A Goal-Driven Framework in Support of Knowledge Management

Jinru Hua, Guoping Rong, Dong Shao
Software Institute, Nanjing University
22 Hankou Road, Nanjing, P.R.China
(86)25 83621362
{hjr08, ronggp, dongshao}@software.nju.edu.cn

He Zhang
National ICT Australia
University of New South Wales
he.zhang@nicta.com.au

ABSTRACT
Knowledge Management (KM) nowadays receives more and more attentions due to its promise that learning from the past will usually help software engineers make sound decisions under uncertainty. However, reality from the industry indicates that application of KM is far from ideal. Several obstacles existed in current KM mechanism, e.g., mass information impedes knowledge discovery; culture of organization may encourage short-term goals and individualism; uncertainty of knowledge for future usage reduces motivation to carry out KM at both individual and team levels; etc. As a result, few organizations established useful knowledge repository. And knowledge and experience cannot be transferred and shared effectively. This paper proposed a pragmatic KM framework (GDD-KM) which featured in its goal-driven nature. With GDD-KM, knowledge and experience are organized by means of goals template and related solutions. Since most goals can be easily shared among different projects within organizations, the corresponding solutions to these goals will also be shared by these teams. GDD-KM has been applied in several projects in both school and industry; results indicated that GDD-KM helped these teams in typically three aspects: to perform better postmortem; to establish the initial version of knowledge and experience repository for future use and to apply some knowledge and experience in subsequent development.

Categories and Subject Descriptors
D.2.9 [Software Engineering]: Software Management – Software development, Software maintenance, Software process and Software selection

General Terms
Management

Keywords
Goal-driven: Knowledge management.

1. INTRODUCTION
Software engineering is knowledge-intensive work. Therefore, development of software requires knowledge and experience in many areas. Software engineering involves several knowledge types—technical, managerial, domain, corporate, product, and project knowledge [1]. These knowledge and experience help people to decrease software projects’ development time and costs, avoid mistakes, reduce rework, and repeat successful processes, as well as increase productivity and the likelihood of further success [1]. They can also help software practitioners make sound decisions under uncertainty and to find better compromises [2]. In one case, effectively exchange of knowledge is the glue that holds a company together [3]. Unfortunately, the reality is that most development teams do not benefit from existing experience as expected and they repeat mistakes even though some individuals in the organization know how to avoid them. Lacking effective KM is one major reason of the reality. Therefore, management of knowledge and experience is a big challenge for most software organizations. As pointed by Ioana Rus et al. the challenges and obstacles of KM came from three major sources, the technological issues, the organizational issues and the individual issues [1, 9].

From the technological perspective, most KM information systems are what so-called closed systems [4]. With closed systems, knowledge and experience are more like documented materials which store answers to questions that might arise during work. Therefore, consumers of the knowledge need to discover the answers when facing specific problems. However, these systems usually contain thousands of pages of unstructured information, as a result, the discovery procedure could be time-consuming and software engineers may lose faith in and patience under tight project schedule. Besides, updating and maintaining knowledge and experience will also be a big issue.

From the organizational perspective, the culture of most organizations usually only encourage finishing current project on time, within budget and with quality. In firms, many resources and much time and effort are required before benefits became visible, as a consequence, project teams tend to spend little effort and resource to help future projects. For example, organizations with hero culture may encourage individualism rather than cooperative work. Therefore, employee might not be willing to share knowledge and experience at the organizational level. What’s worse, few organizations established specific processes for KM and neither maintenance of knowledge and experience nor application of them could be effectively conducted.

From the individual perspective, development team members might lack of motivation to accumulate valuable knowledge and experience in current projects. Project managers would rather focus on completing their current project on time than help the next project manager succeed. Therefore, they often consider accumulating knowledge and experience a burden, not mention that “when and how to use the knowledge and experience in future projects is usually uncertain” [1]. For other engineers, they often “do not have time to input or search for knowledge, do not
want to give away their knowledge, and do not want to reuse someone else’s knowledge” [1]. Besides, technology’s fast pace often discourages software engineers from analyzing the knowledge they gained during the project, believing that sharing the knowledge in the future will not be useful.

In this paper, we proposed a Goal-Driven Development KM (GDD-KM) framework which featured in its inherent Goal-Driven philosophy and cyclic approach. Goal-Driven philosophy helps to organically structure knowledge and experience since only those knowledge and experience which help to satisfy identified goals were captured and organized in the knowledge repository. Meanwhile, the cyclic nature of the GDD-KM helps to enrich, refine and outdate knowledge and experience as necessary. In this sense, knowledge and experience evolve with each cycle. We applied GDD-KM in several projects and helped these teams to accumulate and transfer knowledge and experience effectively and efficiently.

The rest of the paper is structured as follows. Section 2 provides a brief introduction to the background of this paper. i.e. the related work in KM; GDD project management method and the GDD-KM cycle. Section 3 describes the application of GDD-KM in a real project. Some empirical evidences were collected and analyzed to verify the effectiveness of GDD-KM. Section 4 compares GDD-KM with traditional KM methods and further discusses several limitations on GDD-KM at this stage. The paper is concluded in Section 5 with the suggestions on future continuous research.

2. BACKGROUND

2.1 Related Work in Knowledge Management

Research and practice in this area occurred at different layers and in organizations with different size. Schneider et.al [5] compiled five-year-long experience in building, revising, and improving experience repositories in DaimlerChrysler Research Center. They found that for most knowledge repositories, the contents of the database were unordered and the forms they provided were so sophisticated that they could hardly provide any guidance to users. As Schneider et.al pointed, experiences should not be stored on a dusty shelf, but engineered into best practices and processes that guide user work. Another famous KM implementation came from NASA Software Engineering Laboratory as the first implementation of “Experience Factory” [6]. Jay Liebowitz described this KM implementation based on working experience on NASA and concludes that KM should start small and see what works in a specific environment and that knowledge should be collected during projects, not after their completion [7].

Compared with Experience Factory that requires numerous effort and resources, PMA (PostMortem Analyze) offers a quick and simple way to initiate knowledge management in small- or medium-size software projects. Andreas Birk et al. [8] described an effective solution to address knowledge and experience sharing issue by conducting postmortem with an open atmosphere. They received a lot of positive feedback from PMA participants in different companies. By using systematic postmortem analysis for capturing and reusing experience and improvement suggestions, the teams in one organization increased experience understanding and sharing. However, as Birk et al [8] said, PMA has been mainly advocated for situations such as completion of large projects, learning from success, or recovering from failure. With popularity of Agile process and practices [10], continuous and quick knowledge and experience evolution and sharing will be more and more important.

There also exist numerous successful KM practices. Ramesh, B studied best practices in 30 organizations and concluded the importance of link knowledge fragment to guarantee traceability (creating and maintaining relationships between objects and people in software development) [11]. This traceability will facilitate successful knowledge transfer and reuse. Seija Komi-Sirviö et al. emphasized the need for addressing local needs, problems, and specific context for KM initiative implementation [12]. Therefore, to make knowledge and experience useful, project-based KM approach will facilitate knowledge collection and delivery better than large scale KM approach at the organizational level.

Through the above practice and research, several characteristics of effective KM were identified as the following:

- Standard, flexible knowledge structure with a clear specification
- Collect knowledge during projects, not after their completion
- Focus on project-based KM approach and link knowledge fragment to guarantee traceability
- Clear purpose of using the knowledge

2.2 GDD Introduction

Goal-Driven Development (GDD) is a project management methodology which is designed to improve the traditional project management methods by focusing on meeting more project goals. Practice and evaluation of GDD were described in [13]. A typical cycle of GDD is composed of three primary phases: the launch phase, the development phase and the postmortem phase. Figure 1 shows the sequence of GDD method.

![Figure 1. Process framework of GDD method](image)

2.2.1 Practices supporting KM

Knowledge and experience management is a build-in mechanism of GDD method. Several practices in GDD, especially the practices in launch phase and postmortem phase support KM well.
2.2.1.1 Launch phase
A launch phase consists of five meetings as depicted in figure 1, among which the goals elicitation meeting and the goals implementation solution meeting help to establish the initial version of knowledge and experience repository for project teams.

- Goals elicitation

In this meeting, the project team members work together to identify two types of goals: internal and external goals. To elicit goals comprehensively and systematically, a tree-styled diagram (mind-mapping) can be used to facilitate goal identification. The high level goals need to be broken down until the related implementation solutions are applicable and operational to the development team. Typical goals can be grouped into four categories at the top level: product, project, process and team. Figure 2 depicts an example of goal break-down structure. In practice, many projects can group their goals into these four categories.

- Goals implementation solution

To promote full participation, the leaf nodes of goal template should be allocated to all team members according to preference and balanced workload. Therefore, each goal should be assigned to a fixed owner, who is responsible to develop the implementation solution and track the status during development. After that, the whole team should reach consensus on implementation solutions of all the goals on the fine-grained level. Description of the solutions to implement projects goals are the most valuable knowledge and experience for next projects. When all these information has been organized according to the structure of project goals, retrieval of information will be easy. Besides, this goal-solutions structure will also provide software engineers with certain context of application when they create knowledge and experience. Therefore, they may have the motivation to accumulate and share individual knowledge and experience.

2.2.1.2 Postmortem phase
In postmortem phase of GDD method, the development team only discusses two questions. 1) How did all the goals identified in the development process support the expectations of all the stakeholders? 2) How did all the implementation solutions help to achieve project goals?

The answer for the first question will help the team find improvement opportunities on goal elicitation; the answer for the second issue will help the team identify effective solutions as the best practices to support project goals. In this way, knowledge and experiences created during the project can be well validated for future projects.

2.3 Cycle of GDD-KM
While GDD usually occurred within project, KM happened across multiple projects. Therefore, from the perspective of KM, GDD-KM should comprise of five typical steps. i.e. knowledge creation, knowledge integration and formation, knowledge dissemination and application, knowledge evolution and knowledge outdated. These five steps forms a cycle which facilities knowledge evolution. Figure 3 shows the cycle of GDD-KM.

![Figure 3. The cycle of GDD-KM framework.](image-url)
therefore, the knowledge and experiences gained from one project can apply to other projects. With representational formalization, project teams organize knowledge and experience in a way which facilitates access and apply. In GDD-KM, the basic structure of knowledge and experience repository is three layered. Figure 5 shows the three layers. The top layer is the goal identified by project teams in the organization. In fact, the content of the top level also belongs to the concept of knowledge and experience. In GDD-KM, several goal templates will be defined to record the goals identified in former projects hence to help new projects of project goals. The intermediate layer is a brief introduction to solutions to address goal on top level. To be pragmatic, the description must be concise and perspicuous. Details of the solutions are described on the bottom layer. On this layer, detailed and concrete guidance are provided to carry out the solutions and to satisfy goals as well.

![Figure 4. Basic structure of knowledge and experience repository.](image)

2.3.3 Knowledge dissemination and application
GDD-KM is relatively distinct from the traditional KM approaches on its integration of knowledge creation and application. Traditional KM approaches assume that workers perform repetitive and predictable tasks, so they broadcast information (for example, email), provide searchable databases, disseminate knowledge through classroom training or printed reference documents. These approaches separate learning and working [7]. While in GDD-KM, full participation is highly encouraged in both creation and application of knowledge and experience. Based on similar goal template and corresponding solutions to goals across multiple projects, dissemination of knowledge and experience at organizational level could be feasible. In fact, since goals in GDD were interpreted as internal and external expectations from relevant stakeholders, usually these expectations were similar except for some technological details. Typical application scenario is in the launch phase of GDD method when project team elicits goals and develops implementation solutions to goals.

2.3.3.1 Goals elicitation
One challenge a software project team will face at early stage is that how to elicit expectations from relevant stakeholders comprehensively. With GDD-KM, each project team is equipped with a goal template, which provides a basis for further elicitation of project goals. Usually with similar project context, e.g. later development iteration of the same project, nearly no change needs to make to the goal template. Even with different project context, most expectations from stakeholders are addressed by the goal template; hence project team will be at a good position to modify the goal template to meet new expectations.

2.3.3.2 Goals implementation solution
With the basic structure of the knowledge and experience repository as depicted in figure 5, solutions are attached to certain project goals. Therefore, for those same goals, there exist solutions in the knowledge and experience repository already. Since most project goals are similar across multiple projects, this helps a lot for goal owners to develop implementation solution.

2.3.4 Knowledge evolution
Knowledge is refined in evolution phase. In this phase, every layer of knowledge in GDD-KM (goal, solution and detail) is refined, updated, and detailed into full conceptual knowledge structures.

Additionally, the focus of KM moves from a specific knowledge transfer, such as how to execute an experiment and reporting the results of an experiment, to a broader knowledge evolution and sharing, such as experiment improvement and how to combine data from different experiments. It means that, after several refining cycles, the knowledge no longer stays in a specific project level, but becomes resource that can be shared and reused in future projects.

2.3.5 Knowledge outdated
Knowledge is context-specific and thus has to change with changing environment. However, to be useful the knowledge repository should be kept concise. This requires the knowledge stored to be kept up to date. Appropriate changes need to be undertaken, while outdated knowledge also needs to be discarded.

Different from other KM approaches that blend knowledge outdated process into knowledge update (evolution) procedure [8], in GDD-KM cycle, knowledge outdated acts as a specific step at the same level as knowledge update. With caution, outdated knowledge and experience should be removed from the repository. E.g. goal of process discipline could be removed when process discipline was ingraind in the organization culture, solution corresponding to certain goal could be removed while better solutions were identified, and technologies as the solution can be removed when it is substituted or even deprecated, etc.

3. CASE STUDY
We applied GDD-KM in several projects both in industrial environment and academic environment. Results indicated positive effects of GDD-KM in these projects. In one student project, all the team members conducted effective postmortem and accumulated valuable knowledge and experience. The purpose of this project is to develop a general system with the function of generating schematic map for 2014 NYOG (Nanjing Youth Olympic Games). The system should be able to create accurate route, change the relative distance between two adjacent significant features by zooming the image, schematize the route and finally create a schematic map. Using the development method described in [13], the team finished the project successfully. Meanwhile, they also established an initial knowledge and experience repository.

3.1 Implementation of GDD-KM
Typical steps in a GDD-KM cycle were supported by GDD. Inversely, GDD-KM also helped the team practice GDD method.


3.1.1 Creation
During launch phase of GDD, the whole project team established the first version of goals set and grouped goals into four categories, in two kinds: external goals (from expectations from senior management or customers) and internal goals (from team members). In this way, a goal template which is similar to figure 2 was then formed. This goal template was used in the following several iterations of this project.

After each goal was allocated to a team member, solutions to implement the goal were then developed. As shown in figure 5, to meet the quality goal, the team decided to rely on reviews rather than testing to achieve high quality. To direct practice, controlling parameters of the review process were then defined in the details of the solution. The whole team then conducted a meeting to reach consensus on both the goals set and corresponding solutions.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>High quality of the final product</td>
<td>Rely on review rather than testing; Control the review rate (Less than 200 LOC/Hour); Review both detailed design and code by the developer himself/herself; Meanwhile, conduct inspection on core modules;</td>
</tr>
</tbody>
</table>

Figure 5. Solutions to meet quality goal.

3.1.2 Knowledge integration and formation
During the postmortem phase of GDD, the whole project team reviews the data and information recorded during the development to evaluate the effectiveness of goal elicitation process and implementation solution to address the goals. Some valuable experiences to meet similar goals were then integrated. E.g. in this project, continuous integration and testing was also identified as a solution to address quality goal. Therefore, a new solution was added to the goal of “high quality of the final product”.

3.1.3 Knowledge dissemination and application
In this case, since the project team developed the whole system with multiple iterations, each following iteration had similar context as the first iteration. This provides a good foundation for knowledge dissemination and application. In fact, the project used most of the knowledge and experience gained in this project in other projects. Feedback from team members indicated positive results of GDD-KM.

3.1.4 Knowledge evolution
In this project, the development team found several useful approaches to get steady review quality. E.g. making a customized review checklist at individual level, using a total different environment than the lab where the team developed the project, etc. Therefore, details of the review-based solution were supplemented and evolved to include new knowledge and experience.

3.1.5 Knowledge outdated 
At this early stage of application of GDD-KM, knowledge outdated did not happen in this project.

3.2 Project Results
This team finished the project on time and met most of the goals identified during launch phase. We conducted an investigation after the project to interview all the project team members. Results showed several positive aspects of GDD and GDD-KM.

3.2.1 About postmortem
Most team members thought that GDD-KM helped them to do better postmortem due to the structured organization of the postmortem process and knowledge and experience framework. Firstly, the goal template provided them the concrete scope to analyze. Secondly, information summarized during postmortem was provided a certain application context.

3.2.2 About knowledge and experience repository
The basic structure of knowledge and repository provided an easy approach to capture and organize valuable information. With GDD-KM, knowledge and experience can also be accumulated and evolved in the repository. Besides, the goals template provides a fast way to retrieval information from the repository.

3.2.3 About continuous application of knowledge and experience
GDD works well in multiple iterations. Therefore, GDD-KM provided a mechanism for fast and continuous application of knowledge and experiences gained in former iterations. This fact not only helped the team to complete the project successfully, but also increased the motivation of the team members to record knowledge and experience. Besides, it also facilitated the verification and validation of knowledge and experience through repeatedly application.

4. DISCUSSION

4.1 Limitation of GDD-KM
GDD-KM provides a pragmatic framework for knowledge and experience management. However, there still exist several limitations at this stage.

Firstly, GDD-KM is a build-in mechanism of GDD method. Therefore, it highly relies on GDD. Project teams must use GDD to manage their projects so as to provide the data needed for GDD-KM. E.g. the goals template, the solutions attached to certain goals, etc.

Secondly, GDD-KM is more suitable to handle what so-called codification [14] strategy to manage knowledge and experience. Within this approach, the role of KM is to support the storage and retrieval of explicit documented knowledge by people throughout the organization as and when required. However, there exists another strategy called personalization [14]. Within this strategy, KM is used to extend interpersonal networks and the ability to connect and communicate with one another [14]. We believe much can be gained by more researches on how to mix the Codification and Personalization via GDD-KM methods.

Thirdly, for pragmatic purpose, knowledge representation in GDD-KM is based on natural language at this stage. Redundant information and imprecise description cannot be totally avoided. With increased size of the knowledge and experience repository, it will be more and more difficult to conduct knowledge integration and formation.

4.2 Research consideration
The research in this paper indicated several positive results of GDD-KM. However, there still exist several considerations while applying GDD-KM in practice.

Firstly, the project described in this paper is a student’s contest project. The difference on main purpose between industrial projects and this academic project may affect the results. E.g., for industrial projects, to finish project on time, within budget and...
with high quality may constitute the top goals; meanwhile, for contest projects, usually to finish the project on time, with attractive GUI may become the top goals. However, the project in this research was developed in summer school when students worked just as full-time professionals. Research conducted by Runeson et al. [15, 16] indicated that with similar environment, similar improvements trends can be identified both in industrial projects and academic projects.

Secondly, considering the duration and staff size, the projects we applied GDD-KM are not big projects. When the scale of the project increases, the number of goals and related solutions will also increase as well. Therefore, it will be much more difficult manage knowledge and experience.

5. CONCLUSION
Knowledge workers as software engineers now have more information available than they could understand and apply, while on the other hand, finding information relevant to the task at hand is becoming increasingly critical. To address information overload, KM approaches must provide the information workers need, when they need it. The tree-structure goals and three layer mode of representation in GDD-KM provide an organized guidance for users to create, integrate and apply the knowledge to future projects. Moreover, thanks to the shared goal template in different projects within organizations, the corresponding solutions to these goals can also be shared by these teams, which improve the knowledge transfer in different projects.

Another contribution of GDD-KM is to help establish an initial knowledge and experience repository according to a common goal template. Kurt Schneider et.al mentioned that Be Specific and User Guidance should be considered as two significant success factors[17]. It means that, for one thing, Experience Repository should be well-organized; for another, Repository should be organized as simple and flexible as possible to facilitate the usage of knowledge to the full extent.

Our work is a worthy attempt to manage knowledge and experience in software engineering pragmatically. There still exist several interesting issues which need future research, for example:

1) How to combine Codification with Personalization in GDD-KM? As Anthony et al. [18] claimed that the field is moving from first to second generation, KM is characterized by knowing-in-action. This means Personalization became more and more important to make KM useful. Therefore, more research needs to be conducted to combine both strategies in GDD-KM.

2) How to handle the scale issue? With increased project scale, KM will be more and more difficult and ineffective due to mass information. Therefore, to design new approaches to represent knowledge and experience and new methods to store and retrieve useful knowledge in GDD-KM is a promising research field.

6. REFERENCES