], Dromey Model [10] and ISO 9126 model [11]. McCall’s Model and Boehm Model are same and use hierarchical decomposition of quality factors such as maintainability or reliability. After that FURPS model decomposes quality into functionality, usability, reliability, performance and supportability. These quality models become the basis for ISO/IEC 9126 [12].

For survive in competitive environment and perform effectively knowledge management is required because knowledge management (KM) is a quality management (QM). These quality models useful to derive quality but there is a need to merge these quality models to knowledge management for better performance and output. There are few quality models which uses knowledge management [13] such as: Nonaka Model [14], Capability Maturity Model (CMM) [15] and Knowledge Process Quality Model (KPQM) [16].

Organization uses this combination for improvement, enhancement and customization of software development process. Using that transformation we can improve the entire software development process [17]. This trend is continued and lots of new KM model was developed and proposed according the requirement and need. A General Knowledge Management Maturit Models (G-KMMM), objective to assess the maturity of people, process and technology aspects of KM in organizations [18].

In this paper we discuss the knowledge management and how it is related to software quality. We propose a hybrid spiral model which is integration of spiral model and knowledge management. We propose that integrating software development process, quality management with knowledge management concepts and show how knowledge management leads to improve performance. Section II presents the related work, Section III describes the proposed model concept, algorithm and illustration, and Section IV provides the justification of proposed model and Section V conclude the paper.

2. Related works

For improvement in organization performance effective knowledge management is required because knowledge management is a quality management. Knowledge management is an emerging area which is a dynamic process using that we can create, change and reuse the knowledge to develop high quality product at low cost. Recently academics start to relate knowledge management to quality management [19]. The first study that relate quality and learning, when Fine develop an analytical model and studied the relationship between failure and conformance cost, result show that quality level enhance over time due to learning [20].

Nonaka theory uses knowledge creation as central theme for linking quality to knowledge; it considers both tacit and explicit knowledge. Also suggest knowledge creation in spiral form using four modes of knowledge conversion. Knowledge is created through conversion between tacit and explicit knowledge: 1) from tacit knowledge to tacit knowledge (socialization), 2) from tacit knowledge to explicit knowledge (externalization), 3) from explicit knowledge to explicit knowledge (combination), and 4)
from explicit knowledge to tacit knowledge (internalization) which is called SECI model [21].

The research is continued and lot of influences identified on knowledge management. This included culture, technology, leadership, education, measurement, organizational adjustment, administrative knowledge manipulation activities, values and norms, knowledge resources, evaluation of knowledge management activities, employee motivation, knowledge resources, and external factors [22]. KM implementation and its use have rapidly increased to help organization for better productivity. Because software development is changes quickly and many people is involve for development so for improve productivity and quality software organization use KM in software engineering. Using this combination cost and time reduce and quality increase, share and capture knowledge, provides better decision making, know new technology, and accessing domain knowledge [23].

For better management of knowledge experience management also required because knowledge and experience is interrelated. While knowledge management (KM) has received much attention then experience management (EM) to fill this gap a waterfall model for knowledge management and experience management was proposed which integrate experience or knowledge processing and its management [24]. After that a hierarchical spiral model (HSM) for knowledge management was proposed for development of knowledge management (KM) and information systems (IS). This proposed model provides the guidance between the different phases of knowledge management activities [25].

After several researches new direction towards knowledge management (KM) in software requirement engineering (SRE) to join that combination a knowledge management framework in SRE based on the SECI model was proposed. The aim of this research exploit tacit and explicit knowledge related to software requirements in software project. The main part of the proposed framework is a set of four sub systems such as: socialization, externalization, combination, and internalization connected to a couple of domain ontology and a set of knowledge assets [26].

Presently knowledge management becomes more complex and now users demanding more accurate and reliable knowledge management systems (KMS) to enhance quality. There are software quality dimensions, data quality dimensions, information system quality dimensions, and knowledge management system quality dimensions. Using mapping the quality dimensions to the knowledge management processes, the quality of these processes will be improved as a result and assume that the quality of the KMS will be enhanced too [27]. So there is a need for effective knowledge management to develop a quality product.

There are various software development processes such as waterfall model, iterative and incremental model, V model, spiral model and agile model which have its own advantages and disadvantages. These models are selected according the software requirement and projects [28]. As various models have been proposed by various researchers, all of them are based on explicit knowledge. Existing models have used development process, which has been implemented by person who is having explicit knowledge. Proposed hybrid spiral model uses tacit and explicit knowledge in various
steps. In this paper we select spiral model to propose a hybrid spiral model because spiral model is a meta-model and accommodate any development process.

3. Proposed Hybrid Spiral Model

3.1 Concept

At present software development process continue change according the need and time. There are many people involve in development process. Software organizations use a close relationship of people, process and technology for software development and quality based product at minimum resources (time, cost and people). And also want to improve productivity, for that purpose knowledge is required. Using reuse knowledge and experience we can achieve that at a minimum cost [29].

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Main Findings</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Process Improvement</td>
<td>What knowledge management approach you select in Software Process Improvement, you need to create both tacit and explicit knowledge. Tacit is necessary to change practice and method, explicit is necessary to create an organizational memory or document.</td>
<td>[32]</td>
</tr>
<tr>
<td>SPI</td>
<td>A techno-centric approach to Software Process Improvement may impose unnatural work on an organization and fails to take account of how improvements might occur in practice.</td>
<td>[33]</td>
</tr>
<tr>
<td>CMM</td>
<td>Knowledge management is used as development theory in which a set of key process areas supplement to the CMM in small or medium sized enterprises (SME) that develop software.</td>
<td>[34]</td>
</tr>
<tr>
<td>Unified Process</td>
<td>The iterative approach of Unified Process effects learning and improves communication and work distribution in the company.</td>
<td>[35]</td>
</tr>
<tr>
<td>Software Process</td>
<td>It is possible to define and implement software process in a cost-effective way in small organizations. Special considerations must be given to their specific business goals, characteristics, models, and resource limitations.</td>
<td>[36]</td>
</tr>
</tbody>
</table>

Knowledge Management is being used for efficiency, improvement, performance, maturity, and maintenance of software products. Using integration of technical knowledge with business application domain knowledge software development process increases software development effectiveness and reduces defect density. And also increases software development efficiency and performance [30].

Knowledge Management also used for process improvement. Knowledge management approach adopted in a CMM level 3 to support organizational process tailoring to projects and improvement based on metric data collected from past projects and maintains the project according changes. Using experience and knowledge software process performance improve [31].

We have drawn a Table 1 based on various research which impact on software development process using knowledge management in various conditions and environment.

Software development process which are divided into plan-based or traditional development process and agile model. In plan-based or traditional development process there is lack of learning and past experience. In agile model effective use of learning and
past experience are involve for high customer satisfaction, better decision making in changing environment and quick delivery of quality product. Table 2 show the importance of knowledge and changes in software development process according time and need.

<table>
<thead>
<tr>
<th>Software Development Process</th>
<th>Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan-Based or Traditional Development Process</td>
<td>Explicit Knowledge</td>
</tr>
<tr>
<td>Agile Methods</td>
<td>Tacit Knowledge</td>
</tr>
</tbody>
</table>

Table 2 shows that in traditional development process such as waterfall model explicit knowledge is used and at present agile methods uses tacit knowledge for development [37].

Spiral model is a risk driven process model uses two main features. 1) Cyclic approach for incrementally growing a system's degree and implementation while decreasing its risk, and 2) Anchor Point Milestones for ensuring stakeholder commitment to feasible and mutual satisfactory system solutions [38 -39]. So we select spiral model to propose a hybrid spiral model. The main objective of this paper is to propose a hybrid spiral model to improve software quality.

3.2 Hybrid Spiral Model

Proposed hybrid spiral model uses knowledge management which is based on knowledge flow during process. Knowledge flow of proposed model and how the knowledge flows from one phase to another phase as shown in Figure 1. Hybrid spiral model which is integration of spiral model and knowledge management as shown in Figure 2.

![Figure 1: Knowledge Flow in Proposed Model](image)

Both diagrams are used to describe the how knowledge is used to develop a quality product. Every software development process uses four phase for development such as Requirement, Design, Coding and Testing. Figure 1 shows how these four phases use a knowledge management and flow from one phase to another phase to develop a software product.
Figure 2: Hybrid Spiral Model to Improve Software Quality

PHASE 1: In figure 1 tacit to tacit knowledge conversion which is called socialization is used to show the phase 1 (Requirement). Phase 1 uses tacit knowledge as input and output both. User guideline and system analyst's knowledge used as an input and after requirement gathering draft requirement is created which is an output.

PHASE 2: That draft requirement is used as input to phase 2 (Design) in that case after tacit to tacit knowledge conversion draft architecture is build up. That draft architecture is used as input and design document which is an output. Phase 2 uses tacit to explicit knowledge conversion which is called externalization.

PHASE 3: This design document is used an input for phase 3 (Coding) and using explicit to explicit knowledge conversion a code is develop for software. That code is used as input and using knowledge conversion code is run and check by code developer and that module is an output. In phase 3 explicit to explicit knowledge conversion is used which is called combination.

PHASE 4: That module (code) used as input for phase 4 (Testing) and using knowledge conversion module is tested by software testing team and find out errors or bugs and fix it. After correction draft product is created which is used as input and send to user for any suggestions and feedback if any. The user and analyst meeting or gathering used as output. In phase 4 explicit to tacit knowledge conversion is used which is called internalization.

We use a cyclic and iterative flow of knowledge for software development. In phase 4 if user is satisfied that draft product and its quality then we stops further iteration. Otherwise we go to phase 1 and start collecting user guideline and follow the same processing.

The main characteristic of spiral model is its cyclic nature. Each cycle of spiral model uses four stages. Stage 1 is used to determine the objective and alternatives, stage 2 used to evaluate the alternative and find out risks, stage 3 used to develop and identify
next level of product and stage 4 are used to review the results and plan for next iteration [40]. So based upon these concepts we select spiral model to propose a hybrid spiral model to improve software quality.

Figure 2 shows the hybrid spiral model for software development which is combination of spiral model and knowledge management. For requirement tacit to tacit knowledge required, for design tacit to explicit knowledge required, for coding explicit to explicit knowledge required, and for testing explicit to tacit knowledge required.

All four phase have an individual role and importance but requirement and design play a major and important role to derive quality in finished product. So to capture all requirement and quality parameter effective process is required which can be possible to use a balance relationship between development process and knowledge management to improve software quality.

3.3 Algorithm for Hybrid Spiral Model

This hybrid spiral model uses simple algorithm which is based upon very simple and few steps:

Step 1: Collect software requirement using tacit knowledge means experienced system analysts are assigning to do that job.

Step 2: When SRS (Software Requirement Specification) documented using design process software architecture is derived with the help of tacit knowledge means experienced system designer or architect are involve to do that phase.

Step 3: In next step with the help of design document software design is translated into source code using explicit knowledge means fresher code developer perform that task.

Step 4: After coding each module is tested using explicit knowledge means fresher software testing team are used to do that work.

Step 5: In a last step when final product is develop then this product goes through the user or customer for using and checking the software. If user satisfied with the product performance and quality then no further processing is required otherwise go to step 1 and start collecting requirement.

This algorithm follows these simple steps and it is based upon iteration and cycle form same as spiral model except step 1 and 2 we use tacit knowledge in place of explicit knowledge. Using hybrid spiral model (Spiral Model and KM) we save time, cost and resource because using that model we reuse knowledge and experience which can be helpful to develop software in few iteration and quality can be improve with the help of hybrid spiral model.

3.4 Illustration of Algorithm

As per algorithm proposed in section 3.3 we develop a library management system using Java language and NetBeans IDE 8.1 environment to check to how this hybrid spiral model differs from spiral model and how knowledge management is used to develop a quality product. We select two teams which are divided into tacit and explicit knowledge.
In team 1 fresher person are involve which is based upon explicit knowledge and team 2 experience person are involve which is based upon tacit knowledge. We select four persons for each team. For team 1 explicit knowledge person (approximate 1-2 years experience), we denote that team persons as E1, E2, E3, and E4. For team 2 tacit knowledge persons (approximate 12 years experience), we denote that team persons as T1 and T2.

First we develop library management system using spiral model then we develop this software using proposed hybrid spiral model. To develop library management system using spiral model team 1 is selected and team 1 and 2 is selected for proposed hybrid spiral model. Spiral model uses explicit knowledge for development and there is more iteration to develop a quality product. Proposed hybrid spiral model uses tacit and explicit knowledge for development and there is less iteration to develop a quality product. Illustration of spiral model and hybrid spiral model are shown in Table 3. Table 3 shows the working of both model and comparison also.

<table>
<thead>
<tr>
<th>Spiral Model</th>
<th>Hybrid Spiral Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assigned Person</td>
<td>Assigned Job</td>
</tr>
<tr>
<td>Team 1 E1</td>
<td>For software requirement (Step 1) explicit knowledge is used and there are less correctness and consistency</td>
</tr>
<tr>
<td>Team 1 E2</td>
<td>For software architecture (Step 2) explicit knowledge is used and there are less usability and reliability</td>
</tr>
<tr>
<td>Team 1 E3</td>
<td>For software coding (Step 3) explicit knowledge is used</td>
</tr>
<tr>
<td>Team 1 E4</td>
<td>For software testing (Step 4) explicit knowledge is used</td>
</tr>
</tbody>
</table>

When we use hybrid spiral model to develop this software same language, environment and processing is used. In step 1 and step 2 tacit knowledge involve means an experienced person are selected to develop the software so we select team 1 and 2 for development. Using that algorithm we develop quality product at minimum cost and time.

For software requirement and design team 2 is assigned to do that job. Team 1 is selected to develop code and perform software testing. After testing final product is used by user and user feedback suggest that all good but more performance is required. We plan to next iteration and same processing is used to develop software after development of final revise product that software used by user and user satisfied with the performance and quality. This illustration shows that hybrid spiral model is effective and efficient than existing spiral model.

4. **Justification**

Social factors, environment and experience are significant impact upon process of software development [41]. There are environmental standards and management systems for illusion of progress [42]. Knowledge has its own characteristics and importance. Both
types of knowledge has important role for development but tacit knowledge more important than explicit knowledge because it is based upon learning, experience and creativity. Using tacit knowledge in requirement and design we achieve quality in finished product because an experience people are involve for development and few chances in failure and overrun of time and cost. Table 4 show the comparison between existing spiral model and that proposed hybrid spiral model.

<table>
<thead>
<tr>
<th>Spiral Model</th>
<th>Hybrid Spiral Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit Knowledge</td>
<td>Tacit and Explicit Knowledge</td>
</tr>
<tr>
<td>Costly model to use</td>
<td>Easy to use and maintain</td>
</tr>
<tr>
<td>Risk analysis requires highly specific expertise</td>
<td>No risk problem</td>
</tr>
<tr>
<td>Doesn’t work well for smaller projects</td>
<td>Used for small projects</td>
</tr>
<tr>
<td>Spiral may go infinitely</td>
<td>Few iteration are used</td>
</tr>
<tr>
<td>More documentation</td>
<td>Less documentation</td>
</tr>
<tr>
<td>More time and cost used for development</td>
<td>Less time and cost used for development</td>
</tr>
<tr>
<td>Improve robustness and correctness</td>
<td>Improve consistency, correctness, usability, reliability and performance</td>
</tr>
</tbody>
</table>

Hybrid spiral model is more efficient based upon various parameters given Table 4. It proves that using knowledge management we improve the performance of proposed hybrid spiral model. Knowledge management is required for successful implementation of product and achieves quality in finished product.

5. Conclusions

In this paper we focus on knowledge management and proposed a hybrid spiral model. The main aim of this paper is to design an algorithm for hybrid spiral model which is based upon simple iteration. Proposed hybrid spiral model is a modified version of spiral model which uses knowledge management. Step 1 and 2 of proposed algorithm uses tacit knowledge which improves the software quality of finished product. It has been tested on academic environment to develop a library management system that show that using hybrid spiral model we save time and cost due to reduction of iteration or cycle in comparison of spiral model.

We conclude that effective knowledge management is required to drive quality in finished product. Knowledge management is useful to every development phase of software if we use tacit knowledge in requirement and design phase than the output is quality based product. In future this research work may be extended using knowledge flow on various software development processes.

Acknowledgment

Ms. Shikha Gautam is thankful to UGC for providing UGC-RGNF-JRF fellowship to sustain her research.
References


[22] Holsapple C. W. and K. D. Joshi. Description and Analysis of Existing Knowledge Management Frameworks. 32nd Hawaii International Conference on System Sciences,


[41]. Adolpha S., P. Kruchtena, and W. Hall. Reconciling Perspectives: A Grounded Theory


Brijendra Singh is Professor in Department of Computer Science, University of Lucknow, Lucknow and also served as Head in the same department from 2007-2009 and 2013-2015 respectively. He also worked as a Professor and Head, Department of Information Technology, Vellore Institute of Technology, Vellore. Dr. Singh received his PhD degree in the field of Computer Science from J.K. Institute of Applied Physics and Technology, University of Allahabad. He has taught and researched at various reputed institutions including IIT Kharagpur, Vikram Sarabhai Space Centre (VSSC), Trivandrum, REC Kurukshetra and Korea Atomic Energy Research Institute, Taejon, Korea. Besides, he is the author of four books and fifty five technical papers related to Computer Science and Reliability Engineering. He is recipient of Gowari Memorial Award in 1995 from the Institution of Electronics and Telecommunication Engineers, Delhi. He has also received the honour of International Man of the Year (1999–2000) from International Biographical Centre, Cambridge, CB23QP, England.

Email: drbri_singh@hotmail.com

Shikha Gautam received her B.Tech. degree from IET LKO (UPTU, Lucknow) and M.Tech. degree from SITM LKO (UPTU, Lucknow) India in 2011 and 2014, respectively. At present she is enrolled for PhD degree in department of Computer Science at University of Lucknow, Lucknow, India. Her areas of research interests include software engineering and software quality.

Email: shikhagautam90@gmail.com