Chapter 3

Software Process Model and Knowledge Management

3.1 Introduction

When experience and knowledge is created, evaluated, engineered, maintained, and disseminated systematically to solve a problem, then this is called knowledge management. Software engineering is knowledge management based activity which requires creativity and experience [3.1]. Using software development process software is created. To achieve software quality effective process is required. Knowledge helps software organization to perform better, faster, and develop quality product within time and budget. Software quality totally based upon the quality of software process which is based on knowledge and experience of whole team of software development. Well define process is required to achieve quality in finished product.

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, conversations, unwritten processes and procedures, and organizational culture [3.2-3.5].

There is various quality models [3.6] are proposed, to improve and evaluate the quality of software products such as: McCall’s model [3.7], Boehm model [3.8], FURPS model [3.9], Dromey model [3.10] and ISO 9126 model [3.11]. McCall’s model and Boehm model are same and uses 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 basis for ISO/IEC 9126 [3.12].

To 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 quality model to knowledge management for better performance and output. There are few quality models which uses knowledge management [3.13] such as: Nonaka model [3.14], Capability maturity model (CMM) [3.15] and Knowledge process quality model (KPQM) [3.16].

Software organization uses this combination for improvement, enhancement, and customization of software development process. Using that transformation we can improve the entire software development process [3.17]. This trend is continued and lots of new KM model was developed and proposed according the requirement and need. General knowledge management maturity model (G-KMMM) is proposed to assess the maturity of people, process, and technology aspects of KM in organizations [3.18].

In this chapter proposed hybrid spiral model has been discussed which is integration of spiral model and knowledge management, to show how knowledge management leads to software quality. Section 3.2 explain related work, Section 3.3 describe proposed model concept using algorithm, illustration, and comparison between spiral model and proposed hybrid spiral model.
3.2 Related Work

For improvement in software 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 [3.19]. First study that relate quality and learning is, 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 [3.20].

Nonaka theory uses knowledge creation as central theme for linking quality to knowledge which 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 [3.21].

Research is continued and lot of influence identified on knowledge management. This include 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 [3.22].

KM implementation and its use have rapidly increased to help software organization for better productivity because software development change quickly and many people are involved for software development. So to improve productivity and quality product, software organization uses KM in software engineering. Using this combination cost and time reduce and quality increase due to share and capture knowledge which provide better decision making, know new technology, and accessing domain knowledge [3.23].

For better management of knowledge experience management also required because knowledge and experience is interrelated. While KM has received much attention then experience management (EM), to fill this gap waterfall model for
knowledge management and experience management is proposed by Z. Sun, which integrate experience or knowledge processing and its management [3.24]. After that hierarchical spiral model (HSM) for knowledge management is proposed by Z. Sun, for development of KM and information systems (IS). This model provides the guidance between the different phases of knowledge management activities [3.25].

After several researches new direction towards KM in software requirement engineering (SRE) is join to combine KM framework in SRE based SECI model. Aim of this research is to exploit tacit and explicit knowledge related to software requirements in software project. The main part of this 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 [3.26].

Presently knowledge management becomes more complex and now users demanding more accurate and reliable knowledge management systems (KMS) to enhance software quality. There are software quality dimensions, data quality dimensions, information system quality dimensions, and knowledge management system quality dimensions. Using mapping quality dimensions to knowledge management process, quality of these process will be improved and assume that quality of KMS will be enhanced too [3.27]. So there is a need for effective knowledge management to develop a quality product.

There are various software development process model such as waterfall model, iterative and incremental model, V model, spiral model, and agile model which have its own advantages and disadvantages to develop software products. These models are selected according software requirement and projects [3.28].

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 chapter spiral model is selected to propose hybrid spiral model because spiral model is a meta-model that accommodate any development process.
3.3 Proposed Hybrid Spiral Model

At present software development process continue change according need and time. There are many people involve in development process. Software organizations uses a close relationship of people, process, and technology for software development and quality product at minimum resources (time, cost, resource, and people) and want to improve productivity, for that purpose effective knowledge management is required. Using reuse knowledge and experience we can achieve that at a minimum cost [3.29].

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; quality of software development process increases and defect density reduce. It also increases software development efficiency and performance [3.30].

Knowledge Management also used for process improvement. Knowledge management approach adopted in CMM level 3 to support organizational process tailoring to projects and its improvement based on metric data collected from past projects and maintain project according changes. Using experience and knowledge, software process performance improves [3.31].

Table 3.1 has drawn based on various researches which impact on software development process using knowledge management in various conditions and environment.

<table>
<thead>
<tr>
<th>Concepts</th>
<th>Main Findings</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Process Improvement (SPI)</td>
<td>What knowledge management approach you select in SPI, you need to create both tacit and explicit knowledge. Tacit is necessary to change practice and method while explicit is necessary to create an organizational memory or document.</td>
<td>[3.32]</td>
</tr>
<tr>
<td></td>
<td>A technocentric approach to SPI may impose unnatural work on an organization and fails to take account of how improvements might occur in practice.</td>
<td>[3.33]</td>
</tr>
<tr>
<td>CMM</td>
<td>Knowledge management is used as development theory in which a set of key process areas supplement to CMM in small or medium sized enterprises (SME) that develop software.</td>
<td>[3.34]</td>
</tr>
<tr>
<td>Unified Process</td>
<td>Iterative approach of unified process effects learning and improves communication and work distribution in the company.</td>
<td>[3.35]</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>[3.36]</td>
</tr>
</tbody>
</table>
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 user satisfaction, better decision making in changing environment, and quick delivery of quality product. Table 3.2 show the importance of knowledge and changes in software development process according time and need.

Table 3.2: Software Development Process and Knowledge

<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 3.2 shows that in traditional development process such as waterfall model explicit knowledge is used and at present agile methods uses tacit knowledge for development [3.37].

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 [3.38-3.39]. So spiral model is selected to propose hybrid spiral model. The main objective of this chapter is to propose hybrid spiral model to improve software quality.

3.3.1 Algorithm for 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 knowledge flows from one phase to another phase is shown in Figure 3.1. Proposed hybrid spiral model which is integration of spiral model and knowledge management is shown in Figure 3.2.

Both figure are used to describe how knowledge is used to develop quality product. Every software development process uses four phase for software development such as: requirement, design, coding, and testing. Figure 3.1 shows how
these four phase uses knowledge management and flow from one phase to another phase to develop software product.

![Knowledge Flow in Proposed Hybrid Spiral Model](image1)

**Figure 3.1: Knowledge Flow in Proposed Hybrid Spiral Model**

![Hybrid Spiral Model to Improve Software Quality](image2)

**Figure 3.2: Hybrid Spiral Model to Improve Software Quality**

**Phase 1**: In Figure 3.1 tacit to tacit knowledge conversion which is called socialization is used to show 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**: This draft requirement is used as input to phase 2 (design) in that case after tacit to tacit knowledge conversion draft architecture is created. This draft architecture is used as input and design document developed 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 code develop for software. This code is used as input and using knowledge conversion code is run and check by code
developer and that module used as output. In phase 3 explicit to explicit knowledge conversion is used which is called combination.

Phase 4: This 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 develop which is used as input and send to user for any suggestions and feedback if any. If any correction required then user and analyst meeting or gathering are used for further processing otherwise this draft product is final product. In phase 4 explicit to tacit knowledge conversion is used which is called internalization.

A cyclic and iterative flow of knowledge is used for software development. In phase 4 if user is satisfied with draft product and its quality then stop further iteration. Otherwise 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 if require [3.40].

Figure 3.2 shows 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 important 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.

### 3.3.2 Illustration of Algorithm

Proposed 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 analyst is assign to do that job.

Step 2: When Software Requirement Specification (SRS) 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 work.

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 software coding.

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

Step 5: When final product is develop then this product goe through user for using and checking the software. If user satisfied with the final product 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 same as spiral model except step 1 and 2 where tacit knowledge is used in place of explicit knowledge. Using hybrid spiral model (spiral model and KM) time, cost, and resource are save because using that model we reuse knowledge and experience which can be helpful to develop software in few iteration and software quality can be improve with the help of hybrid spiral model.

As per proposed algorithm, library management system (LMS) is developed using Java language and NetBeans IDE 8.1 environment (see Appendix A for Java coding) to check how hybrid spiral model differ from spiral model and how knowledge management is used to develop quality product.

Two teams are selected which belong to tacit and explicit knowledge. In team 1 fresher person are involve which is based on explicit knowledge and team 2 experience person are involve which is based on tacit knowledge. Four persons for each team are selected. For team 1 explicit knowledge persons (approximate 1-2 years experience) denoted as E1, E2, E3, and E4. For team 2 tacit knowledge persons (approximate 12 years experience) denoted as T1 and T2.
Firstly LMS is developed using spiral model then this software is developed using proposed hybrid spiral model. To develop LMS 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 quality product.

Proposed hybrid spiral model uses tacit and explicit knowledge for development and there is less iteration to develop quality product means time and cost save. Illustration of spiral model and proposed hybrid spiral model are shown in Table 3.3. Table 3.3 shows working of both model and comparison too.

<table>
<thead>
<tr>
<th>Spiral Model</th>
<th>Assigned Person</th>
<th>Assigned Job</th>
<th>Proposed Hybrid Spiral Model</th>
<th>Assigned Person</th>
<th>Assigned Job</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Team 2</td>
<td>T1</td>
<td>For software requirement (Step 1) tacit knowledge is used and there are more correctness and consistency</td>
</tr>
<tr>
<td>Team 1 E1</td>
<td></td>
<td>For software requirement (Step 1) explicit knowledge is used and there are less correctness and consistency</td>
<td>Team 2 T1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Team 1 E2</td>
<td></td>
<td>For software architecture (Step 2) explicit knowledge is used and there are less usability and reliability</td>
<td>Team 2 T2</td>
<td>For software architecture (Step 2) tacit knowledge is used and there are more usability and reliability</td>
<td></td>
</tr>
<tr>
<td>Team 1 E3</td>
<td></td>
<td>For software coding (Step 3) explicit knowledge is used</td>
<td>Team 1 E3</td>
<td>For software coding (Step 3) explicit knowledge is used</td>
<td></td>
</tr>
<tr>
<td>Team 1 E4</td>
<td></td>
<td>For software testing (Step 4) explicit knowledge is used</td>
<td>Team 1 E4</td>
<td>For software testing (Step 4) explicit knowledge is used</td>
<td></td>
</tr>
</tbody>
</table>

Proposed hybrid spiral model is used to develop LMS using same language, environment, and processing. In step 1 and step 2 tacit knowledge used means experienced person are selected to develop the software so team 1 and 2 are selected for software development.

For software requirement and design team 2 is assigned to do that job. Team 1 is selected to develop software code and perform software testing. After testing final product is used by user and user opinion or feedback suggests further iteration are required or not. If iteration is require than same processing is used to develop software. After software development revise product is used by user and user satisfied with the product performance and quality. Illustration shows that hybrid spiral model is more effective and efficient than spiral model.
3.3.3 Comparison between Spiral Model and Hybrid Spiral Model

Social factors, environment, and experience are significant impact upon process of software development [3.41]. There are environmental standards and management systems for illusion of progress [3.42]. Knowledge has its own characteristics and importance. Both types of knowledge has important role for software development but tacit knowledge is more important than explicit knowledge because it is based upon learning, experience, and creativity. Using tacit knowledge in requirement and design; quality achieved in finished product because experience person are involve for software development and few chance to failure and overrun of time and cost. Table 3.4 show the comparison between existing spiral model and proposed hybrid spiral model.

<table>
<thead>
<tr>
<th>Spiral Model</th>
<th>Proposed 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,</td>
</tr>
<tr>
<td></td>
<td>reliability, and performance</td>
</tr>
</tbody>
</table>

Proposed hybrid spiral model is more efficient based upon various parameters as given in Table 3.4 which proves that using knowledge management software quality is improved. Knowledge management is required for successful implementation of software product and achieves quality in finished product.

3.4 Conclusion

Objective of this chapter is to propose 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 (as section 3.3.2) of proposed algorithm uses tacit knowledge which develop quality product. If we use tacit knowledge in requirement and design phase then output is quality product.
3.5 References


