2016 IEEE/ACM 38th IEEE International Conference on Software Engineering Companion

Assessing the Process of an Eastern European Software SME using Systemic Analysis, GQM, and Reliability Growth Models - A Case Study

Vladimir Ivanov
Innopolis University
1, Universitetskaya Street
420500 Innopolis, Russia
v.ivanov@innopolis.ru

Manuel Mazzara
Innopolis University
1, Universitetskaya Street
420500 Innopolis, Russia
m.mazzara@innopolis.ru

Witold Pedrycz
Department of Electrical and Computer Engineering
University of Alberta
Alberta, Canada
pedrycz@ee.ualberta.ca

Alberto Sillitti
Center for Applied Software Engineering
Genova, Italy
asillitti@gmail.com

Giancarlo Succi*
Innopolis University
1, Universitetskaya Street
420500 Innopolis, Russia
g.succi@innopolis.ru

ABSTRACT

This paper reports on the experience of the authors in quantitatively assessing the development process of an Eastern European software SME (Small or Medium Size Enterprise). The company produces a very successful workflow and documentation tool, features about 30 full time developers and has a customer base of about 40 major organizations. It has hired the authors as consultants to address quality and productivity issues raised by the upper management and customers. The adopted approach is based on systemic analysis, and starts with a comprehensive GQM session with the top managers of the company, to fully define the scope of work, and progresses analysing the documentation, interviewing the manager and the lead developers, and quantitatively analysing the issue tracking system in place. Specific attention is placed in identifying "schismogenesis", situations that may lead to unresolvable conflicts. The approach has been proven successful in providing a result in short forecasted timeframe, and systemic analysis has been effective in spotting the most critical situations present in the company. The result has been a set of prioritized recommendations, centered first in eliminating the schismogenetic situations and then ranging from adopting a more quantitative process control, to streamline the activities, to organize a line of product.

*Address for correspondence: Giancarlo Succi, Innopolis University, Universitetskaya Street, 1, Innopolis 420500, Republic of Tatarstan, Russian Federation. Email: g.succi@innopolis.ru

ICSE '16 Companion, May 14-22, 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2889160.2889250

CCS Concepts

•**Software and its engineering** → **Empirical software validation;** *Risk management; Software product lines;*

Keywords

Software Engineering, Systemic Analysis, GQM, Software Reliability Growth Models

1. INTRODUCTION

This paper reports an experience in running a software process assessment in a medium-sized Eastern European software development company that is successful and profitable, but that has never run any such assessment nor has formalized its development process. The name of the company and its internal details are not revealed for obvious confidentiality reasons. The experience started when the top executives of such company asked the authors of this paper to "have a look" at their operations. Their desire was to get a "quick-and-fast" assessment and to receive silver bullets for improvement. In their view the assessment should have lasted at most one day and the time to achieve results was planned to be about a week. After negotiation, the management of the company realized that running a fully comprehensive analysis was appropriate and useful, and such analysis would have lasted two weeks of workload, necessarily spread across two calendar months in order to accommodate logistic and calendar needs of all the parties involved. The authors have proposed to run an assessment based on systemic analysis aiming at representing the organization as a network of relations, and therefore studying the various perceptions of such relations among the key personnel.

The assessment has been guided by a GQM session with the top managers in order to fully define the scope of work, and was based on the review of the existing documentation, on individual interviews of the five senior managers (lasting each about 90 minutes). On the top of that more than 1000 "tickets" of the various tasks performed have been analyzed. Finally, a reporting session has been scheduled and the re-

sults have been presented to the top executives. During the assessment, specific attention has been placed in identifying "schismogenesis", situations that may lead to unresolvable conflicts. The report included the mentioned assessment of the overall process, the identification of the areas for potential improvements, and a timeframe to implement them.

This paper reports on the experience achieved, and it is structured as follows. Section 2 describes the problems and its practical importance. Section 3 briefly outlines the state of the art in this field. Section 4 summarises the approach, the techniques, and the methodology being employed. Section 5 outlines the results of the assessment, identifies its innovation, its efficacy and effectiveness. Section 6 presents the best practices that emerged. Section 7 discusses the lessons learnt, and Section 8 draws some conclusion and identifies the future research in the field.

2. DESCRIPTION OF THE PROBLEM

The problem faced is how to organize the process of a company which was born and grown up quite successfully, however without any specific discipline and which has already attempted and failed to "put some order" in its process, and it is now facing the hardness of growing further without a complete restructuring of its daily operations. An important aspect of this problem is that, typically, these kind of companies are run by their founders; the founders have built them typically by hard work and by overcoming all the technical problems by their strong wills, even lacking specific competence in software engineering. They have learnt the importance to deliver products on time and have achieve it usually working extremely long hours, so they have become quite impatient with the consultants and researchers they hire. It is therefore hard for them to understand what it takes to assess the process of a software company, and that the improvement of such process is a long and tortuous path, not a straightforward (even if costly) decision. They may be ready to pay, also a large amount, but they are reluctant to wait and even less available to undertake a step-wise improvement process. In essence, they want silver bullets, and we know silver bullets do not exist. In the case under consideration the company features about 30 full time developers and has a customer base of about 40 major organizations plus several minor. The founder has a Master's degree in Economics. The developers of the company have all an education equivalent to a Master's degree in a technological field and have from 2 to 15 years of experience in their discipline.

The company has been founded in the early 2000s around a workflow and documentation tool, and has grown from a single-product, single-customer organization to a full multi-products, multi-customers one. The original product is still the major source of revenues for the company and it has been customized to the needs of the different customers. The customization has occurred in a quite unstructured way, so now in practice there are several different code bases that are completely independent one-another, with a significant amount of code duplication and the problem that when errors appears from the original code, the fixes have to be manually propagated across all the code bases. About 5 years ago a software architect has been brought in to start systematizing the development process and the overall architecture of the product, but no significant results have been achieved because of the inertia of the overall development process and some internal opposition. The company has a set of documents describing its process, but such document does not reflect in any way the reality. It uses a scripting languages for development and keep all the issues in an issue tracking system. Testing is done by a separate department, and there is neither regression testing nor any kind of systematic automated testing. The development activities are around the different code bases, each one of which has a technical leader, and, sometimes, a business leader.

As mentioned, the founder, major owner, and CEO of the company asked to have a quick process assessment to identify how to improve the process, possibly in one shot. He was aware in principle of the need of time and of institutionalizing an overall software process improvement initiative. Still, he explicitly mentioned that he wanted a few silver bullet as a result of, possibly, a one day assessment, for which, indeed, he was ready to pay without questioning the amount. It took about 4 interactions spanning over two weeks to convince him that it was impossible, so he reluctantly accepted to have a more comprehensive analysis, still providing very strict time constraints (no more that 45 days) and questioning every time additional data was requested to perform the analysis. It is ironical to notice that just convincing him of the infeasibility of his request required more effort than what he originally foresaw for the entire assessment. We claim that the problem under consideration has a significant practical importance because the situation presented in this case it is quite common in success ventures, especially in Europe, where the vast majority of software companies are of this kind. To have an idea of the size of such sector, just in the EU, in 2013 (a year of crisis) there were 21.6 million SMEs in the non-financial business sector, employing 88.8 million people and generating €3,666 trillion in value added; even though a precise identification is not possible, it is fair to say that out of them more than 10% related to software, with a result of roughly more than 2 million SMEs employing more than 8 million people generating revenues for more than €366B [16]; and this is just the EU, so, altogether, worldwide, the numbers are impressive. It could be objected that the definition of SME is different between EU, US, Russia, Canada, etc., still the substance of the relevance and the size of the problem does not change. Most SMEs start like the one under consideration, and they grow then facing exactly the situation presented in this case. The authors had previous experience in similar situation, regrettably, never recorded in a scientific paper.

3. STATE OF THE ART

Assessing the process of a software company is not a new discipline. Assessors have inspected software companies since their inceptions, and since the mid '80s there have been systematic approaches to process assessment, such as the Capability Maturity Model Integration (CMMI) [18, 21], the Software Process Improvement and Capability dEtermination (SPICE) [11], ISO 9000:2000 [4] and TickITplus.

The ISO 9000 family of quality management systems standards helps organizations meet the needs of customers and other stakeholders. ISO 9000 includes the eight management principles upon which the family of standards is based. ISO 9001 deals with the requirements that organizations wishing to meet the standard must fulfill. TickITplus offers a flexible, multi-level approach to IT quality and certification assessment that can be applied at whatever level is deemed appropriate to the quality and process maturity of the or-

ganization, and is based on TickIT, a certification program that allows software development companies to improve software quality.

Despite many claims and many attempts to customize these approaches to small and mediums size companies like [1, 2, 3, 10, 12, 15, 19], these approaches are still perceived by SMEs as complicated approaches intended for large companies and may scare the management away due to uncertainty during the implementation of changes. Moreover, all these approaches tend to be focused on the process items themselves and not on identifying the root causes of the problems, while identifying such root causes could be the easiest way to find a path for improvement. Altogether, two requirements for an effective process assessment suitable for SMEs are emerging and are the subject our effort:

- to be applicable also in small contexts, not requiring a major investment;
- to go to the root causes of the problems, and rather than prescribing practices and disciplines that would be hard, if not impossible, to follow, to present how changing some of the dynamics inside the existing process of a company, such problems would almost automatically dissolve.

The approach adopted in this work is composed of three elements:

- systemic analysis, schismogenesis and systemic interviews;
- the Goal, Question, Metrics Approach;
- Software Reliability Growth Models.

Systemic analysis is a technique to analyze comprehensively complex issues evidencing the mutual and reciprocal perspectives, relationships, and influences of such issues [6]. This approach has been used to analyze complex organizations and to define for them significant improvement strategies [14], but it has never been significantly applied in software companies. Within systemic analysis, schismogenesis [22] is the process that occur when two or more individuals in search of a balanced relationship enact mechanisms in order to achieve such balancing actually making the relationship divergent with positive feedback loop dynamics. It is a phenomenon that can be simply represented by the example of a cat waving its tail to signal a dog its desire to stay alone, while the dog interprets this as a welcoming signal, therefore moving toward the cat and expecting warm greetings. The final result is of triggering even more waving with a clear escalation of the friction. Schismogenesis have been found to be a significant risk factor in flexible organizations [25] and a root of divergent behavior.

Systemic interviews are a proven mechanism to identify situations of schismogenesis, and the associated imbalances and misalignments in relationships for people, families, and also organizations [17]; they focus on understanding from the participants the different subjective perceptions of the relations existing in organizations, their motivations, and alternate possible practices, with the result of making evident such perceived relations, of overcoming potential conflict, and thus of making the overall organizations more harmonious. In systemic interviews, participants are asked to express their views of the organization and of their roles and ambitions with it. The interviews are recorded and then transcribed, and a model is thus extracted on the basis of the interview itself and of its recording, posing particular attention to the words used and to the body expressions.

The Goal, Question, Metrics Approach, in short GQM [5], has been initially conceived to define a set of software metrics, but then used also in business alignment activities [9], in which a session is conducted with the top management to brainstorm the business goals and then questions are associated to goals to specify unambiguously the goals, and finally metrics are linked to questions to quantify the, possibly multifaceted, achievement of the goals.

Software Reliability Growth Models (SRGM) [26] are mathematical models that are able to describe the discovery times of bugs, and, to a certain extent, to predict them. In practice, they are curves that fit "well" with empirical data describing when a bug is found. There are two major kinds of SRGMs, ones which are pure convex curves leading asymptotically to the unknown number of total bugs inside the system, other starting concave and then turning into convex, still leading asymptotically to the total number of bugs. Research has evidenced that they work adequately also with various kinds of service requests, and in presence of noisy, manually collected data [20].

4. APPROACH, TECHNIQUES, AND METHODOLOGIES EMPLOYED

To properly understand the approach, the techniques, and the methodologies employed, it is important to remind that the CEO desired to identify process improvement initiatives able to improve the status of its company. This would have led to significant grow in the years to come. At the basis of any improvement initiative there must be an assessment of the state of the art of the *real* processes currently in place. In synthesis, the proposed approach uses the systemic theory as the cornerstone to understand the process in place inside the company, to identify its limitations, and to determine appropriate proposals for its improvement and/or reengineering. However, at the basis of any software process improvement initiative there must be the definition of the overall goals of such initiative, if possible in the framework of more comprehensive, company-wide goals. To this end we use the GQM. The goal of the GQM is twofold. It is used to define the technical and business goals, the associate questions, and the metrics to evaluate the operations of the company and the process assessment itself. Moreover, the GQM is the reference support to understand the existing alignment of the business of the company with its current operations, which is then at the basis to identify possible situations of schismogenesis. Finally, since issue management systems are now very popular, we analyze the time of arriving and of closing issues with SRGMs, which, as mentioned, have also been able to handle all kinds of issues and to be robust to human error [20]. Moreover, typically these assessment are done in context where there is already some documentation (as in this case), but the documentation was not reflecting the reality. Furthermore, the management typically has some skepticism toward long initiatives that might be perceived as "academic" or "for larger and better organized companies", therefore, there is an intrinsic risk that in between the assessment the top management might step back from the commitment, so throughout the process the artifacts created

also responded to the need to be understandable and "justifiable". Practically, the approach consists in the following major steps:

- a comprehensive GQM session with the top management, where the business and technical goals of the company and of the assessment are defined, a sequence of questions are identified to determine if the company is achieving such goals, and finally a set of metric is developed together with mechanisms to compute such metrics to provide answers to the questions;
- a set of individual interviews conducted following the principles of the systemic organization theory, centered on the concept of schismogenesis, and duly recorded; during the interviews the developers are encouraged to speak freely of the company, its processes, its goals, and also about the personal ambitions, what they would improve in the company, and what they perceive as major issues and challenges;
- a detailed analysis of the recording of the interviews to extract relationships, metaphors, ambitions, and issues of the participants;
- a careful reading of the existing documentation of the process;
- an analysis of the arrivals of bugs and issues as recorded in the issue tracking system using SRGM, to identify specific working patterns inside the development organization and the typical quality level of deployed products, especially with reference to remaining bugs after delivery and the customers perception of such bugs;
- a reconstruction of the process in a format understandable by non-technical people, where the different viewpoints are reconciled whenever possible and the differences are made evident; differentiation is a key issue in systemic analysis, since it is the way to profile different people.

4.1 The GQM Approach

The comprehensive GQM helped to frame the initiative and to elicit in a clear and measurable way the business and technical goals of both the overall company and of the specific process improvement initiative. It also served as an important tool to generate trust in the senior management, who felt properly heard and understood.

4.2 The interviews and their analysis

The interviews provided a deep understanding of the process in place, of the position of each individual with respect to this process, and of ambitions and the frustrations of each interviewee. Analyzing them in deep and crossing them with the written documentation allowed to form a synoptic view of the different perceptions of the process in place, of its advantages and of its limitations. Every interview was then the basis for an individual reconstruction of the process based on the outcome of such interview using a tabular formalism, aiming at being "quite" unambiguous, but still understandable also by non-technical people, as in the following excerpt.

As it is evident, the table contains the description of each activity to perform, the people involved (marked with an "X"), and the artifact in use (market with a "W" if they are written or modified, and an "R" if they are only read, that is, used but not modified. Particular attention is posed on (a) detecting differences in the use of term between interviewees and among interviewees and written documents, and (b) uniforming the terminology in the tables. The differences in terms or in meaning assigned to terms are used to discover schismogenesis.

Figure 1: Number of active bugs in the issue tracking system

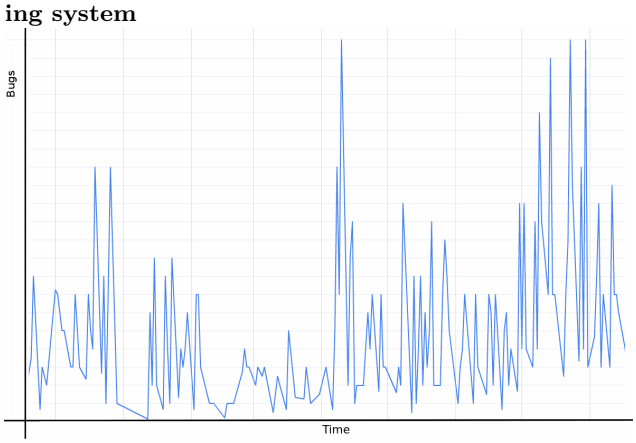
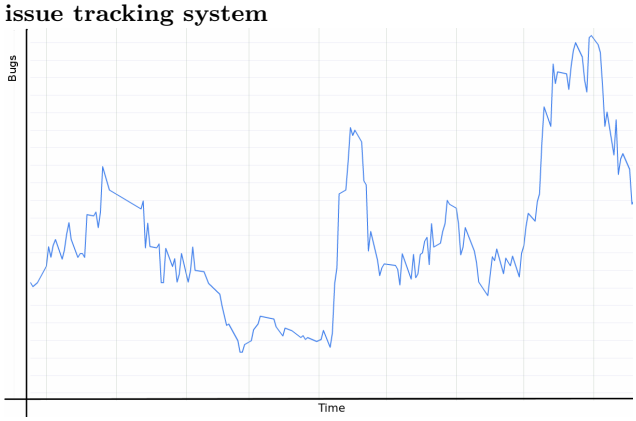


Figure 2: Smoothed number of active bugs in the issue tracking system



4.3 Arrivals of bugs and issues

After this stage, the service requests are analyzed to discover, through their analysis, one the actual releases were performed, the leftover bugs before deployment, the arrivals of bugs from the customer after deployment. The figure (Fig. 1) represents the number of active bugs in the issue tracking system. Details about exact number of bugs and timing are omitted for confidentiality. This curve is a bit noisy, but simply taking the moving average of it over a period of one week, cleans most of the spikes, and we obtain something like the following (Fig. 2).

The diagram itself is quite informative on the process of the company, since it evidences:

- It is possible to determine a few key moments for production:
  - when testing starts, when the curve starts increasing steadily, as a result of the process of start de-

Table 1: Activities, roles and artifacts

Activity	Customer	Senior Architect	Lead Analyst	Developer	Requirements	Specification	Release
The Senior Architect defines a list of requirements through a direct interaction with the customer and writes them in natural language.	X	X			W		
The Lead Analysts analyzes the written documentation from the Senior Architect and produces the specification in free text.			X		R	W	
Senior Architect, based on the current workload the urgency of the specification and its expected duration determine in which release to put it.		X				R	W

bugging;

  - the release time, typically when the curve has a local minimum, followed by a some increases;
  - the sequences of releases, through determining the distance between successive local minimums.

- The curves informs also of the number of left-over bugs before release and their discovery at the customer's site, providing a very accurate description of the overall quality of the production process.

- Analyzing the distributions of the different other collected information (for instance, severity, priority, etc.) it is possible to determine whether such variable are effectively relevant to describe the development process, and therefore it is possible to identify which variables are really in use.

Moreover, applying reliability growth models, it is possible to estimate additional parameters of the development process, such as the effectiveness in defect identification and removal, etc.

4.4 Reconstruction of the process

Based on the collected information, it is possible to start the reconstruction of the process. This takes the form of a synoptic diagram where the different tables can be synthesized as follows. In this table the process is streamlined and there is one column per developer or per written documentation (it is possible to have multiple and non-coherent description of the same process). For simplicity we limit the visualization to two people who has been selected only as exemplar. Whenever possible, joint activities should be represented by merged cells, or, when not possible, by cells with the same background color. Parts common across all phases are marked in green and missing activities in red. When, at a given step, all activities are different no background color is present. In addition to this table, there is a list of observation coming from this synoptic view, from the review of the "free statements" of the developers, and from the analysis of the data on bugs and issues.

5. RESULT OF THE ASSESSMENT, INNOVATION, EFFECTIVENESS AND EFFICIENCY OF THE PROPOSED SOLUTION

5.1 Result of the assessment

The first results of the assessment came from the GQM itself. The CEO of the company made clear to us and to the people reporting to him that he wanted a scale up of the business and to outfit competitors both in terms of quality of the product and of features to develop. Indeed, these are quite generic goals, but moving from business goals to technical goals, he linked such goals to the necessity of creating a single product line into which all the code bases should be merged and to establish sounder testing practices.

The process assessment had therefore such business and technical goals in mind. It emphasized different kinds of problems, the hardest of which were related to some evidenced presence of schismogenesis, while other were technically challenging, and other simple a matter of education. The bottom line is that it was more than evident, simply form the tables above, that the process of the company needed to be restructured and streamlined. The written documentation was not reflecting the reality and different teams around different code bases were behaving differently not only internally but also with respect to the end customer; this is indeed something very far from efficiency and a situation where it is clearly impossible to insert any product line. However, the key question that systemic analysis helped us to answer was "why?" Through it, we identified a couple of severe sources of schismogenesis and a few additional problems, and only fixing these problems it would have been possible to improve the process.

The first source of schismogenesis was in the position of a couple of Senior Architects. They were performing really well, and they were also among the first employees of the company. Therefore, the CEO promoted them to the management positions in order to lead the development process and covering leadership roles. However, they were not keen to cover such roles, and did not enjoy the interaction with customers, which was a major part of their job description.

Table 2: Written documentation versus interviews

Written Process	Interviewee 1	Interviewee 2
The Senior Architect defines a list of requirements through a direct interaction with the customer and writes them in natural language		
The Senior Architect presents the information to the Lead Analysts and some implementation scenarios; together they write the specifications of each task to perform in natural language.	The Lead Analyst analyzes the written documentation from the Senior Architect and produces the specifications of each task to perform in free text.	The Senior Architect writes the specifications of each task to perform and discusses them with the customer
The Lead Analysts transfer specification back to the Senior Architect for the effort estimation.		This activity has not been mentioned
The Senior Architect and the Lead Analysis analyze the current situation and determine in which release it is possible to include the task	Senior Architect, based on the current workload the urgency of the task and its expected duration determine in which release to put it.	The Senior Architect determine in which release to put each task to perform.
This activity has not been mentioned	This activity has not been mentioned	The assignment of the task to a specific release is confirmed by the customer
The Lead Analysts with the Senior Architect assigns the task to a Developer	The Senior Architect assigns the task to a Developer	
The Developer writes the code		
This activity has not been mentioned	The Developer tests the code whether it is done correctly	
The Senior Architect verifies the code done by the Developer		
This activity has not been mentioned	The Senior Architect devotes approximately 25% of his time to the architectural changes	This activity has not been mentioned

Giving their position of influence in the company, they were managing to bend the process according to their preferences. It has to be noted here the presence of schismogenesis: due to their excellent technical skills they were moved up, and because of this, they could avoid those tasks that their rise would have implied. This situation demanded a clarification of the mutual expectations of each of these Senior Architects and the CEO.

The second source of schismogenesis, evidenced clearly analysis the situation of the leftover bugs in the issue tracking system, was related to testing. Testing was not particularly liked by the developers of the company. So a group of testers was created and was apparently populated by the people who were not particularly appreciated inside the direct development group and had the lowest salaries. Indeed this was worsening testing and was requiring more testing and more problems, while the testers were further penalized with an obvious positive feedback loop that was making the situation worse. Also these demanded a top-level clarification, and possibly a revision of the role and of the prestige of testers inside the company. There are also other problems in the company spotted by the systemic analysis even if not generating the very dangerous positive feedback loop of schismogenesis. The company also had a separate analysis group outside the development groups. Given how it was created, the company had a culture for development, so the analysis group was also the place where those not so strong in programming were dispatched, quite like the testing group. Indeed, these people did not have the same understanding as the developers of the software being developed, so the overall analysis process was quite ineffective and some groups simply ignored the requirement of asking the analysts to write the specifications, doing them themselves. Also, given the profile of the analysts, it was not possible to ask them to write acceptance test for the specifications they were writing.

Moreover, it was found analysing the time of resolution of the issues that there was the contradicting desire of overspecifying issues and of then not using such information properly. Issues were classified on multiple level of priorities. Launching a Kruskal-Wallis test on the resolution time of the issues of different priorities, it was found that no statistically significance difference was presence. There was, though, a statistically significant Spearman's rank correlation between priority and resolution time, meaning the developers and the analysts were able to have higher priority tasks handled faster, even if apparently the specific numeric assignment of priority to task was not reflected in the resolution time. The consequence is that reducing the levels, say, to three would have helped to have an effective and statistical significant, grouping. Other minor problems were present, but too specific and not critical enough to be presented in this kind of forum, given the strict space constraints.

At last we mention that the systemic analysis evidenced that there was a lack of a clearly defined process, shared across all the development groups of the company. This was a major lack of the company and might look like the largest impediment for the company to achieve its technical and business goals. Notice that progressing toward a more defined process would not necessarily mean to have as a first step an identical process for all the development groups, still it would be essential to formalize such process

even if the different groups would have proceeded differently, otherwise, the risk is that any improvement would try to hit an unexistent goal. Despite being so evident and appearing a major issue in progressing toward higher level of efficiency and productivity, we did not consider this as the root cause of the problem nor as the first target to hit in moving toward the strategic goals of the company. Forcing a defined single process across the company would have been a failing initiative if first the company would have not targeted its schismogenetic situations: to the very best, it could have worked for a while, as long as the attention for the process would have been low, and then it would have disappeared. It would have been like trying to block oscillations at resonance frequency of a mechanical device - the positive feedback loop at the end prevails.

Altogether, our recommendation to the company have been to first understand these situations and to later progress to formalize the process taking care of its lack of homogeneity, as evidenced by the systemic analysis and in parallel to solve minor aspects, like the ones on priorities in the classification of issues. Only after all these moves would have been completed, a product line approach, a streamlining of the activities, and a more systematic testing process appeared to be desirable and pursuable. At the end of the work the company appreciated the analysis performed and, in a comprehensive closing session, claimed to be satisfied by it.

5.2 Innovation of the proposed solution

It could be argued that our approach does not carry a significant novelty. Overall, all the issues of schismogenesis could be superficially flagged as "changing the culture of a company". Indeed, the issue of schismogenesis and the overall systemic analysis is linked to the "culture" of the company, however, it cannot be reduced to it. With a metaphor, studying the "culture" of a company is quite like psychoanalysing it, since it requires to go deep to how it has been generated, how its myths have been established, etc. Systemic theory moves away from such approach; understanding the culture of a company and its history is indeed still very important, but the core is modeling the current situation in terms of relationships between entities, with particular emphasis on the difference of perceptions and of meanings [7]. Some studies have been already performed, for instance evidencing the difference between endogenous and exogenous process control [13].

Indeed schismogenesis occurs only in presence of exogenous process control. Still, this is the first work based on the systemic approach and acknowledging the role of schismogenesis. The field of process reconstruction is vast. There have been hundreds of papers on the subject. Here the approach is significantly simplified because the assumption is that once the conflicts with positive feedback loop have been identified and resolved, and once also the other major issues identified by the analysis have been fixed, then the process almost ready to be described and managed, without needing to cover all possible details, since it is anyway a knowledge intensive task, not prone for overspecification [23].

5.3 Effectiveness and efficiency of the proposed solution

In terms of effectiveness, the approach based on systemic analysis in this case appears to have unveiled the root causes of the problems present in the company, that were otherwise hidden, and has paved the way to a solid process improvement activity, that could lead to the achievement of the technical and business goals defined by the top management of the company. In terms of efficiency, the work done was intense, but in comparison with a full process assessment was quite light weighted. Overall we budgeted about 250 hours of work, of which, about 100 were for the transcription of the recorded interviews and were performed by an assistant. The rest involved attending the meetings (altogether about 20 hours), reviewing and building the models (about 80 hours), analyzing the bugs and their arrival times (about 30 hours), and writing the final report and recommendations (about 20 hours).

6. EMERGED BEST PRACTICES , TOOLS, AND SOFTWARE PROCESSES

In this experience it has emerged that systemic analysis is applicable to understand both the existing process of a software production organization and the most critical obstacles that it is facing to make its process more efficient, often generated by what is called schismogenesis. A few additional good practices have also been appeared. First, as proven also in other contexts, the GQM provides an effective guidance in interacting with the top management of a company, since it helps to define in a clear and unambiguous way quality measures and associated values for both the context of work (in our case, the overall business of the company) and the expected activities (in our case the assessment). Needless to say, several times we had to go back to the discussion we had during the GQM session to reassure the management that we were on track. Second, recording the interviews have been an effective mechanism for understanding the underlying process, posing specific attention to the words that were in use by the company. Third, companies, and this case is not an exception, are usually impressed in dealing with professionals that, even though offer services in process improvement, have also a strong technical, system level, background, and to receive, together with high-level advices, also some short-term tip, like use this tool, fix this other tool. It helps to ensure them that they are not in the hand of some extravagant and highly paid business consultant.

Moreover, being able to show the value of quantitative analysis, for instance demonstrating that plotting simply the arrival time of issues in the issue tracking system we can tell them the time of releases (the peaks in the arrival time of issues), is a major plus. This experience has not used of any specific tool, apart the suite of Google Drive to produce and share documents and graphs and R to compute the statistics [24]. No specific software process has been involved, unless we can consider a "software process" the overall approach to software assessment that has been previously discussed.

7. LESSONS LEARNT

7.1 Overall assessment of benefits, risks and mitigations

As already mentioned, we consider the experience a success. Among its benefits we list the ability to spot the hardest problems easily, the fact that after solving the hardest schismogenic problems, the process starts naturally to realign itself, the power of quantifying properly the process,

and the pivotal role of the GQM. As also already mentioned, these initiatives are always subject to the risk that the management drops (suddenly) its support. For this reason, the tabular representation of the process was selected and natural language was adopted even in the formalizing tasks we were involved.

7.2 Broader applicability

The use of systemic analysis to assess and improve processes and organization is already somehow diffused outside software development environments. In this paper we have evidence one use in a software development company and we see no obstacles to its wider diffusion. Moreover, in this paper we have evidenced two schismogenetic situations. They appear to be situations that may occur also in other circumstances, thus we think that it could be possible to build a catalogue of risky schismogenetic situations to guide manager, consultants, researchers, and software engineers in general, such catalogue could resemble something like a collection of process anti-patterns [8], but with the advantage of fully understanding the dangerous dynamics of such situations.

7.3 Other lessons learned

In this case, it appears evident - as previously mentioned in the work of Thomassen [22], - that the flexible environment typical of knowledge based organization, like software organizations, are clearly exposed to the risk of schismogenesis.

8. CONCLUSIONS

In this paper we have reported our experience in running a software process assessment in a medium-sized Eastern European software development company using an approach based on systemic analysis, the GQM and analyzing quantitatively the issue tracking system. The approach has been proven successful, and systemic analysis has been effective in spotting the most critical situations present in the company. The result has been a set of prioritized recommendations, centered first in eliminating the schismogenetic situations and then ranging from adopting a more quantitative process control, to streamline the activities, to organize a line of product. It would be interesting to expand the work with additional experimentation and to follow up on the long term the effects of the analysis. Only these two steps are able to determine the effective success of the approach beyond one case study, whose result may depend on many covariate factors. As mentioned, it would be also interesting to collect across the industry examples of schismogenesis, which could help software managers, engineers, consultants, and researchers in their work, gathering a better understanding of how software is being produced.

9. ACKNOWLEDGMENTS

The authors thank Innopolis University for providing funding for this research and Ekaterina Yamshanova for her support in some of the activities related to the collection of the data used in this research.

10. REFERENCES

[1] Aleksandar Aleksić, Miladin Stefanović, Slavko Arsovski, and Danijela Tadić. An assessment of organizational resilience potential in SMEs of the process industry, a fuzzy approach. *Journal of Loss Prevention in the Process Industries*, 26(6):1238–1245, 2013.

[2] MD Ashrafuzzaman, Mohammad Shahadat Hossain Chowdhury, MD Akhtaruzzaman, and Hasan Sarwar. An assessment of software development practices of SMEs in Bangladesh. In *International Conference on Advances in Computer Science and Electronics Engineering - CSEE 2014*, 2014.

[3] Yirsaw Ayalew and Kris Motlhala. An ISO/IEC 15504 based software process assessment in small software companies. *International Journal of Software Engineering & Its Applications*, 8(6), 2014.

[4] Robert Bamford, William J Deibler, et al. *ISO 9001: 2000 for software and systems providers: an engineering approach*. CRC Press, 2003.

[5] Victor R Basili, Gianluigi Caldiera, and H Dieter Rombach. Experience factory. *Encyclopedia of software engineering*, 1994.

[6] Gregory Bateson. *Steps to an ecology of mind: Collected essays in anthropology, psychiatry, evolution, and epistemology*. University of Chicago Press, 1972.

[7] Carlos Guillermo Bigliani, Rodolfo Moguillansky, and Carlos E Sluzki. *Shame and Humiliation: A Dialogue Between Psychoanalytic and Systemic Approaches*. Karnac Books, 2013.

[8] William J Brown. Raphael C. Malveau, III Hays W., McCormick, and Thomas J. Mowbray. AntiPatterns: refactoring software, architectures, and projects in crisis, 1998.

[9] Ernesto Damiani, Fabio Mulazzani, Barbara Russo, and Giancarlo Succi. SAF: Strategic alignment framework for monitoring organizations. In *Business Information Systems*, pages 213–226. Springer, 2008.

[10] Susanne Durst and Michael Leyer. How can SMEs assess the risk of organisational knowledge? 2014.

[11] Khaled El Emam, Walcelio Melo, and Jean-Normand Drouin. *SPICE: The theory and practice of software process improvement and capability determination*. IEEE Computer Society Press, 1997.

[12] Vassilis Gerogiannis, George Kakarontzas, Leonidas Anthopoulos, Stamatia Bibi, and Ioannis Stamelos. The SPRINT-SMEs approach for software process improvement in small-medium sized software development enterprises.

[13] Andrea Janes and Giancarlo Succi. *Lean Software Development in Action*. Springer, 2014.

[14] Charles B Keating, Paul Kauffmann, and David Dryer. A framework for systemic analysis of complex issues. *Journal of Management Development*, 20(9):772–784, 2001.

[15] Deepti Mishra and Alok Mishra. Software process improvement in SMEs: A comparative view. *Computer Science and Information Systems*, 6(1):111–140, 2009.

[16] Patrice Muller, Dimitri Gagliardi, Cecilia Caliandro, Nuray Unlu Bohn, and Demetrius Klitou. Annual report on European SMEs 2013/2014-a partial and fragile recovery. *Brussel: European Commission*, 2014.

[17] Mara Selvini Palazzoli. *The hidden games of organizations*. Oxford University Press, 1990.

[18] Mark C Paulk. Using the software CMM in small

organizations. Pacific Northwest Software Quality Conference, 1998.

[19] Alberto Simboli, Andrea Raggi, and Pietro Rosica. Life Cycle Assessment of Process Eco-Innovations in an SME Automotive Supply Network. *Sustainability*, 7(10):13761–13776, 2015.

[20] Giancarlo Succi, Witold Pedrycz, Milorad Stefanovic, and Barbara Russo. An investigation on the occurrence of service requests in commercial software applications. *Empirical Software Engineering*, 8(2):197–215, 2003.

[21] CMMI Product Team. Capability Maturity Model® Integration (CMMI SM), version 1.1. *CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1. 1)*, 2002.

[22] Bjørn Thomassen. Schismogenesis and schismogenetic processes: Gregory Bateson reconsidered. 2010.

[23] Ashutosh Tiwari, Chris J Turner, and Basim Majeed. A review of business process mining: state-of-the-art and future trends. *Business Process Management Journal*, 14(1):5–22, 2008.

[24] WN Venables, DM Smith, et al. An introduction to R: notes on R: a programming environment for data analysis and graphics, version 3.2.2. 2015.

[25] Henk W Volberda. *Building the flexible firm: How to remain competitive*. Oxford University Press, 1999.

[26] Alan Wood. Predicting software reliability. *Computer*, 29(11):69–77, 1996.