Experience from process improvement in a SME

Hans Jørgen Lied, M.Sc.
Telenor Geomatikk AS, Trondheim, Norway
Tor Stålhane, Ph.D.
SINTEF, Trondheim, Norway

Introduction

This paper describes a case study that was carried out in 1997-1998 and supported by the Norwegian national research program SPIQ. The company involved wanted to find a way to identify problem areas in the software development and to reduce the amount of rework caused by the introduction of errors during software development. The case study was motivated by the challenge of finding a way to identify problem areas by introducing the process improvement framework offered by SPIQ.

Our opinion is that the best way for a company to improve is through learning from their own data and experiences. In order to perform data analyses, we therefore have to combine collected data with experiences that are available in the organisation. We need to decide what to measure and how, how to make sense of the data and how to use the knowledge. From SPIQ we got the idea to use a combination of three methods:

- Goal Question Metrics (GQM), to decide on what and how to measure.
- Pareto analyses, combined with a robustness analysis to analyse the collected data and to assess the confidence we should have in them.
- Root Cause Analyses (RCA) [8], augmented by use of the Ishikawa diagram [7], to identify the root causes of our problems.
The results from these analyses identified the main problem areas. The feedback sessions in the GQM method were used to combine data, prior knowledge and experience during data analysis. Our approach and results will be useful for other organisations facing a similar challenge.

The collected data open for several interesting analyses such as how the differences in the projects influenced the types of errors, their corrections cost and time to discovery. We will in this paper, however, only focus on how to improve the development process so that we get fewer errors in the future. The rest will have to wait.

The rest of this paper is organised as follows. The first section describes the SPIQ program. The next section describes the organisation and projects on which this case study was performed. After this we describe the approach we chose. Next we describe the analyses we did and the results we obtained. Finally we present some conclusions.

**Software Process Improvement for better Quality (SPIQ)**

The SPIQ program was started in April 1997 and is sponsored by the Norwegian Research Council (NFR). Its main goal is to “increase the competitiveness and profitability of Norwegian IT-industry through systematic and continuous process improvement”. The SPIQ program is based on the process improvement principles of Total Quality Management, [1], [3], GQM and the experience factory [5]. See http://www.geomatikk.no/spiq for more information about SPIQ. The process improvement offered by SPIQ builds on three pillars:

- The PDCA – Plan, Do, Check, Act – cycle.
- All decisions should be based on facts – observations and experience
- Developer participation is an essential part of all process improvement activities.

The work described in this paper has benefited from SPIQ in several ways:

- SPIQ has provided valuable method support
- SPIQ has motivated and partly financed the internal work of introducing process improvement methods to the company
- The methods, experience and changes were discussed at the SPIQ meetings

**Environment**

The company where we did this case study is a medium sized Norwegian company. It has a total of 270 employees with offices in Trondheim, Oslo, Tønsberg and Kristiansund. In addition they have offices in Sweden. The software development division at the present employs 14 persons.
The software development division is of a size that makes it possible to easily adapt to changing market demands. At the present the market wants tailor-made systems and special changes performed on already existing software products. There is a large amount of companies that deliver solutions in the same area and the competition is fierce.

Projects are often undertaken, based on a fixed price, which demands a streamlined development process and little room for rework caused by the introduction of errors. Systems with few or no errors are also important as a way of selling stability and reassurance to customers who invest money into software development. A happy customer is likely to return.

The two development projects were chosen because they were quite different and, at the same time, representative for projects this company is involved in. The only parts that were identical in the two projects was the project team and that they used the same development process model. The differences are much more prominent. The most important ones are:

- Different methods of development; Project 1 had its structure given right from the start, while Project 2 depended on prototyping.
- The customer defined quality assurance requirements for Project 1.
- Project 1 started with a readily made requirements specification from the customer, while Project 2 started with an idea from the customer and the requirements specification was a co-production between the customer and the developers.

**The role of GQM in Root Cause Analysis**

**Why we chose the GQM method**

Our first problem when starting to use RCA was the lack of a method for collecting data to be used. There was no direct support for this in the SPIQ framework but we found a way of combining parts of GQM – mainly the GQM abstraction sheet - with the Pareto analyses that proved to be efficient. In addition to the abstraction sheet, another important reason for using GQM was its strong focus on developer participation. We dropped the lower part of the GQM abstraction sheet – the part that contains the hypotheses concerning values of the focus parameters and the hypotheses concerning the impact the variation factors had on the focus parameters. To fill in these two quadrants would have been interesting, but was dropped since the data later should be used in an RCA setting.

The GQM abstraction sheet was used to manage and structure the brainstorming sessions and to identify the causes we would like to analyse in the later RCA. By getting a strong participation, we hoped to obtain two things:

- More reliable data. The developers should presumably be more interested in obtaining correct data if they felt that they had ownership to the goal and purpose of the data collection.
More interest in the results. One of the factors that can really kill an SPI initiative is lack of interest from the participants. By involving the developers from day one, we hoped to create this interest and thus keep the improvement program alive.

In addition, we would get a better set of basic root causes to analyse if we tapped the large amount of experience that is available from the developers. Without this knowledge, we would have been forced to collect data on many more possible causes, thus creating a large and possibly unwieldy model. This would have cost extra resources without adding anything to the value of the final results.

The GQM process

Since none of the project participants had any prior knowledge of GQM, we started with a two hours GQM workshop. This was done with the help of research scientists from the SPIQ program. As in earlier cases, it turned out that GQM was easy to understand and use. During the two hours session we were able to describe the method and work through the goal that the company had defined. We use this as a standard approach, as opposed to start with a toy example. One of the strong points with GQM is that it tries to maximise the use of expert knowledge. A toy example will thus not help the users to understand why GQM is a good idea.

In order to get a practical set of failure causes categories, we started with the following GQM goal: Analyse the reported failures in order to understand causes for failures seen from the developers in the X project. The GQM process produced a GQM abstraction sheet, which was converted to a data collection form. This was done by converting all questions to entries in the form. This process was straightforward and gave a practical data collection form. The data form represents the metrics in GQM.

After the initial process only small adjustments were needed. This was done in the first feedback session, where component complexity was added as a failure cause. The collection form had the following categories for reporting a failure:

Focus – main cause
1. Incorrect use of cut-and-paste from old code
2. Incorrect reuse of old code
3. Incorrect use of language features
4. Incorrect use of library components
5. Incorrect attempts to make the code general - prepare it for later reuse
6. Attempted reuse by introducing global variables
7. High component complexity
8. Large, unanticipated variation in input data
9. Incomplete or bad test data
10. Too large code components
11. Missing or incomplete configuration management
12. Incorrect component integration
13. Insufficiently detailed requirements specification
14. Wrong code logic
15. Errors in hardware – software communication

Variation factors – secondary causes:
1. Time pressure during development
2. Missing competence in use of programming language or development tools
3. Uncontrollable, external disturbances – for instance fire fighting old systems
4. Errors in development tools
5. Errors in libraries and subsystems supplied by subcontractors
6. Lack of user interaction, which caused lack of understanding of user needs
7. Lacking or missing motivation among the developers
8. Incomplete project model, for instance too few check points in the project

The item numbers used in the lists above are the same as the once used in data collection form shown in Figure 1 below.

In order to get a more detailed picture of the reported failures; some of the categories were split up into subcategories. It turned out, however, that this did not add important information, and these subcategories were dropped from the later Pareto analyses. They were, however, useful during the RCA process when filling in the Ishikawa diagram.

In addition to the possible causes, we registered for each reported failure; its degree of seriousness, consequence and resources needed to correct it. The seriousness was scored on a five point scale, the resources needed for correction was registered in person-hours, while the failure consequences were registered in free text format. The final form – in Norwegian – is shown below.
**Experience from process improvement in a small company**

Figure 1: Data collection form – in Norwegian

**What is our experience**

The use of GQM worked as intended and we were able to create interest and ownership for an improvement initiative. The failure cause categories that came out of the GQM process were practical and useful. A good indicator of its success is that only one failure cause was added later and none of the original failure causes were removed.

**Data analyses**

**Pareto analyses**

Vilfredo Pareto, an Italian Bachelor of Commerce who lived in the nineteenth century, did research on poverty in Italy. He discovered, from extensive statistical material, that approximately 20% of the people in Italy owned or controlled 80% of the country’s total resources. This 20/80 distribution has proved to be applicable in many different environments, among them sources of failures in software development, and is known as Pareto’s Law.

The Pareto analysis is a deceptively simple, yet powerful, way of looking at data to help find the root-cause of a problem. We used a Pareto analysis on our collection of data to identify areas of improvement rather than direct problem removal. Our goal was to identify the main problem areas so that they could serve as a basis for a root cause analysis applying Ishikawa diagrams in combination with a brainstorming session.

The Pareto diagrams are informative in the way that they visualise the main problem areas. The diagram also fits in nicely with our experience that plotting is the best way to start any data analysis process [6]. Below we show the diagrams for the two projects in the case study. This shows us which main causes are prominent in this study. The cause numbers along the x-axis in each Pareto plot refer to Figure 1.

![Pareto Diagrams for Two Projects](image)

Figure 2: Pareto diagrams for the two projects. Project 1 to the left.
Experience from process improvement in a small company

We found that combining Pareto analyses, brainstorming sessions and Ishikawa diagrams in a root cause analysis to be an efficient method for identifying problems and analysing problem areas.

Robustness analyses

The amounts of data collected in this case study is small. The reason for this is that the company is small and with only a few projects running at the same time. We can not collect data over a long period of time, because collected data will soon be outdated or irrelevant due to changes in the company’s environment. We therefore chose to extract information from the collection of data and convert it into improvement actions without collecting large amounts of data needed for a traditional statistical improvement approach.

Since we had few observations, each data point can be critical for the conclusion. We thus introduced robustness analysis to assess the confidence we should have in the data – and thus in the conclusions.

Broadly speaking, a robustness analyses is done by checking whether the conclusion changes when one of the observations was removed from the data set. To do this we created new data sets consisting of one observation less than the total data set. The amount of new data sets is equal to the total amount of reports. They are all different because there is a different observation missing in each of the new data sets. In the first new data set, we removed the first observation. In the second new data set, we put the first observation back and removed the second observation and so on. In figure 3, the letters A, B, C and so on marks these data sets.

By plotting all the new data sets into the same Pareto diagram, we can validate our conclusion. If all the columns are identical and they are identical to the original Pareto diagram based on the total data set, we have a robust conclusion. The total set of data gave us the Pareto solutions presented in figure 2. Combining all the sets from each project for the focus cause gave us the following two Pareto diagrams:
Figure 3: All data sets presented in one Pareto diagram for each project to see if it is robust. Project 1 on top, Project 2 at the bottom.

In our case several of the new data sets indicated a different conclusion. The observations removed from these data sets are to be considered critical to the conclusion because of their ability to change the conclusion. An example from the graphs above are set K – indicated by an arrow - from Project 2 which show causes 13, 14 as a result instead of causes 13, 15, 14.

The number of new data sets giving us different results can tell us how much confidence we should have in our conclusions:

\((\text{Total amount of observations} - \text{amount of critical observations}) / (\text{Total amount of observations})\) which gave us 81% in both projects.

The critical observations should be checked one extra time. We did this by going back to the person who filled out the error report asking him to check it. This check fortunately confirmed all the critical data.

After analysing both projects as described above, we ended up with the following main problem areas:

Project 1:
- Incorrect use of language features
- Incorrect use of library components
- Wrong code logic

Project 2:
- Incomplete or insufficiently detailed requirements specification
- Incorrect component integration
- Wrong code logic

All the identified main problem areas were then analyzed by using an Ishikawa diagram.

**Root cause analysis**

We had two main requirements for our choice of method for data analyses in this project. It should be simple to use with no - or almost no - training needed in advance and it should encourage and facilitate developer participation.

The result was a two pronged approach. We decided to use the Ishikawa diagram for documentation and visualisation. The use of the Ishikawa diagram should take the results from the Pareto analysis as its starting point and then be combined with general brainstorming techniques.

**The Ishikawa diagram**

The data analysis in the improvement project was done by using the Ishikawa diagram. There were two reasons for this: First and foremost, this method is well known in industrial activities. It is easy to understand and apply and it is possible to
identify improvement possibilities once the diagram is finished. Secondly, the method is well suited for developers' participation, which we consider to be an important feature in any SPI activity.

The Ishikawa diagram has been used for many years in improvement activities in a wide variety of industrial settings. It is part of the seven old TQM tools and there is a large amount of literature that covers how the technique can be used in an efficient manner. The method is application area independent and could thus be used without any modifications or adaptations.

An example of an Ishikawa diagram is shown below. The horizontal line – the trunk - is the problem that we want to analyse. In this case it is “incomplete requirements”. The main branches are filled in with the main causes as the participants see them. The next level horizontal lines represent possible causes for the main causes and so on. When the brainstorming session is completed and the Ishikawa diagram is finished, we go through all the identified causes and move or delete causes as necessary. As a last activity, the most important main causes should be identified and put up as candidates for improvement activities.

![Cause and Effect Diagram](image)

Figure 4: Cause and Effect Diagram from one of the brainstorming sessions

**Brainstorming sessions**

Brainstorming sessions are well organised, structured meetings integrating the project team and the measurement team. The collected data are presented to and interpreted by the project team members. Combining the knowledge of the team members with available data, we hoped to identify the root cause.

Since none of the project participants had any prior knowledge of RCA, we started with a half-hour RCA introduction. As we have observed in earlier cases, it turned out that RCA itself was easy to understand and use. During this first half-hour session we were able to describe the method and work through a simple example.
Experience from process improvement in a small company

One of the strong points of a brainstorming session is that it maximises the use of expert knowledge in combination with the collected data.

After this short introduction to the project team on what we expected and what we had found, we started the brainstorming session. The results from this two hours plus session were summed up in a report that was used as input to the post-mortem analyses at the end of the two development projects.

Results

The splitting up of the possible causes into focus and environment, forced upon us by GQM was of great help in setting up the Ishikawa diagram. The secondary causes identified during data collection was used as a starting point when identifying the second level horizontal lines representing possible causes for the main causes.

When we felt confident that the right root causes had been identified, we started to work on the improvement steps. This last step proved to be the easiest one. The discussion among the participants in the feedback session was already going high on what to do to improve the situation in the company.

One of the first things that came up was sharing of competence and experience within the company. Many employees with seniority have a lot of experience that can be shared with the rest if the employees. By sharing this information through courses, lectures or written reports, valuable information will be spread in the organisation. Another thing that came up was availability of information, templates and routines. The lack of an Intranet was also mentioned, and this was one of the points discussed where everybody agreed. Other improvement actions from this session are presented in the list below:

- Sharing of competence internally
- Checklist on Intranet
- Common routines/solutions for standard tasks (best practice experience)
- Common dynamic document templates
- Start the planning, design and development of an experience database

The next activity will be to look at cost and risk in implementing one or more of the identified improvement activities. This is beyond the scope of this case study but work is in progress in the company.

The success of new improvement activities will depend on the participation of all the developers in the company. Our experience is that this method of introducing process improvement to all the involved parts in an organisation increases the probability of success.

Conclusions

We started on this case study with an idea that we could do magic for the company by just waving our magic wand – the improvement framework originally proposed by the TQM fathers. It was not that simple, in fact it was not simple at all. There is a lot
Experience from process improvement in a small company

of “trial and error” behind this case study, but the important part is that we learned, and that the company learned.

It is important that those who shall collect the data participate in the improvement process right from the start. Introducing the participants to the necessary parts of GQM, and involving them in defining what to measure and how to measure it, motivated the participants and thereby increased our confidence in the collected data. The concrete focus on improvement we set in this session had a positive impact on the improvement culture in the organisation.

It seems clear that we seldom or never will get enough material for solid statistical analyses in a company of this size. By introducing robustness analysis as an enhancement to the Pareto analysis, we found a way to assess the confidence we could have in the data and thereby the conclusions.

In order to identify the main causes in the development process, we held a brainstorming session supported by Ishikawa diagrams. This gave us a new opportunity to involve the project teams in a discussion on what were the main causes and what the possible causes to these main causes could be. In addition to giving us the root causes it also gave us a lot of suggestions as to what to do about it.

As a result of this case study the work started on creating an Intranet site. It was decided that a set of pages on this site should be dedicated to points on the list shown above. One person was also dedicated to keep the site alive. All together we can summon up what the company learned from this case study into:

- Individual experiences for the team members
- Measurements gave knowledge of how things really are in the company
- Common understanding of what we learned from this knowledge
- Explicit shared knowledge about the process that the company can reuse

The most important experience, however, is that it is possible to improve the process through data analyses and the use of simple problem solving techniques.

References


Experience from process improvement in a small company


[8] Stålhane, T., Forbedring gjennom årsaksanalyse (RCA). SPIQ technical paper (in Norwegian)
Experience from process improvement in a small company

**Fjellanger Widerøe AS** is a company based in Trondheim and Oslo. They have 270 employees and their main area of expertise is aerial photography and digitizing of these. The photos are used for making digital maps. They have a group of software developers working on internal development projects of among others WinMap. For further information see www-fw.no

**Telenor Geomatikk AS:**
Telenor is the name of the public limited company Telenor AS including branch companies. With its 18400 domestic employees, 2700 employees abroad and a 1998 turnover of NOK 28,4 billion, Telenor is among the largest and most value adding companies in Norway. Telenor Geomatikk AS is owned by Telenor IT-service AS and is the main operator of Geographical Information Technology (GIT) and Document Management Systems within Telenor. Telenor Geomatikk has 300 employees at 17 different geographic locations in Norway. The company’s services and products are GIS/GIT consultancy and knowledge management, scanning, vectorizing and thematic mapping, land surveying, GIS and CAD conversion, documentation of company internal networks/CAD, mapping, orthophotos and digital terrain models (DTM)

**Hans Jørgen Lied** was born in 1970 and became M.Sc. at the Stavanger Polytechnic in 1998. During 1997 to 1998 he worked at Fjellanger Widerøe AS in Trondheim. Here he mainly worked with software development and quality assurance. He finished his master thesis on “Process improvement by using Pareto analyses in a small company” in 1998. In 1999 he started working for Telenor Geomatikk AS in Trondheim. During this period he has mainly been working with software testing, quality assurance and software development.

**Tor Stålhane** was born in 1944 and became a M.Sc at the Norwegian Institute of Technology, University of Trondheim (NTNU) in 1969. During 1969 to 1985 he worked at SINTEF - UNIT, department for languages and compilers. From 1985 he worked on his Ph.D. studies and finished his thesis on software reliability in 1988. From 1988 he was back at SINTEF where he mainly worked with quality assurance and software safety and reliability. In 1997 he became professor in Computer Science at the Stavanger Polytechnic. During the latest decade he has been mainly been working with safety analyses of software intensive systems and measurement based process improvement.