Abstract—Technologies and methods for the development of embedded system projects are highly constrained by predefined hardware and software platforms. In this sense, embedded system projects may have more goals (derived from constraints) to achieve than regular software projects. Without pragmatic support, engineers from different disciplines are likely to neglect some project goals in the real-world embedded system projects. As a consequence, the success of embedded system projects may be more difficult to achieve than regular software projects. In this paper we report experiences gained during applying a goal driven project management methodology on several embedded system projects in a software company. We evaluated the effectiveness and efficiency of our Goal-Driven Development (GDD) methodology in practice by both projects results and feedbacks from relevant stakeholders. The results of our study show that GDD enables embedded system project teams to systematically and effectively identify, understand, track, and ultimately realize the project goals to meet relevant stakeholders’ expectations. Being supported by GDD, explicit linkages and assignments are established between goals and solutions with project team’s commitments.

Keywords— Embedded System Engineering; Project Management; Goal-Driven

I. INTRODUCTION

With the proliferation of embedded ubiquitous systems in all aspects of human life, these embedded systems have become a fast growing industry. Embedded software is being used in more and more products, such as automotive systems, telecommunications, and consumer electronics. Both the size (complexity) of embedded software and the percentage of software in a product grown rapidly. This trend raises big challenges for the embedded system practitioners. Embedded systems have to rely on high quality hardware as well as high quality software. But as the complexity and diversity of applications increase, a number of embedded system projects are struggling to deliver a quality product within budget and on time. The recent Embedded Market Survey [1] indicates top four concerns are related to process and project management aspects during embedded systems development: meeting schedules, the debugging process, increased lines of and complexity of code and sticking to cost budget. To optimize the timeliness, productivity, and quality of embedded system development, industry needs pragmatic process techniques and project management methods that are able to effectively tackle the specific situations.

Compared to regular software development, embedded system development appears more complex and difficult due to the inherent constraints within the embedded systems to be developed as well as with other systems outside but working with the embedded systems. For example, most embedded systems are tightly coupled based on particular predefined hardware platforms and software platforms. Besides, many embedded systems work as real-time systems, on which performance metrics (e.g., response time and stability) becomes critical criteria. On the other hand, the processing capability and equipped memory of embedded devices are usually limited by the cost. The issue of putting “ten pounds in a five-pound sack” [2] becomes more and more critical to the success of embedded system projects.

Compatibility usually relates to more limitations or constraints, which may further result in more goals attached to embedded system projects than regular software projects. In real embedded system projects, we found even skilled engineers are likely to neglect several goals (constraints). This ‘occasional’ negligence may cause serious, sometimes fatal, consequences of embedded system projects. For example, in a typical lifecycle of embedded system project, performance goals are usually only identified in the requirements phase and verified in the testing phase but may be ignored during the development phases. If testing results indicate unacceptable performance of the embedded system, project rework or even failure cannot be avoided.

This paper reports experiences and lessons gained from applying a Goal-Driven Development method (GDD) in DONN Tech., one Chinese software company whose main business area is embedded system development. Several projects adopted GDD, and in this report one project was selected as a focused project. This project was to develop indoor locating system (the ilocation system). The project included both hardware development and software development. In the study, we explored an initiative to adopt GDD as its project management method and evaluated the impact of GDD method during the ilocation project.

The rest of the paper is structured as follows. Section 2 provides an introduction of GDD method. Section 3
describes the context of the case study. Section 4 elaborates the implementation details of the case study. Section 5 discusses the reflections, best practices and limitations of the case study. Section 6 concludes this paper and suggests the future research on GDD.

II. GDD METHOD

In GDD, a goal is defined as an explicit or implicit expectation from the relevant stakeholders of a certain project. These relevant stakeholders who are involved in a project may typically include customers, end users, management and developers. In this sense, goals are quite different from traditional concept of requirements [3, 4], since most requirements may be raised by the stakeholders outside a project team. For example, a response time less than 2 seconds can be a performance goal, while a balance between job time and family time can be a team member’s personal satisfaction goal. The former may be the expectation from the customer or the management which in most cases will also be taken as requirements. Meanwhile the latter may be expectation from the project staff which in most cases will not be interpreted as requirements.

A GDD cycle 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)

1) Phases
   a) Launch phase

   A typical launch may last 2 or 3 days, which mainly depends on the scale of the project. A launch phase consists of five meetings respectively, the goals elicitation meeting, the goals analysis and criteria identification meeting, the goals allocation meeting, the goals implementation solution meeting and the implementation solution consensus meeting.

   - Goals elicitation

   This meeting identifies two types of goals: internal and external goals. External goals usually represent expectations from senior management and customers, e.g., functionality, embedded platform, quality, cost limits and process discipline etc. Internal goals usually denote expectations from the team members, e.g., balance between work time and family time, career development, etc. 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 analysis and criteria identification

   This step analyzes all the goals on leaves to identify any potential conflicts and duplications among the goals on particular implementation solutions. Conflicts are very common when goals come from different stakeholders. For example, in embedded system projects, conflicts often exist between the hardware performance and the software algorithm on its top. Besides, although goals are grouped into four different categories, duplicates among the implementation solutions may be very common. For example, the solution to finishing the project on schedule may include similar practices to the solution to collecting process data, because process data is needed to show the status of project progress objectively. Accordingly some goals can be assigned to one team member to follow when their solutions are interrelated. Objective criteria should also be developed to help determine whether or not a goal is satisfied. And if possible, metrics to measure the status of each goal are also needed. A GQM [5] approach may be used to establish the appropriate metrics. In addition, team judgment is also a very useful way to establish metrics. For instance, one tries to measure the status of a specific goal-to balance work time and family time. Using GQM, the metrics can be normal working hours and overtime hours. While using team judgment, the metrics can be percentage of team members who feels peace with the current workload.

   - Goals allocation

   Figure 2. A break down structure of project goals.
This step allocates all the goals on leaves among team members according to preference and balanced workload. Normally each goal should be assigned an owner, who is responsible to develop the implementation solution and to track the status during development. Allocation of goals among all the team members also encourages the full team’s participation in project management which will in return contribute to the establishment of a self-directed team.

- **Goals implementation solution**

The owner of the goal is responsible to lead the development of the solution when it is complicated. For instance, the solution to satisfying schedule goal usually needs support from an accurate project plan. Such a plan is difficult to be developed by any individual team member. In this situation, the owner of the schedule goal may lead the team to develop a project plan. When it’s impossible to satisfy a certain goal, the alternative goals and their associated implementation solutions should be proposed.

With embedded system projects, solutions to implementing certain project goals usually encounter more constrains than regular projects. Although it’s inevitable to neglect some goals or the related constrains during the intricacy of project tasks, failing to recognize, monitor and re-emphasize them may easily lead to failure of embedded projects. Therefore, solutions to goals in embedded system projects need to be verified constantly with the increasing knowledge of the project context.

- **Implementation solution consensus**

This step is used to reach consensus on implementation solutions of all the goals on the fine-grained level. All the stakeholders should be invited to attend the last meeting of launch phase, where the team leader will present all the goals and the corresponding solutions. Then all the relevant stakeholders of the project will discuss the goals and try to reach consensus. In embedded system projects, the situation that the engineers come from different disciplines is common; hence reaching consensus on the solution becomes especially important in order to avoid misunderstanding among those engineers during the project.

b) **Development phase**

Different from the lifecycle of regular project management process; the practices of development phase are relatively simple in GDD. Only one practice is compulsory in this phase, the goals tracking and status reporting. Depending on the tracking results, re-launch is an optional practice in the development phase.

- **Goals tracking and status reporting**

This practice tracks and reports the status of all the goals identified by the development team. Team weekly meeting is the main event to track and report the status of all the goals. Before the meeting, the owner of each goal is responsible to collect evidence and data to characterize the status of the goal. And then the team leader will use a tool called goal profile to summarize all the goals’ status in one single chart. Figure 3 shows a sample goal profile chart. Where the blue bars indicate the deviation of each goal and other colors indicate the health status of the goals.

```
Figure 3. A goal profile chart of GDD method.
```

- Color green indicates that the goal is in a good status, e.g., the project progress is just as planned.
- Color yellow indicates that small deviation exists on this goal, but everything is still tolerable and under control, e.g., one day delay on one of the milestones.
- Color red indicates that a large deviation may exist on this goal and corrective actions must be taken to mitigate the deviation and its impact immediately.

The criteria and metrics established in the second meeting of launch phase will be used to determine the color of each goal. Goal profile chart can also be used to report to the management. For reference purpose, the reasons to color the bar should also be described in the report. And the owner of each goal has the responsibility to prepare evidence to support color ranking.

During weekly meeting, the team can determine whether a re-launch is needed. Usually there are two typical reasons why the team needs to conduct a re-launch. One might be more new goals are identified during the development phase, e.g., new goals identified due to requirements changes. The other possible reason is that too many red bars in the goal profile chart, which means the status of the project is catastrophic, and the team needs a re-launch for significant adjustment.

c) **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, project experiences can be well documented for future projects.

2) **Roles**

Three typical roles are designed in GDD method: GDD coach, team leader and team member. Their main tasks in different phases of the GDD method are summarized in Table I.

- **a) GDD coach**

GDD coach is an expert of GDD method who is quite experienced with the practices and tools used in GDD. Besides, the coach also needs to possess rich experience with (embedded) project management so that he or she can.
provide the team the guidance on goal elicitation and solution development whenever needed.

**Table I. Main tasks for different roles in different phases**

<table>
<thead>
<tr>
<th>Phase Role</th>
<th>Launch</th>
<th>Development</th>
<th>Postmortem</th>
</tr>
</thead>
</table>
| Coach      | ● helps the team identify project goals  
● helps to develop associated implementation solutions by following the comprehensive approaches | ● assists the team in understanding the goal profile chart  
● helps further planning the necessary corrective actions | ● helps the team to summarize their project experiences and lessons  
● helps the team to identify improvement opportunities. |
| Team Leader | ● leads the team to fully understand the expectations of project stakeholders  
● leads the team to identify project goals  
● leads the team to allocate these goals  
● leads the team to develop a practical and effective solution to each goal identified  
● leads the team to reach consensus on both goals and implementation solutions | ● encourages and motivates all the team members acting toward the project goals using the goal profile.  
● leads the team to identify any risks that may impede the achievement of project goals  
● leads the team to take appropriate corrective actions to remove deviations to project goals. | ● leads the project team to review the work carried out  
● leads the project team to seek improvement opportunities. |
| Team Members | ● participate communication with stakeholders to understand expectations  
● identify project goals  
● raise internal goals  
● allocate project goals  
● develop implementation solutions for allocated project goals  
● participate discussion to reach consensus | ● tracking status of allocated goals  
● identify deviations from allocated goal  
● identify necessary corrective actions  
● identify related risks to achieve allocated goals | ● provide detail information on implementation of allocated goals.  
● participate discussion on project experiences and lessons  
● identify improvement opportunities. |

**b) Team leader**

Team leader acts as the soul of the team. In GDD, a team leader should know how to motivate the development team effectively to ensure the team continuously working toward the goals.

**c) Team member**

All the team members need to be collectively involved in the project management in GDD. GDD requires each team member constantly tracking a number of goals and the corresponding implementation solutions, reporting the status of the goals, and identifying any potential problems. The practice of assigning each goal to a dedicated team member (goal owner) helps to setup the commitment among the team members and deepen their understanding of the goals and project status. Then the owner of the goal has to explain the status, identify potential problems, and suggest feasible solutions of the goal to the whole team in a regular basis such as weekly meeting.

3) **Principles**

Several principles were applied in GDD method to guide practice, i.e. full involvement in project management, team decision, management of dynamics and unity of the simple and complex.

**a) Full involvement in the management**

In embedded system projects, due to constraints of predefined hardware platforms, more goals than regular projects may exist. Therefore, full involvement in the management is highly encouraged in GDD method. All the project goals should not be tracked only by the team leader individually but by all the team members. The allocation of the goals will not change during the development which may enhance the team members’ understanding of their goals in the course. As a result, this will ease the status tracking and solution suggestion.

This principle helps maintain teamwork and motivate team members when all the goals and their status are clear to them. All the team members manage the allocated goals collaboratively; the success of the project relies on the success of the management of the project goals assigned to individuals. According to Victor Vroom’s motivation theory [8], team members can be highly motivated by clear approaches to achieve success.

**b) Team decision**

Common understanding is very important in GDD method. Not only should the goals of the whole project, but also the solutions to implementing the goals be understood by all the team members. Therefore, although goals are allocated among team members, the owner of a certain goal has the responsibility to interpret and manage the goal and the corresponding implementation solutions in the team and seek consensus among the team.

**c) Management of dynamics**

As the project progresses, the set of project goals may change, the status of each goal and the resource to achieve the goals vary over time. The critical point is to make sure that the team is approaching the success, not deviating from it. Based on a timely status tracking of all the goals, the goal profile and report play an indispensable role in making rational predictions and adjustments. Typical adjustments include both identification of new set of goals and development of corresponding implementation solutions.

**d) Unity of the simplicity and complexity**

In GDD, the meaning of goals and their status should be transparent to all the team members as clear as possible so that all the stakeholders of the project can easily reach a common understanding. However, the procedure to this transparency could be very complex. Take the quality goal for example, the status of the goal can be easily understood...
using a goal profile. But behind the visualized presentation of the goal profile, the process to reach the result could be very complex. It may include necessary tools to collect data and sophisticated models to predict the final product quality. Thanks to the research and practice in software process over the years, many tools and models are available to facilitate the above task. In addition, by allocating goals to team members, the goal owner is able to manage the quality goal in a professional fashion.

III. CASE STUDY CONTEXT

DONN Tech is a Chinese software company engaged in embedded system development. It started software process improvement based on SEI’s CMMI [3] model in 2007 and achieved maturity level 3 in 2010. However, with the increase of both project complexity and scale, development teams face big challenges to handle more and more goals simultaneously in a single project. Their original project management methods that were designed for CMMI based process improvement but only focused on tracking the goals in The Iron Triangle (cost, time and quality) [6] cease to be effective. In one case, project team developed an embedded system to support locating of indoor objects using UWB (Ultra Wideband) wireless technology. Using the original project management method, due to the lack of systematic identification and definition of project goals, the hardware performance and software algorithms of the delivered system were not compatible with each other. For instance, neither the software components nor the hardware units were able to deal with undesired signal which is common in an indoor environment due to signal reflection. However, engineers from both parts claimed that the other was responsible for handling the issue. As a result, several pending issues such as insufficient process capability to handler planned number of UWB tags in the locating area; unstable locating results etc., remained in the initial release of the system. The development team spent lots of time trying to fix the issues which led to late delivery and low morale in the end. Two core developers left after the project.

Both the management and staff then agreed that remedial actions must be taken to solve the management issues. The goal driven project management methodology (GDD) was then developed in DONN Tech. Before GDD can be deployed across the whole company, both advantages and disadvantages of the method must be identified. This became the original intention of the study. The iLocation II project then started in Nov. 2009 and planned to be finished in 3.5 months. It was selected as the focused project for the following reasons:

• The initial release of this project reflected a number of typical issues that can be observed in many embedded system projects, e.g., lack of communication between hardware engineers and software engineers, inability to track other important goals other than time, cost and quality.

• The project scale is not large, which may ease the team training and project monitoring in this pilot case.

• This project has no specific customers at the current stage which provides enough space for both the team and the researchers to conduct the study.

However, in addition to this focused project, we also have several other embedded projects applying the GDD method, some of which are still in progress. One author acted as the GDD coach and guided the focused project team throughout the complete project lifecycle. The other authors helped to evaluate the projects results and report the findings.

IV. CASE STUDY IMPLEMENTATION

To conduct the study, the iLocation II project was to improve the old system. The new team included seven full-time developers, two of them were hardware engineers and five were software engineers, among which three software engineers were new to this project.

By following the GDD method defined above, the iLocation II project team started to develop the project. Since GDD method was new to all the team members of the new project, we did 2-day training before the formal kick-off of the project. The training covered GDD method, UWB technology and locating algorithm. Afterward the project launched as planned.

1) Goals elicitation

Led by the GDD coach, the whole team and relevant stakeholders used the method introduced above to identify the project goals. A portion of the goals identified by the project team is depicted in Figure 4.

2) Goals analysis, criteria identification and candidate solution to achieving goals

In order to reach common understanding on the goals identified above, the project team analyzed each goal to clarify its meanings, defined criteria to determine whether or not the goal is satisfied and the method to measure the deviation. Besides, candidate solutions to addressing the goals were also developed based on the common understanding of the goals. Some of the detailed goal analysis results are described in the following sections.

a) Goal of high quality of the system

Elaboration

Figure 4. A portion of goals identified in the project.
The number of defects removed during system testing was selected as the index of software product quality. The rationale is that the less the defects in system testing, the better the product quality.

**Criteria and metrics**

In this project, team set the quality goal as 1.5 defects/KLOC, i.e. 5.25 defects should be removed in system testing given planned size of the final product is 3500 LOC. By using the defect injection and removal model based on yield calculation introduced in [7], the team estimated the number of defect injected and removed in each development phase. Hence deviation can be calculated as:

\[\text{Quality deviation} = \frac{\text{predicted number of defects} - 5.25}{5.25}\]

Where predicted number of defects means the estimated number of defects removed in system testing phase based on actual number of defects injected and removed in early development phases using the defect injection and removal model in [7].

**Candidate solution**

The main solution to address this quality goal is based on a simple divide-and-conquer strategy. The project team divided the final quality goal into several smaller sub-goals, i.e. the estimated numbers of defects injected and removed during each development phase. Achieving the final quality goal accordingly depends on realizing all these sub-goals.

Figure 5 illustrates the quality plan to support this team's quality goal. Phase Yields is calculated as the defects removed during a phase (Review, Compile, UT, etc.) as a percentage of those present at the start of the phase plus those injected during that phase. When phase yield is determined according to historical experiences, the estimated number of defects injected and removed in each phase can be figured out, e.g., the estimated number of defects removed in code review phase is 49.5. In this way, the estimated number of defects in system test phase is 2.11. Since phase yield in system test is 50%, the team can expect 2.11 defects remaining in the final product. Consider the quality goal is 1.5 defects/KLOC in system test phase and planned size of the final product is 3500 LOC, this quality plan might result in a better quality than the original goals.

**Goal of number of tags**

**Elaboration**

The location II system should support multiple UWB tags in the locating area without sacrificing response time, the more the better. In the new system, the expected size of the locating area was extended to nine times as the original area, which became a critical goal to the success of the project.

**Criteria and metrics**

In the project, the team established a goal to support as many as 25 UWB tags in locating area according to customer's requirements and hardware/software constraints. Deviation of this goal can be measured as actual number of tags in locating area to 25.

**Candidate solution**

Solutions to address this goal need the cooperation between hardware engineers and software developers. The team decided to build prototype systems and run simulation in the early phases to verify whether this goal could be satisfied and monitored the goal in the subsequent phases. Besides, to filter undesired signals, the hardware group decided to add a special unit to the system to improve the processing speed.

**Goal of schedule**

**Elaboration**

As required by the organization’s biggest customer, the duration of this project is not negotiable. In this project, the team was required to finish the development within 3.5 months.
Deviation of this goal can be calculated as predicted closure date of the project based on current information of the project to the original planned finish date. Both relative difference and absolute difference should be considered when judging the status of this goal, e.g., the percentage of deviation and the number of days of deviation.

**Candidate solution**

To meet the schedule goal, the team decided to develop the *ilocate II* system in multiple iterations with several predefined milestones. They also planned to establish a progress tracking mechanism based on EVM (Earned Value Management) [9] method. With EVM method, both progress and effort on making progress can be tracked based on real data. Then the project could be warned of the schedule deviations as early as possible and necessary corrective actions can then be taken to mitigate the deviation.

*d) Goal of process discipline*

**Elaboration**

Process discipline required every team member working by following the GDD method and any other necessary process practices during the development. Plus, all the required process data must also be collected as documented procedures.

**Criteria and metrics**

In this project, the team made commitment on process discipline, i.e. all team members must follow the process discipline. Hence the deviation from this goal can be calculated via the actual percentage of staff who violated process discipline.

**Candidate solution**

According to the team’s decision, in the early stage, the owner of this goal led the team to define the processes which would be used in the project. Then, he or she would check the conformity of process discipline daily in the early stage and weekly in the middle and late stages. Besides, the team drew a policy of no weekly meeting without process data. Hence all the team members would be forced to record the process data needed.

e) Goal of satisfaction from all team members

**Elaboration**

Although there were no specific definitions about satisfaction from all the team members, some factors were selected to be considered about team members’ satisfaction.

- Balancing work time and family time
- Pleasant working environment
- Valuable experience
- Effective support from team members
- Workable project plans

**Criteria and metrics**

The approach to measure this goal was a questionnaire-based survey conducted every week. Every team member gave his/her answer (satisfied or not satisfied) about the current project status. Anyone answered ‘not satisfied’ was asked to provide at least one reason. The team established a goal that at least 70% of the team members should feel satisfied, which means that 5 or more engineers ‘satisfied’ with the current project status.

**Candidate solution**

The owner of this goal was responsible to track its status and if there were any significant deviations, the goal owner should report the reason(s) to the project team. The project team then took measures to remove or mitigate unsatisfying causes.

3) Goal profile

As described above, projects applying the GDD method use a goal profile chart to track the status of all the identified goals. Figure 6 illustrates the goal profile chart of the *ilocate II* system project at Week 7. In this goal profile, for example, the first yellow bar from the left means that the quality goal of this project may have some small deviation. This conclusion was drawn based on the actual defects injected and removed by the end of HLD (High Level Design) inspection. Table II presents both the planned and the actual numbers of defects injected and removed in the first four phases, i.e. the REQirement phase, the REQirement INSpection phase, the HLD phase and the HLD INSpection phase.

The actual defects injected and removed were much more than the estimated, which indicates high risk on the project quality goal. However, according to the size data shown in Table III, the size of requirement specification and design specification is much more than the planned size. The planned defect density is 0.625 per page while the actual

![Goal Profile of Week 7](image)

Figure 6. Goal profile of week 7 of *ilocate II* project
defect density is 0.704 per page. It’s a deviation less than 20%. Without more actual defect data to help predict defects in the final system, the color of this goal was determined to be yellow by the team. However, further attention should be paid to this goal in the subsequent phases.

Table II. Number of defected injected and removed

<table>
<thead>
<tr>
<th>Phase</th>
<th>Plan injected</th>
<th>Actual injected</th>
<th>Plan removed</th>
<th>Actual removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQ</td>
<td>2.5</td>
<td>6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>REQ INS</td>
<td>0</td>
<td>0</td>
<td>1.25</td>
<td>5</td>
</tr>
<tr>
<td>HLD</td>
<td>3.75</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HLD INS</td>
<td>0</td>
<td>0</td>
<td>2.5</td>
<td>14</td>
</tr>
</tbody>
</table>

Table III. Size of work product

<table>
<thead>
<tr>
<th>Phase</th>
<th>Plan</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQ</td>
<td>SRS</td>
<td>5 pages</td>
</tr>
<tr>
<td>REQ INS</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>HLD</td>
<td>SDS</td>
<td>5 pages</td>
</tr>
<tr>
<td>HLD INS</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

V. DISCUSSION

GDD method is a goal-oriented approach to organizing and managing embedded system projects. GDD provides an alternative viewpoint to embedded system project management: project consists of a set of goals; the success of the project depends on the coordination and realization of all the goals (monitoring the status of the goals and correcting the deviation if necessary). However, there are still some considerations in both the study and in applying GDD in software projects.

A. Reflections

1) Project result

From many aspects, the development of the ilocate II system can be considered as a successful project. The project finished on schedule with very high quality. No defects have been reported after delivery. This system supports locating more than 40 UWB tags with a precision of less than 1 meter and response time of less than 2 seconds, etc. Besides, the entire team considered this project a valuable experience.

In order to evaluate the GDD method, we collected the feedbacks from both the staff and the management after project. We received both positive and negative comments on GDD method.

2) Feedback from project team

a) “Easy to learn and use”

GDD method is easy to understand and apply in embedded system projects. But more detailed guidelines need to be developed to guide practices, especially on how to develop effective solutions to some project goals. Method coach can be a great help in the initial stage of adoption the method.

b) “Meet project goals better”

It is the first time project team took project goals seriously. In other projects, goals were identified in the early stage, but nearly no practical solutions were designed and discussed to ensure the achievement of these goals. Therefore, project goals remained only on paper and never affected project activities. But in GDD, all project activities were driven by goals and solutions to these goals were carefully designed and thoroughly discussed. Both goals and solutions were constantly revisited during the development process. As a result, we managed to accomplish the project goals better.

c) “Feeling of confidence”

When the team reached consensus on goals and the status of the goals, the team felt confident that the project was under control. Everyone was aware of the project status and the actions to make progress. This feeling of control about the project within the whole team promoted teamwork and improved team morale. With the support of GDD, since goals can be assigned among team members, more project goals can be identified and tracked effectively. The goal tree and goal profile are also able to facilitate the job.

d) “Effective and efficient communications between hardware engineers and software engineers”

In the development of the original system, hardware engineers and software engineers seldom communicated with each other due to their different disciplines. In the new project using the GDD method, both groups were put under the same pressure to reach their common project goals. Therefore, they had to understand both the project goals and status of these goals. By sharing such common understanding of the goals, communications between both parts turned to be more effective and efficient.

e) “Lack of enough guideline on specific practices”

The philosophy behind the GDD method is easy to understand and adopt. But application of GDD method in the first iteration is not easy due to the lack of concrete practices. For example, how to make accurate estimation? How to tune project process to achieve project goals? An experienced coach played a vital role in adopting GDD smoothly at the early stage. In addition, the postmortem phase can support the accumulation of project experience on goals identification and implementation.

3) Feedback from the management

a) “Know status better and become more confident with the team’s performance”

The goal profile presents the project status within one page, which helps all stakeholders learn the project status quickly. The tri-color (green-yellow-red) chart helps them focus on the critical issues of the project. With suggestive discussion on these problematic issues, the management was able to know the status better and aware of how to support the team better. As a result, the management was more confident of the project success.

Meanwhile, some reflections on GDD also need to be considered for method improvements, among which the overheads of applying the GDD method was considerable when first adopting it in a project team.

B. Best practices

This section discusses the learning from the case study. Based on the observed facts and feedbacks from the ilocate
If project, GDD can adapt to embedded software projects and introduce improvements in several aspects:

1) *Facilitate the management of more project goals*

   Embedded system development project teams may have more project goals to be satisfied than regular projects. The goal tree and goal profile facilitate the identification and tracking of multiple goals. Besides, assigning all the goals to team members reinforces the staff’s commitments. Fixing a owner of each project goal also helps to deepen the understanding of different goals.

2) *Promote communications between hardware engineers and software engineers*

   Communications between the two different disciplines are fostered when the engineers from both sides are involved to identify and track the same project goals.

3) *Provide a simple description of project status*

   With the goal profile, all the relevant stakeholders are aware of the project status better and know how to contribute the project better.

4) *Help accumulate project experience and knowledge*

   Actions during postmortem usually help to transform a goal tree to a reusable template, which captures common goals among projects with similar context. This template can be used in the follow-up projects to identify project goals quickly.

   Besides, the way that all the experience and knowledge are attached to certain goals also helps share and transfer project experience and knowledge.

5) *A qualified GDD coach is very important in early adoption*

   In the early stage, an experienced coach who is capable of not only GDD method but also domain knowledge is critical in a successful adoption of the GDD.

C. *Methodological comparison and limitations*

   Capers Jones conducted a survey on project management methods being used in top companies and identified TSP [7] and SCRUM [11] as two best methods in his book [10]. Compared to these two methods, GDD is similar with the practice to identify project goals. However, the extent to which project goals are systematically identified and tracked is different. Goals elicitation in GDD usually is supported by a goal template. Goals profile depicts intuitive status of all the projects. Besides, explicit linkages between all goals and solutions are established and maintained which is usually ignored both in TSP and SCRUM.

   GDD method provides an abstract strategy to satisfy expectations from project stakeholders. The abstraction helps extend its applicability and mitigate intricacy of project activities, i.e. GDD can be adapted to many other project circumstances easily and presents concise status of project on a very high level. However, some shortcomings of this strategy are also obvious. GDD method can rarely work alone. It must be combined with concrete (process) practices to carry out software projects. In practice, each goal identified must be provided with a feasible solution. The concrete practices must be included in these solutions. For instance, it is difficult to apply the GDD method in managing the quality of work products alone. The quality goal must be implemented by a quality solution, in which some quality assurance practices, such as review, inspection, testing and monitoring of quality index, have to be considered. Therefore, the effectiveness of the GDD method will increase when team members experience and accumulate more solutions to certain project goals. The way in which the GDD method conducts postmortem will support the effective aggregation of the related experiences. However, GDD is an open and scalable process framework. The project team has the flexibility to define development steps aiming to achieve project goals. In practice, different process practices can be employed by different team members in terms of their individual contexts. For example, the different disciplines (electronics engineering and software engineering) in the case reported above. This makes GDD method able to be adapted to many circumstances. In this sense GDD method may reconcile the conflicts among different styles of processes, i.e. agile processes and non-agile processes.

D. *Other considerations and limitations*

1) *Research limitations*

   The focused project in this research was at its second release. Therefore, experiences and lessons from the first release may contribute to the success of the project. However, since two core engineers left and three new engineers joined the team, the impact of inherited experiences is limited. Besides, some critical goals (e.g. number of tags, size of locating area, etc.) in second release were so significantly different from the first release that totally new solutions should be developed to address these goals. With GDD, not only experiences and lessons but also new goals can be comprehensively examined so as to make project success more achievable.

2) *Project characteristics*

   In terms of the duration and the team size, i.e. typically 5 to 10 developers within 6 months, the projects we adopted the GDD methods are not large scaled. Perhaps this is one of the reasons why all the completed projects succeeded to achieve their goals in the end. When the scale of the project increases, the number of goals increases as well. Therefore, it may be more difficult to identify and manage project goals. Future empirical research need to be conducted to investigate the impact of project scale to the GDD method.

   Besides, since team’s common understanding and consensus play a very important role in the GDD method, we suggest project using the GDD method better to be co-located. Before having effective approaches to address typically issues in distributed software development, e.g. limited communication channels, culture difference and process difference, the GDD method is not ready for distributed teams at this stage.

VI. CONCLUSION

In embedded system projects, more constraints are involved than regular software projects and combined with more complicated expectations from the stakeholders. To achieve the success of embedded system projects, development teams need to balance and satisfy most of these
expectations. This difficulty challenges many embedded project teams in managing their projects. This paper reports a Goal-Driven Development (GDD) method that was developed to improve the traditional project management methods by focusing on identification and implementation of more project goals. The results from the pilot projects in this research indicate that GDD is able to systematically and pragmatically transform relevant stakeholders’ expectations into the project goals by following a well-defined procedure. With the agreement and commitment on the solution to each goal, the overall project success becomes more achievable. Using the goal profile chart, all these goals can be effectively tracked and monitored in a timely manner.

GDD as the outcome of our attempt to seek and continuously improve project management method for embedded systems development offers a promising approach to managing a number of constraints in complicated settings. Our research at this stage also ends with several interesting questions related to the GDD method which may lead to future research in this direction, such as

1) How to identify project goals more efficiently? The identification of project goals is usually a time-consuming task. Although different project context leads to different set of goals, some common goals can be shared among different projects. Therefore, a general template formulating the common goals may significantly shorten the time needed to elicit project goals. Besides, developing project specific goal templates based on the general template for a particular application/industry sector may also have its research value.

2) How to meet stakeholders’ expectations better? In practice, a predefined priority of the expectations (goals) may mislead the practice. However, the definition of the importance of the project goals, e.g., required goals, expected goals, etc. may have their research value. Research and practice are needed to develop specific criteria to guide this kind of division of goals.

3) Supporting tools for GDD method. An integrated supporting tool for GDD method will facilitate the implementation of GDD in projects. Key features of such a tool may include the following:

- Supporting the management of project goals, i.e. to control the changes to the goals, to keep history of all the changes, to present goals and corresponding solutions clearly.
- Supporting the generation of the goal profile chart. The procedure to generate the chart can be semi-automatic given the diversity of data sources.
- Supporting the management of experiences for GDD method. e.g. goal templates, experience retrieval.

ACKNOWLEDGMENT

This work was partially supported by the Fundamental Research Funds for the Central Universities, Project of Software Engineering Specialty Construction (2007-2201–2203), the Project of Software Engineering Teaching Team (2010-118), and the Project of Software Engineer Education Innovation (2008-042) by the Ministry of Education of China.

Corresponding author: Dong Shao (Email: dongshao@software.nju.edu.cn)

REFERENCES