Software process improvement plan for integration projects

Project INF5180

N.N.

1. Contents

1. Current situation

1.1 What is integration

The topic of this improvement plan is integration projects and the improvement of the process used in these projects. The IEEE Standard Computer Dictionary defines integration as “the process of combining software components, hardware components, or both into an overall system” [1]. The integrations discussed in the improvement plan do not include hardware. The relevant integrations are concerned with “making applications work together that were never intended to work together by passing information through some form of interface” [2], i.e. making two or more independent systems able to communicate and share information.

There are mainly two ways of integrating systems. One solution is to develop an intermediator that all systems are connected to. This means that all systems will only communicate with the component and the component will make sure that the message will reach the system it is intended for.

The other way is to integrate all the systems directly to each other, i.e. one-to-one. This will result in n^2 connections when there are n systems integrated.

Both solutions have advantages and disadvantages. The development of an intermediator is often very expensive and time consuming. Still, it will be easier to integrate new systems in the existing integration. In an one-to-one integration there will be no cost in setting up the integration, but every connection will be more expensive than in the intermediator solution.

1.2 Organization

The integration projects in the suggested improvement plan are carried out in an international company located in the Nordic countries. The company has almost 3 000 employees and delivers systems for economy and administration, like CRM (Customer Relationship Management) and ERP (Enterprise Resource Planning) systems and systems for HR (Human Resource) and salary. Often their customers want to integrate two or more of their systems to improve workflows. Because of the high competition in today’s market there is a desire to reduce unnecessary work and make the workflows as effective as possible. Integration of systems is one solution to this.

On request by the customer the company will then develop an integration for the relevant systems. There are never more than two systems integrated with each other. Therefore all integrations are one-to-one, an intermediator would have been too expensive for this purpose.

A project group is established, usually consisting of people from the development of one of the systems in the integration. A normal integration project last for about three months. One week is spent on requirements specification. In the next 6-8 weeks the integration is developed. Before the product is released to the costumer another 5-8 weeks is spent on testing and stabilizing the product.

The integration is made by integrating this system with the other; there is not a two-way integration. Developers from the other system usually have no interest in the integration project, and would not like to make changes to their own system.

A major part of the integrations that are carried out involve the ERP system.

Most of the systems are old and not built for integrations. They were usually developed a few

years ago when there was little or no concern for integration. This has resulted in systems that are difficult to integrate because they are very different in technology and semantics.

1.3 Identified problems

The management is not satisfied with the amount of time spent on integration projects. Due to a lot of problems, these projects take unnecessary much time. To find out where the problems are rooted the management decided to make a master theses of this problem. The mission was to find out why the integration projects were more time consuming than expected.

Interviews from a total of nine persons that had participated in previous integration projects were conducted. To capture possible differences in different levels in the organization, three persons from each level were interviewed. These three levels are project management, architects and developers. The management suspected that developers and architects sometimes misunderstood each other due to different terminology.

To conduct the interviews a technique called Repertory Grid was used. This is a technique for deriving mental models and is well suited for a problem like this. It is based on elements, like activities and terms from integration projects, and the comparing of these. During the comparing one will get a good picture of how the interviewee feels about the activities, e.g. if there is something that is particularly problematic, and how the interviewee uses the terms.

In this case the interviews revealed a lot of problems concerning the integration projects. The main reasons for the problems are communication issues between developers and architects, limited reuse, high amount of time spent on stabilizing the product and problems understanding the semantics.

- Because each integration is made specific for the two systems there is limited reuse of code. The integrations probably have some similarities even though there are different systems. By supporting reuse of code, and taking this into account in the architecture of the systems, there would probably be possibilities to make the integration projects less time consuming.

- The high amount of time spent on testing and stabilizing the product is probably a consequence of the design specification not being good enough. In an integration project there will also be more components to test than in many other systems because there is always more than one system involved.

- To overcome the problem with understanding the semantics there would probably have to be a standardization of the different variables in the systems or a standard for mapping variables between different systems. For example, a customer number always has to consist of a fixed number of digits where each digit has a meaning that is valid for all systems the company develops. Along with the semantic web there has evolved a lot of methods and techniques to solve this problem, such as semantic annotation and reference ontology. Still, these are methods under development and it will also require a lot of time to set up for this.

1.4 Focus of the improvement plan

Although all the mentioned problems have improvement potential, it will be focused on the communication issue between developers and architects. Due to limited time it would not be possible to improve all of these problems at the same time nor would it be realistic. It is decided that it is the communication issue that has the most potential. Improving this issue may help to resolve some of the other problems as well (e.g. the stabilization issue).

The developers and architects have to agree on interface between the two systems, and how the development of the interfaces is done on the design and the code level. At the moment they spend a lot of time on agreeing on how the interface should look like and how to solve disagreements between the systems. As this is a complicated subject, a lot of time is spent on these discussions.

1.4.1 Conflicts in terminology

The repertory grid interviews showed that there are mainly two problems in the communication between developers and architects.

The first problem is that the developers and architects use the same terms for different concepts. An example of such a term is component. A component is a part of a system, but there is no clear definition of this term. Architects, who work with the whole system, may define a component as a larger part of the system than developers, who work with smaller parts of the system.

Nan Niu and Steve Easterbrook [3] proposes a model with four conditions as shown in figure 1. These conditions represent how the use of terminology and concepts work together. When the stakeholders (being developers and architects in this case) use the same terminology on different concepts, it will lead to a conflict (the lower left square). The desired situation is that developers and architects are located in the upper left square, consensus.

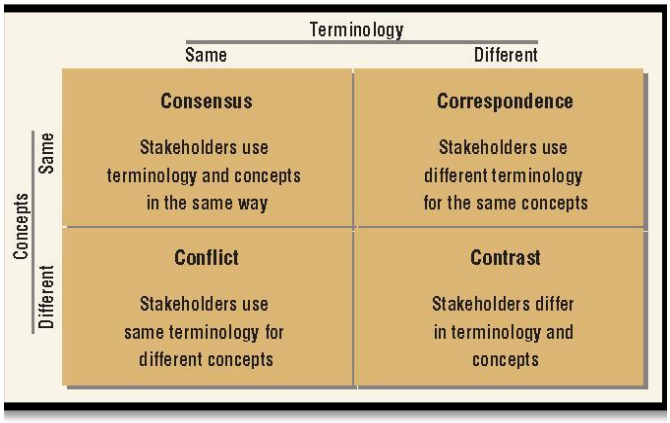

Figure 1

The different usage of terms will naturally cause problems in the specification of the interfaces. Sometimes these misunderstandings will be discovered in the specification phase. If so, a lot of time has probably been spent on the communication before this issue is worked out, but hopefully no harm has been done to the system.

If, on the other hand, the misunderstandings are not discovered until the actual implementation has started, more serious damage might have been caused. This may result in the integration not fulfilling the requirements, or faults in the testing phase.

1.4.2 Problems with decision making

Another problem is that a lot of time is spent on discussing and making decisions about the interface because there is no support for how an interface should look like. Every integration is made specific for the two systems and there is no reuse of such decisions.

There are several decisions that have to be made. Usually, one of the systems is the master, the system that will own the data. Often it is not obvious which of the systems this will be. Another issue is the format and semantics of the messages that will be sent between the systems. It is essential that the other system is able to interpret the information it receives.

There are several international standards for integrations. These standards include protocols for the exchange of messages and information, descriptions of the format of the messages and so on. Several standards have been tried out in the integration projects. However, they are too comprehensive and not always suitable for the integrations. Therefore they have

been rejected and currently there are no standards in use.

Communication regarding decisions made about interface is probably an issue in other development projects as well. The reason why this seems to be a bigger problem in integration projects is that in integrations there are at least two systems that needs to be considered. It needs to be an agreement between the two systems on how the communication between the systems will be. This requires more detailed specification and a closer collaboration between developers and architects.

1.5 How the communication is carried out

In the current situation most of the decisions about the interface are made in the specification phase, which means during the first week of the project. All discussion is usually done in meetings with the developers, architects and other relevant people from the project group. It is normal that the whole project group is located in the same building so the meetings can take place face to face. If this is not the case the communication is done by email or telephone. This is however unusual to happen.

If it turns out to be any difficulties with these decisions later in the project, new meetings are arranged in a similar manner. The previous decisions are revised and new ones are made.

The project members do normally not move their offices for the project, even though they are located in the same building. Since the projects are quite short, the management does not find it necessary to move the project members close to each other.

2. Improvement plan

For the improvement plan the Quality Improvement Paradigm (QIP) will be used [4]. As shown in figure 2 this is a continuous improvement cycle with focus on the organizational learning, project learning and feedback during the project. The focus will be on the organizational level by going through all the six steps and explaining how they will be applied.

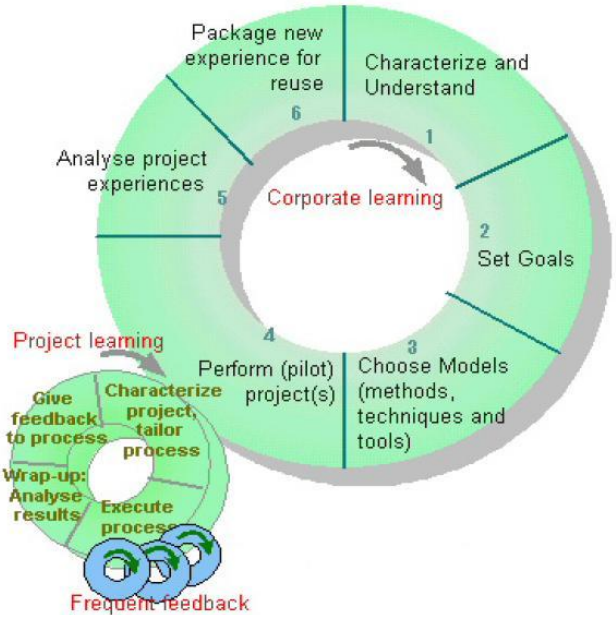

Figure 2

2.1 Characterize and understand: August 1st – August 31st

In this step the organization is characterized and problem areas are identified.

In the integration projects the problem area is already identified as the communication between developers and architects (from the Repertory Grid interviews). The management has also characterized how the communication is carried out. This step is therefore partly already done.

Still, there are some activities in this step that has not yet been performed.

This step also involves preparations, such as establishing a group with responsibility for the execution of the improvement plan, later referred to as the improvement group. The identification of improvement areas was initially started by some of the project managers. It would however not be sufficient to let them lead the improvement project alone. These managers should therefore put together a group that should have the responsibility for the measurements and the execution of the improvement plan.

The improvement group should consist of people from different levels in the organization, e.g. senior managers, project managers, architects and developers. The size of the group should be about 7-8 people and there should be two representatives from each of the levels in the organization, except from the senior management level.

On each level, one of the persons should be involved in the pilot project, while the other should have experience from integration projects, but not be involved in the pilot project. One of the members, preferably an architect, should be responsible for the measurements.

The first step of the improvement cycle also includes looking at data from previous projects. The improvement group should at least collect data about root cause and criticality of faults discovered in the previous projects. All these data should be stored properly in a separate database for later comparison.

It is important to carry out this step properly in the first cycle of improvement. This step will therefore be estimated to take one month and will start up in the beginning of August. It may overlap with the next step because the goals set in the next step may result in a need for new data to be collected.

2.2 Set goals: September 1st – September 14th

The second step in the cycle is to set the goals. In addition to the overall goal for the improvement plan, it is important to address the more specific goals that are possible to verify after the improvement actions have been carried out. Several measurements will be performed to be able to verify the goals.

The overall goal of the improvement plan is to reduce the amount of time spent on integration projects. It is important to emphasize that the quality of the integration should not be reduced. The plan will focus on the communication issue, and by making the communication more effective and provide support for decision making, the idea is that this will contribute to achieve the overall goal.

To set the goal and plan the measurements a goal-oriented measurement method called GQM (Goal Question Metric) will be used. Important elements in this method are goals, questions and measures.

The goal should address the object of the measurement, the purpose, the quality focus, the viewpoint and the context.

The following goal has been defined for the measurements:

Analyze ***the development process***

For the purpose of ***improvement***

With respect to ***effectiveness in communication regarding interfaces***

From the viewpoint of ***developers and architects***

In the context of ***integration project X***

Based on this measurement goal it is possible to derive measurement questions to address what data will be necessary to reach the goal. In this improvement plan the questions are:

1. How much time is spent on communication regarding interfaces?
2. How many faults are discovered in the testing phase that is caused by bad decisions made about the interface?
3. How critical are these faults?
4. How do the developers and architects perceive the communication about interface?

To clarify the knowledge about the communication problem, interviews should be carried out with developers and architects from previous integration projects. These interviews should be done individually, and abstraction sheets should be used as a support. Abstraction sheets are a simplified view of the measurement plan used for interaction with the viewpoints. Important aspects to address are quality focus, variation factors, baseline hypothesis and variation hypothesis. Emphasis should be put on the baseline hypothesis where the developers and architects should estimate what the measures should be.

The interviewees should be developers and architects with experience from integration projects. Whether these people also were interviewed in the Repertory Grid-interviews is not important. A number of 5-6 people are sufficient. The interviews should be carried out by a person from the improvement group with overall knowledge about integration projects, e.g. a project manager.

This step is estimated to last two weeks and will start up in the beginning of September.

2.3 Choose models: September 15th – November 30th

In this step the measurements are chosen and the improvement plan is finalized.

Figure 3 shows the connection between the goals, questions and measurements. The goal is the upper level, the questions on the second level and the measurements derived from the questions are the lowest level.

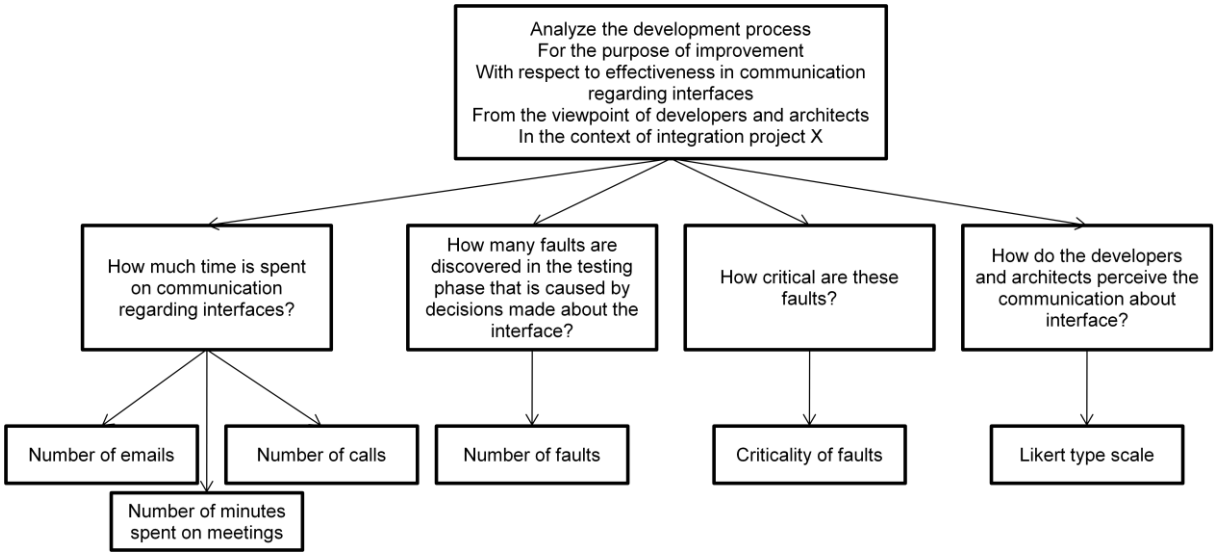

Figure 3

The measurement of time spent on communication will naturally be done by the developers and architects, and collected by the one responsible of the measurements from the improvement group. Each of the developers and architects should register the amount of time they spend on discussing interface. Since communication is done in many ways, there are provided three measurements for this question.

When such communication is carried out on meetings (face to face or telephone meetings), one should count the minutes from the start to the end of the discussion. Telephone calls and emails should be counted separately, i.e. just the number of calls and emails, not the duration.

The measurement of time should be carried out throughout the whole project, although most of the communication will be done in the requirements phase. The improvement group should develop a simple tool where the value for the time measure can be inserted, and the developers and architects are ensured to be anonymous.

The measure of faults derived from communication issues should be collected from the testing tool where it is possible to determine the cause of the fault. This should be performed by one of the testers and is an important part of the testing phase.

How critical the faults are should be determined by an architect in collaboration with the tester in charge of counting the faults. If there are uncertainties this could be brought up for discussion in the improvement group. A scale for how to make these decisions should be defined.

A survey should be carried out to get an understanding of how the communication is perceived by the developers and architects. The survey should address how the

communication is perceived by the developers and architects, and whether they think the improvement actions has helped. The improvement group should manage the survey, i.e. compose the questions, distribute it (preferably online) and collect and analyze the result. The survey should use likert type scales. Naturally, such a survey should be carried out at the end of the project.

To summarize the planning of measurements:

Goal nr	Measure	Source	Collector	When
1	Number of minutes spent on meetings	Developers and architects	The one responsible for measurements	During the whole project
2	Number of emails	Developers and architects	The one responsible for measurements	During the whole project
3	Number of calls	Developers and architects	The one responsible for measurements	During the whole project
4	Number of faults	Testers	Testers	In the testing phase
5	Criticality of faults	Testers	Testers and an architect	In the testing phase
6	How the communication is perceived	Developers and architects	Measurement group	When the project is completed

Table 1

2.3.1 Improvement actions

The improvement actions need to be settled in this step. There will be four improvement actions introduced in the pilot project:

- **Reference ontology**: A common set of terminologies should be established to overcome the problem of developers and architects misunderstanding each other. This could have the same form as a reference ontology from the semantic web. Such a repository should ideally contain all terms in use and a description of the term stored in a database. The reference ontology will be set up by a group of people from previous integration projects. One week should be allocated for this activity.

- **Patterns**: At the moment all decisions regarding the integration has to be made specifically for each integration. An improvement activity that will address this issue is to introduce patterns for how to make these decisions. The patterns introduced in the project should be based on best-practices from earlier projects and created by the same group that is setting up the reference ontology. They should address topics such as which system should be the master, how to solve semantic issues, standards for exchanging messages etc. One week should be allocated to create the patterns, and it should be done after the reference ontology has been set up. As a part of the continuous improvement of the patterns, suggestions to changes should be reported to the improvement group whenever they arise.

- **Review of documents**: To make sure that the design of the integration is feasible, the integration design documentation should be distributed to and reviewed by key persons before the implementation is started. Key persons in this case mean all the

members of the project, perhaps the customer and an architect from outside the project.

- **Location of the project members**: All the members of the project should be placed in the same location, e.g. close to each other in an “open landscape”, to ease the communication.

2.3.2 Selection of pilot project

In this step of the improvement cycle it should also be chosen an appropriate project to apply the improvement actions on. This project should be as representative as possible regarding size and complexity. As all integration projects are approximately of the same level of size and complexity, it would be sufficient to select the next integration project coming up.

This step in the improvement cycle is crucial and there should be allocated enough time to complete the activities in this step properly. It is estimated that this will take about one month and two weeks, and the step will start up in the middle of September.

2.4 Perform projects: December 1st – March 31st

After the planning is done, the plan should be carried out on a project. The improvement actions should be applied and the measures carried out on the decided times in the project. The data collected should be saved in a separate database controlled by the improvement group.

2.4.1 Preparations

Before the pilot project starts, the project members should be informed about the results so far in the improvement plan (e.g. the result of the Repertory Grid interviews) and what the improvement actions are. They should study the patterns and usage of the reference ontology, and there should be guidance by the improvement group if necessary. It is important that the improvement group verifies that this is actually done because the patterns need to be understood before the project starts so that possible ambiguities are clarified. Otherwise this may affect the measurements. Information need to be provided about the other improvement actions as well.

The project members should move their offices closer to each other before the project starts. In this first cycle of improvement, temporary offices could be set up in meeting rooms and so on. Since the project will only last for about 3-4 months, there is no need to move the entire office, but the permanent location of the project members should be the temporary offices during the whole project.

The improvement group should be available for support throughout the whole pilot project. It is reasonable to assume that a lot of questions concerning the patterns and reference ontology will arise.

2.4.2 Feedback

During the project there should be arranged feedback-meetings where the members of the project are updated on the status of the project and measurements, i.e. the measurements that it is possible to provide feedback on during the project. These meetings should ideally be short and frequent, about 15-20 minutes once a week. The meetings should be lead by the person in charge of the improvement group and all the members of the pilot project should participate.

In the feedback-meeting, the data that is collected will be represented and interpreted. One should especially look for any statistical outliers and try to conclude whether they are valid or for some reason wrong. This may reveal that the collection procedures are not good enough. If it turns out that the measure or the procedure is not good enough, changes to the measurement plan may be carried out. Such changes could be to define new ways of collecting the data (maybe set another person in charge of that) or add new measurements.

There may also emerge other change requests to the improvement plan. If these are not critical change requests it is suggested to wait until the next cycle.

These meetings could also be used for exchanging experiences during the project, not necessarily with the measurements in focus. Still, the main purpose of the meetings is to update the projects members, especially the data collectors, on the progress of the measurements.

After the feedback meeting a summary should be written by the leader and distributed to the members of the project.

The duration of this phase will depend on the length of the project. Normally an integration project last for about three months, so it is estimated four months for this step (because it may take longer because of the new actions). The step will start up in the beginning of December.

2.5 Analyze: April 1st – May 31st

In this step the results from the pilot project is collected and analyzed. The results of the analysis should be documented properly. Because data and experience from the pilot project is the essential part of this step, it should be performed as soon as the pilot project is completed.

2.5.1 Analysis of the measurements

When the pilot project is completed, a survey should be carried out by the improvement group. The questions should address how the communication was perceived in the pilot project, if the developers and architects felt there have been any improvements and other viewpoints from the members of the pilot project. Preferably, the questions are accessible online.

The data from the measurements taken during the pilot project should be collected, analyzed and compared to the data from previous projects. The intention is to be able to decide whether there has been an improvement or not.

For the first question, time spent on communication, one should try to decide whether the improvement actions caused a decrease in time spent on communication. It can be hard to draw conclusions on this point as there is no data on how much time that was spent on communication before the improvement actions were introduced. Still, it can be compared to the baseline hypothesis set by the developers and architects before the project started. This will also be a valuable measure for later projects to ensure continuous improvement.

The measurements of question two and three, number and criticality of faults, should be compared to the data collected from previous projects. A decrease in both parameters would be the desirable, but a decrease in only one of them is also a step in the right direction.

The survey will give us the answer to the fourth question, i.e. how the communication is perceived by the developers and architects and whether they feel there has been an improvement.

How the improvement actions are perceived by the developers and architects is the most important result. Because it is the first time patterns are used in an integration project the time measurement may increase due to the fact that the developers and architects are not accustomed to using patterns and reference ontology. Therefore, one should be prepared for that the first cycle of improvement may not give a significant decrease in time. Still, a decrease in number and criticality of faults is preferred.

2.5.2 Postmortem analysis

After the data have been analyzed, a postmortem analysis (PMA) will be performed to capture the experiences from the project. A PMA is an activity done in the end of a phase in the project or when the whole project is completed. The purpose of the postmortem is to identify what were the successes and what could have been better in the project.

A postmortem analysis can be performed in many ways. For the PMA in the integration projects all the members of the pilot project should be invited together with the improvement group. If there are any new integration projects coming up, the members of these projects should also be invited. Since integration projects do not occur that frequent, people that normally are involved in such projects could be invited as an alternative.

The meeting should be carried out with everyone present. If this turns out to be a problem, it is possible to arrange a video conference or have some of the participants on the telephone. It is more important that the members of the pilot project are present. The PMA should under no circumstances be performed by email.

Before the PMA meeting takes place, all the attendants should prepare for the meeting. The improvement group should make a few questions that the attendants should think through before the meeting. The intention of these questions is to make the project members think about how they perceived the project. The questions should therefore address how the project members personally experienced the project, what went well and what did not go well. The questions should be distributed about a week before the meeting.

There should be allocated about 3-4 hours for this meeting. The leader of the meeting should be the leader of the improvement group. The meeting should address these topics in the following order:

- What was done and what was not done (relative to the improvement actions and project plan)?
- What went well and what were the reasons for this?
- What could have been better?
- What problems were dealt with?
- Are there any actions that can prevent such problems in the future?

During the PMA the results of the analysis should also be fed back to the project members and made available for the whole organization. One way of doing this is to put the report on the intranet.

The PMA meeting should result in a summary written by the leader and distributed to the project members. As with the results of the analysis, the PMA summary should also be made available for the rest of the organization, especially for subsequent integration projects.

This step is estimated to last for about two months and will start up in the beginning of April.

2.6 Package experience: June 1st – June 31st

The experiences from the pilot project should be well-documented and made available for subsequent projects. This involves making the experiences applicable to other projects, i.e. discussing whether there have been variation factors that affected this project that will not affect the next.

The result of the previous step will to some extent determine what the rest of this step will look like. If the improvement activity turns out to be a success there may be a recommendation to introduce the improvement actions to all integration projects. Because the improvement cycle is a continuous cycle, there will probably be made some improvements to the actions. Especially the patterns should be updated according to the experience from the pilot project.

If it turns out to be no improvements, a new cycle with new improvement activities may be started.

This step is estimated to last for about one month and will start up in the beginning of June.

2.7 Timeline

Figure 4 shows a Gantt diagram to give an overview of the improvement plan.

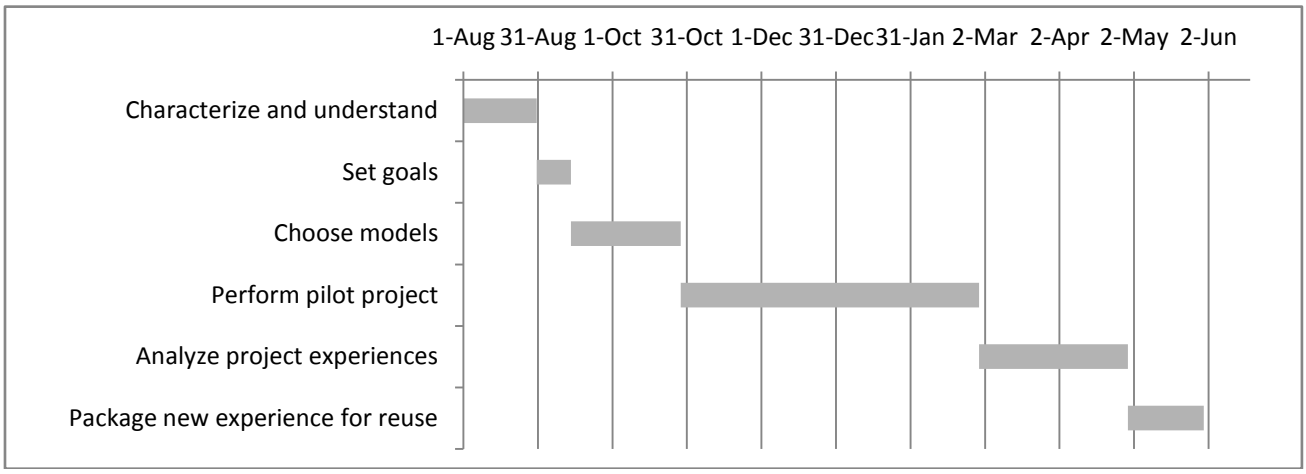

Figure 4

3. Rationale of improvement plan

3.1 Choice of method

3.1.1 QIP

There are several process improvement frameworks that have been evaluated for the integration projects, such as TQM and EFQM.

TQM, Total Quality Management, addresses some of the issues that are important for the improvement plan for integration projects, such as continuous improvement, empowering of the employees and the use of measurements to back decisions. TQM also has a strong customer and quality focus, and cooperation with customers is essential in TQM. This is an important aspect of software development, and one could argue that the customer would appreciate faster delivery of the integration. However, this is not the main focus of the suggested improvement plan, and an improvement method with a strong customer focus would therefore not be suitable.

EFQM, European Foundation for Quality Management, is a method based on TQM. The EFQM-method is used to identify improvement areas and focuses on the organization as a whole. In the case of integration projects the area of improvement is clear and a method with this focus is not necessary. If the improvement plan is a success and it is desirable to also improve other areas in the organization or project afterwards, the EFQM method may be applied. It is however important to focus on the identified problem first.

The QIP (Quality Improvement Paradigm) was chosen as an improvement model because the steps in the QIP-cycle are well suited for the suggested improvement plan. A major part of the work with this improvement plan is the planning. If a plan-do-check-act cycle had been applied, over half of the work would be the planning, i.e. the first step. Therefore it is more appropriate to split this step into three smaller steps.

The QIP model is also appropriate to use together with the GQM-method chosen for the measurements. The steps s*et goals* and *choose methods* will correspond to the planning of the measurements.

The QIP model could be seen as a more detailed version of the plan-do-check-act cycle. However, it also takes into consideration the organizational learning. One can say that this model has three levels: the organizational learning, the project learning and the feedback during the project.

The focus of the plan will be on the organization level because it will only be performed one pilot project. If more pilot projects should have been carried out, it would have been more relevant to follow the steps for the project level as well. Then the experiences from each of the projects could have been analyzed separately.

The feedback meetings planned for the integration project correspond to the frequent feedback suggested by the QIP-cycle.

3.1.2 GQM

There are mainly two approaches to software process improvement, top-down and bottom-up [5]. Top-down approaches, such as CMMI, are used to assess the process to find where the main problems are. This way of doing process improvement is also common for deciding the maturity level of the organization and improvement areas derived from this.

The method is more a bottom-up approach if measurements are applied to a specific problem without deriving them from an overall goal. The GQM method is a combination of bottom-up and top-down approaches. This method is appropriate for the integration projects because the problems are clear, but it is not obvious what the measures should be.

Collection of unnecessary and inadequate data is a typical problem in improvement projects that include measurements. It is difficult to predict what data will be necessary and what will not, simply because it is difficult to predict the future findings in the study. It may also be the case that it is only defined overall goals, and it can be difficult to decide what measures will be needed to verify this goal.

The GQM-method is helpful in such cases because it provides guidelines on how to make clear goals, questions and measurements. Because the measurements are derived from an overall goal, the method provides a stepwise approach to define the measurements, breaking down the overall goal into subsidiary questions that can be addressed and measured.

The goals may also act as an "explanation" to the measures. Developers and architects may be more willing to give away data because it is easy to see what goals and questions the measurements are derived from, i.e. the reason why they should provide data for the measurements.

3.2 Composition of improvement group

The size of the improvement group is set to be about 7-8 people. It should not be over 10 persons in the group. A group of up to 10 people is more effective than a group of over 10 people [6]. The reason for this is that in a smaller group everyone will have an opportunity to be active in the group. For a larger group it is not possible for everyone to express their opinions. In the improvement group everyone should have the same opportunity to express themselves.

The composition of the improvement group responsible for the improvement is important. Practitioners often think that process improvement is the management's responsibility and do not think of themselves as accountable for process improvement [7]. It is, however, the developers that usually have the best knowledge about the technicalities and are closer to the processes. More involvement of developers in the process improvement would most likely be beneficial. One way of encouraging involvement in the process improvement could therefore be to include developers in the group.

Since it is the communication between developers and architects that is the focus of this improvement plan, these should definitely be represented in the group. As stated in the previous paragraph, they are the ones with the most knowledge about the current situation. However, they could have problems seeing the issues regarding their own roles. Therefore, others should also be included in the group. Another effect of involving developers and architects is that they hopefully feel that their interests are taken into account, and that they take part in decisions made about their way of working.

It is planned that there will be two persons from each level of position in the group, where one person at each level is participating in the pilot project and the other is not. This will give inside information and the group will have insight in the project the whole time. Still, there will be people from outside the project to ensure that all activities are carried out in a proper manner.

It is important that all the people in the improvement group have some kind of experience from integration projects so that they have the insight that is needed to understand the problem. Integration is a challenging topic and people that do not have any experience from

this may not understand the problem at hand. The members of the improvement group will most likely also be more committed to the improvement plan if they can benefit from the improvement actions themselves.

The improvement group is also responsible for the measurements. It is important to have a central unit for the measurements and define the measurements properly. If not, it might happen that the members of the project define their own measurements which do not correspond to each other.

3.3 Measurements

3.3.1 Why collect data from previous projects

To be able to decide whether there has been an improvement or not, it is an advantage to have data from previous projects. It is rarely the case that data for all the measurements exist, as it would be very expensive to perform all the thinkable measurements in advance. Later, when the areas of improvement have been identified, one will naturally take improvement actions into the planning process.

A better solution to this is to collect the data that are available from previous projects, and accept that these probably do not cover all the measurements. For the suggested improvement plan it will only be the measure of time that is not available from previous projects. The number and criticality of faults will be collected from a testing tool. The survey reflect the personal opinion of the project members from the pilot project, and since some of these persons have taken part in previous improvement projects, one could assume that they will remember better themselves how it used to be in previous projects.

As for the measure of time, the baseline hypothesis will be helpful in determining if there has been an improvement. In the baseline hypotheses the developers and architects should estimate how much time they think they spend on discussing the interface. This will be important when compared to the actual measure.

The baseline hypothesis will also push the developers and architects to quantify their expectations. Then it will be easier to compare the actual result with the expectations and hypothesis.

3.3.2 Goals

The major part of the measurement goal is self-explaining, except from the viewpoint. According to Briand et.al. [8] “the people performing the development activities affected by measurement will be chosen as viewpoints”. It is mainly be the developers and architects that will be affected by the measurements. Still, it will also affect the testers. This may indicate that the goal should have been split in two goals. However, Briand et.al. also suggests to start with few measurement goals. It has therefore been provided only one goal with the developers and architects as viewpoint. In the next cycles of the improvement the number of goals may be expanded.

Briand et.al. also state that there should be only one viewpoint. In the case of integration projects the focus is at the communication between two roles, i.e. developers and architects, and to look after both parties’ interests we need to include both in the viewpoint.

3.3.3 Subjective and objective measurements

In the measurement plan it will be used both subjective and objective measures. Objective measures do not involve human judgment and will provide the same result regardless of who carries out the measurement. In the suggested measurement plan the number of faults and

complexity of the integration will be objective measures. Subjective measures, on the other hand, will involve human judgment. The criticality of faults and the survey are examples of such measures.

Objective measures will always be more reliable because they will always give the same result for the same situation, and they are therefore easier to compare. Still, in a complex situation like integration projects, objective measures will not provide the whole picture. It is therefore planned some additional subjective measures.

It is important that the objective measurements are done in quite a similar way, e.g. determination of criticality of faults may vary from person to person. To reduce this effect the determination of criticality will always be performed by the same person. He or she may of course consult the rest of the improvement group in certain cases. This will ensure that there is a consistency of the categorization in the project. If the responsibility is transferred to another person in the next project, the consistency will of course no longer apply. However, you should not expect huge differences as there is a scale defined to use in the determination.

3.3.4 Measurement of time

The amount of time spent on discussing interface will be an important measure. The reason is not that the measure in itself is valuable, but it will be very useful in combination with the other measures. The main goal for the management is to reduce the amount of time spent on integration projects. The improvement actions could therefore not be said to be a success if the number of errors has been reduced, but the amount of time spent was twice as much as normal.

Still, one should be prepared for an increase for the amount of time in the first cycle of the improvement plan. It might be an increase because the developers and architects are not used to the improvement actions. Especially the patterns and the reference ontology may require some time to get used to. It is reasonable to expect a decrease in the amount of time spent after the improvement actions have been applied to more than one project.

To measure time might be problematic because it can be hard to determine starting and ending points. When there are meetings it should be quite straight forward to measure this as there are several people involved, and they can all contribute to deciding the number of minutes they think were spent.

As for the telephone calls, the number of calls should be counted. This is not because it is not possible to measure the duration of a call, but measuring the time will require a lot of self-discipline from the developers and architects. It might therefore be difficult to carry through. If it is desirable for the improvement group to have all the measures of time in the same measurement unit, it is possible to add a constant number of minutes per call. It is also possible to distinguish between short and long calls.

The number of emails should be counted to include the communication that is done electronically. It would not make sense to measure the time spent on writing the email as one would probably encounter a lot of variation due to different skills on writing on a keyboard. One might argue that if a person spends a lot of time writing the email, he or she has spent more time on the communication than a person that is quick on the keyboard. Still, this will be personal variations and will not be interesting.

The measure of time will focus on time spent on formal meetings, ad-hoc meetings by the coffee machine or in the hallway are not taken into account. It may seem contradictory that this kind of communication is encouraged, i.e. moving people closer to each other, and yet it is not measured. The reason for this is that it would be very difficult for the developers and

architects to measure the time they spend on this communication. It is also positive that issues are dealt with as soon as they occur, and an increase in this kind of communication may reduce the overall time spent on communication.

3.3.5 Measurement of faults

By counting faults in the integration that are caused by decisions made about the interface, one would get an indication of how well the communication about interface has been. If there have been misunderstandings or disagreements regarding the specification of the interface, this will probably result in errors in the stabilizing phase.

The criticality of the faults should be seen together with the number of faults because it is not desirable to reduce the number of faults if the faults turn out to be more critical.

3.3.6 Survey

There are two ways of measuring how the developers and architects perceive the communication issue, survey and interviews. A survey was chosen to be carried out in the end of the pilot project because there have been a lot of interviews earlier in the project, i.e. Repertory Grid interviews and interviews regarding the baseline hypotheses. It may be hard to get project members to participate in the interviews, and a survey is less time consuming.

A survey will also be easier to compare, i.e. make a syntheses of the results, than interviews because there will be used scales in the survey.

3.3.7 Collection of measurements

To ensure that the data is collected in a proper manner, it is important that it is the right people that do the collection. The whole improvement group will have responsibility for that the measurements are carried out, but each of the measures will be collected by one specific person. Briand et. al. [8] suggests a list of criteria when selecting who should collect the data:

- Expertise: the person needs to have expertise to provide the data accurately
- Bias: the collector should not be biased when collecting the data (e.g. if the data may be used to assess him or her)
- Access: a person involved in the product or process may be the best person to collect
- Cost: the measurement should not increase the costs of the project
- Availability: the person should have enough time to spend on collecting the data
- Motivation: the person should be committed to the measurement program

For the measurement of faults and the criticality of these, a tester will collect the data. He or she will not be biased because the measures will not be used to assess him or her. This will also be the person that has the best access to the data, and the cost will be low because the tester is using the testing tool anyway. Yet, a tester may not have enough expertise to decide the criticality of the faults. This will therefore be done in collaboration with an architect. An architect will probably have more overall knowledge about the integration and be able to use this knowledge when determining which faults are critical and which are not.

The measure of time will be inserted into a system anonymously by the developers and architects, and the improvement group will collect the data from this system. Because the measures may be seen as an assessment of the developers and architects, they may be

unwilling to give away data for this measure. The assurance of anonymity is a way of preventing this. Anonymity will make it impossible to assess individuals, only show the group of developers and architects as a whole.

3.4 Why the improvement actions will work

3.4.1 Reference ontology

Actually, the fact that the developers and architects are aware that they use different terminology may help in solving the problem with misunderstandings. This awareness may lead to more clear expressions. The goal is to make the developers and architects using the same term for the same concepts. One way of doing this is to introduce a reference ontology.

In semantic web an ontology is referred to as an explicit and formal specification of a conceptualization [9]. A reference ontology is then a specification for a shared concept agreed by both parties. This is used to ensure that the communicating systems use the same terminology for the same concepts. A reference ontology can also be used in other kinds of settings, such as integration projects. In the case of the suggested improvement plan the ontology will not be concerned with the communication between the systems, but between the developers and architects.

Along with the terms is a description of what the term means and how to use it. In the semantic web this description is formal and often based on logic because it should be machine processable. There is no need to make the descriptions that strict in the case of integration projects as it will only be interpreted by humans. By using this reference ontology, it is ensured that the same terminology is used for the same concepts.

A reference ontology may also work as a basis for glossaries included in documents (if that is practiced). The ontology will then ensure that the same descriptions of terms are used for all documents.

3.4.2 Patterns

Introduction of patterns is an improvement action that will help in the decision making in integration projects. Hirokawa [10] states that groups that spend some time on defining how to make decisions, make better decisions later. By introducing the patterns that will be used in the decision making process, the developers and architects have defined how to make decisions. They are not restricted to follow the patterns in all cases, but the patterns can act as a support tool in case of disagreements.

Patterns will also provide more structure to the process. Hohmann describes structure as what “defines the form and content of outcomes and supports the processes we use to create them” [11]. This means that we can talk about structure for both the process and the outcome. In the integration projects it is the application of structure to the process that is most important.

There are several criteria to help determine how much structure is needed. If the group is small and the duration of the project is short, a smaller amount of structure is needed than if the group is big and the duration is long. The complexity of the problem will also affect the amount of structure needed. So, even though the integration projects are quite short and there are normally a small number of people involved, more structure is needed because of the high complexity of the problem.

Previously there had been little structure in the process of discussing the interface in integration projects. Decisions had been made for each integration and there were no

guidelines provided on how to do this. The patterns that are introduced are one way of adding more structure to the process.

Settling the interface for an integration can be seen as the process of solving a problem. When we try to solve a problem, we look for similar problems solved before and explore whether these can be used as a pattern. It might seem strange that the same problems are encountered in every integration when the project members often have experience from other integration projects. This is probably a result of the problem being complex and that there are communication issues in the project group.

Patterns are a way of describing how a problem may be solved. Patterns should also include advice on how to adapt a pattern to the problem and in which cases the pattern should not be used. In integration projects the patterns should be based on the experience of previous project members. This way their experience can be used by others with little or no experience. Settling the patterns in advance will hopefully prevent misunderstandings.

An essential prerequisite for the success of patterns is maintenance. This is ensured by the continuous reporting of errors and suggestions from the developers and architects.

3.4.3 Review of documents

The review of documents will imply that all relevant persons will have a chance to express their opinions about the design of the integration. A result of this will hopefully be that everyone's interests are looked after. Still, this is not the main reason for the improvement action. It may be difficult for the developers and architects to see the problems regarding their own roles, e.g. communication issues can be hard to notice. Such issues will hopefully be discovered when others review the document.

3.4.4 Location of project members

In all kinds of projects the closer the project members are located to each other, the easier it is to communicate. The members of the integration projects have normally just been sitting in their regular offices. In some cases this means that they are located quite far away from each other, even though they are located in the same building.

By placing the project members close to each other it will be easier to discuss and solve issues as they occur. This will help to ensure that the issues are dealt with and not just forgotten.

Random discussions between the members of the project might also be overheard by other project members if they are located close to each other. Information from these discussions may be useful if the person overhearing the discussion is encountering the same problems.

3.5 Selection of pilot project

The pilot project in the suggested improvement plan is selected as the next project coming up after the improvement plan has finished. It could look like the selection is done by the "the first and best" approach. A pilot project should normally be a typical project to be able to make sure that the experiences from the project will apply to other projects as well.

However, all integration projects are relatively equal with respect to complexity, duration and number of project members. The reason why the integrations are approximately on the same level of complexity is that they usually integrate one specific system with various other systems. The small variation in these factors will probably not affect the result of the analysis. Another argument for selecting the first upcoming projects is that integration projects do not occur very often, and it may be a long time to wait for the perfect project.

The rarity of integration project is also a reason why the improvement actions are applied to only one pilot project. If the actions should have been applied to more than one project before the analysis, it would not have been possible to draw conclusions within reasonable time.

3.6 Feedback

For the project members to feel that they are a part of the improvement project, feedback is important. When data is collected, the purpose of the measurement needs to be clear. By providing feedback, the project members will see that the data they provide are useful and not just forgotten by the management. This may give a feeling of importance and the further collection of data will be easier. In feedback meetings it is also possible to discuss the data with the project members to get their interpretation. This may result in a new insight and conclusions may be made on a smaller amount of data [12].

Even though it is only the developers and architects that the improvement actions are intended for, all the members of the project should be included in the feedback meetings. This is important because data are collected throughout the whole project.

The feedback meetings are planned to be quite short. This is mainly because the integration projects are quite short (about three months) and the meetings should not take up too much of the time. It is important that the members of the project participate in these meetings because of the reasons given above. If the meetings are long lasting and hinder the project members in doing their job effectively, they might not show up on the next meetings.

Short meetings will also force the participants to be effective, and therefore they will not affect the productivity as much.

The thought behind these short meetings are derived from the agile development method Scrum. There they have 10-15 minutes meetings each day to discuss the work ahead of them. The feedback meetings will not happen as frequent and will maybe last a little longer, but the idea of frequent status meetings is the same.

In Scrum, these meetings serve a teambuilding purpose [13], and in integration projects they will hopefully result in more commitment to the improvement program. The meetings are therefore beneficial for both the management and the project members. The leader and other members of the improvement group will get important feedback on the measurements and the overall improvement plan, and the members of the project will have the opportunity to express their meanings and be involved in the improvement work.

3.7 Research methods and analysis

To investigate the effect of the improvement actions Briand [8] et.al. suggest two main empirical investigation methods to check whether the improvement actions has helped, case study and experiment. In this improvement plan the improvement actions will be checked during a pilot project and a light-weight case study will be carried out on the project. This method is appropriate because the area of problem is complex and only a field study will be adequate.

An experiment is more suited for a less complex problem where it is possible to control the variation factors. It would also be difficult to check whether the improvement actions had helped or not. The faults are normally discovered in the testing phase which would not be possible to incorporate in an experiment because it would take too much time.

The strengths of a case study are that there are relative low costs and that they can be applied in the field. This is appropriate for this case as the improvement actions can be

applied to a project that is to be carried out anyway.

There are, however, also some weaknesses of a case study. No statistical results can be drawn, it requires a lot of knowledge about the case and domain and it may be difficult to ensure that the case is representative. In this case the last two weaknesses will be of less importance. As the case study will be performed by the improvement group where there are employees with experience from integration projects, it is ensured that they have enough knowledge about the domain. Since all integration projects are approximately of the same level of complexity and size, there will not be a problem to find a representative project to study.

3.8 Postmortem analysis

To be able to benefit from the experiences from the pilot project, it is important to capture these experiences and make them available for subsequent projects. In process improvement it is not important how this is done, the essential part is that the project members reflect on what went well and what did not. Learning from past experiences is an essential part of process improvement and is necessary for continuous learning.

For the integration project the postmortem analysis (PMA) method has been chosen. By following a defined method one is more ensured that the analysis actually will be performed and that all necessary steps are done. The postmortem is a method that is simple and easy to learn. It requires no external learning, yet it provides explanations on how to carry out the analysis.

3.8.1 Transferring of knowledge

The experience needs to be leveraged from the individual level to the organizational level to benefit from the pilot project. Nonaka and Takeuchi [12] suggest a model for the different types of knowledge that show how the knowledge is transferred and developed in an organization.

As shown in figure 5, there are two kinds of knowledge, tacit and explicit knowledge.

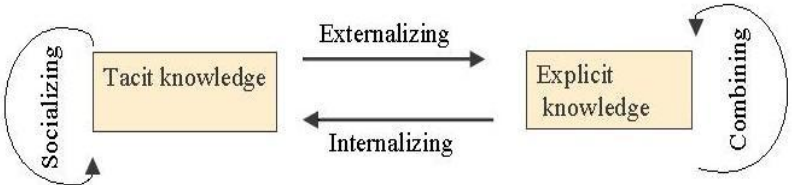

Figure 5

The explicit knowledge is knowledge that is possible to write down and which is easy for others to use, e.g. a description of how to perform a task. Tacit knowledge, on the hand, is knowledge that people have embedded and affect the way they act and perform tasks. This kind of knowledge is also referred to as skill.

In a PMA the important concepts are externalization and socialization. By sharing their knowledge and experience from the project with the other participants in the meeting, the project members externalize their tacit knowledge. This way, other people can benefit from this knowledge and use it in their projects. By listening to others and see how they have solved their tasks in the project, the other participants in the meeting will acquire tacit knowledge through socialization.

3.8.2 Execution of the PMA

The reason why it is important that the PMA meeting is done face to face is that communication and learning is more effective when people meet physically. Learning by socialization is not possible when communicating through email. Figure 6 shows how the form of communication varies with the effectiveness of the communication [12]. By

conducting a PMA through email the possibility to discuss is reduced and misunderstandings may not be discovered. As we can see in figure 6, communication on telephone is more effective than email. This is why it is okay for some of the participants to attend the meeting on telephone.

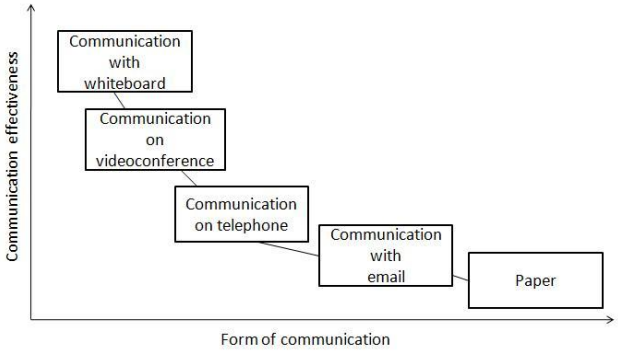

Figure 6

Still, a video conference will be slightly more effective. As communicating face to face with a whiteboard is the most effective way of communicating, this is the preferred way of conducting a PMA. It will then be possible for the participants to discuss and gesticulate. The learning may also be more valuable and misunderstandings are more likely to be discovered.

Before the meeting is held, the participants will receive a few questions from the improvement group as a preparation for the meeting. This is an optional part of the PMA method. This is incorporated in the suggested improvement plan because this is a way of achieving more reflected answers from the members of the pilot project. By being able to think through the questions in advance of the meeting one is also more ensured that all the participants express their personal meaning. If the participants do not prepare for the meeting it is a possibility that people that are insecure may just agree with the others. It is also possible that people will not give their own experiences and opinions a thought, but rather agrees with the others' experiences.

By incorporating homework, the risk that this might happen is not eliminated, but hopefully reduced.

3.9 Package experience

It is important to package the experience so that it can be used in later projects. An important step in this activity is to identify incidents that will probably not occur in the subsequent projects. For example, if one of the architects became sick, it would probably have a huge impact on the integration project. This may have a negative effect on the measurements. If such variations are not taken into account when the experiences are packaged, the results may be interpreted wrong be the subsequent projects.

The improvement actions and measurements should be maintained and updated. If this is not done there will not be a progress in the improvement.

3.10 Discussion of the schedule

For each step in the improvement cycle the duration time has been estimated. The first step, where the improvement group is established and data from previous projects are collected, is estimated to last one month. The reason for this is that it may be difficult to predict how long it will take to establish the improvement group. It is provided quite strict guidelines on the composition of the group and it may be difficult to find the right people that both are willing and have time to join the group.

The collection of data from previous projects will probably be a small part of this step. It is not much data that needs to be collected, but if problems are encountered, e.g. people do not want to give away the data, it might take longer than one month. However, this is not likely to happen.

The second step, set goals, is estimated to last two weeks. The major part of this step,

defining the goal and questions for the measurements, has already been provided. Still, the interviews have to be carried out. Since it is only 5-6 people to interview, two weeks will be sufficient.

For the third step, one month and two weeks has been allocated. All the measurements and improvement actions have been provided, but it might take some time to prepare for the pilot project. A group of developers have to develop the system for collecting the measurement of time. It will not be a complicated system, but it should be made in a proper manner to ensure that it will not spoil the measurement.

The same applies to the other improvement actions, it might take some time to set up the reference ontology and it should be made sure that it is possible to locate the project members close to each other. As the offices will only be temporary, one month and two weeks should be enough for all these activities.

A normal integration projects lasts for about 3 months. As the new actions probably will take some extra time, it is allocated four months for the execution of the pilot project. It is difficult to predict how the project members will adapt to the improvement actions and how much time will be need for this. Still, one month extra should be sufficient because if too much time is allocated the project group may become ineffective.

The analysis of the measurements and the pilot project is an important step in the improvement cycle. The step has therefore been estimated to last for two months. The analysis of the measurements will probably last for at least a month since there it quite a lot of data to analyze. It may also require a lot of time to discuss and draw conclusions from the data. There should also be enough time to make the preparations for the PMA meeting, e.g. create the questions that should be distributed, make graphical notations of the data and so on.

If the analysis of the pilot project has been carried out properly, the last step in the improvement cycle should not take long. The packaging of experience is crucial for the subsequent projects to benefit from the experiences. Still, it may be tempting to carry out this step in an inadequate manner if there have been delays in the improvement plan. It is therefore allocated one month for this step to ensure that it is done properly.

3.11 Aspects left out of the improvement plan

Risks and costs have not been discussed in the improvement plan. These are of course important aspects in process improvement. However, there are quite low risks associated with the suggested improvement plan. All the improvement actions, except from the relocation of the project members, are supporting actions that are carried out in addition to the regular activities. It is also allocated some extra time for the project, so there will hopefully not be any deadlines exceeded etc.

The improvement actions are not expensive compared to other possible improvement actions, e.g. introduction of a new tool. The patterns and reference ontology are developed in two weeks and if it turns out to be no improvement, there have not been huge losses.

It has not been planned training in any of the methods used in the improvement plan. This will probably not be necessary because information about most of the activities that may need training has already been provided, e.g. the GQM-method. It is possible to plan training in the GQM-method if it is desirable to expand the measurement plan.

4. References

[1] “IEEE standard Computer Dictionary – A compilation of IEEE Standard Computer Glossaries 610”, *The institute of Electrical and Electronics Engineers*, New York, USA, 1990

[2] Gulledge, T. “What is integration?”, *Industrial Management & Data Systems. 2006;106:5-20*

[3] Niu, N. and Easterbrook, S. “So, you think you know others’ goals? – A repertory grid study”, *IEEE Software. 2007; March/April. Page number not provided.*

[4] Lecture notes in INF5180, spring 2008 part 12 slide 10.

[5] van Solingen, R. and Berghout, E. “The Goal/Question/Metric Method: a practical guide for quality improvement of software development”, *McGraw-Hill Publishing Company,* Berkshire, England, 1999.

[6] “Reference for Management - Encyclopedia for Management” http://www.referenceforbusiness.com/management/

[7] Baddo, N. and Hall, T., “Practitioner Roles in Software Process Improvement: An Analysis using Grid Technique”, *Software Process Improvement and Practice. 2002;7:17-31*

[8] Briand, L. C., Differding, C. M. and Rombach, H. D. “Practical Guidelines for Measurement-Based Process Improvement”, *Software Process Improvement and Practice. 1997;2*

[9] Antoniou, G. & Harmelen, F., “A semantic web primer”. *The MIT Press*, 2004

[10] Lecture notes in INF5180, spring 2008 part 4 slide 60.

[11] Hohmann, L. “Journey of the Software Professional: A Sociology of Software Development”, *Prentice Hall PTR, 1997.*

[12] Dybå, T., Dingsøyr, T. and Moe, N.B., “Praktisk prosessforbedring – En håndbok for IT-bedrifter”, *Fagbokforlaget Vigmostad & Bjørke AS,* Bergen, Norway, 2002.

[13] Rising, L. and Janoff, N. “The Scrum Software Development Process for Small Teams”, *IEEE Software. 2000; July/August. Page number not provided.*