Software Process Improvement Plan for "Reflector"

Submitted for partial fulfilment of the requirements for "INF5180 Software Product and Process Improvement in Systems Development" – course

Department of Informatics
University of Oslo

EXECUTIVE SUMMARY

This report presents a software process improvement (SPI) plan regarding a business critical software system called “Reflector”. The relevant background information, a detailed improvement plan and a discussion of the approach, are provided.

The SPI plan is developed on behalf of “Critical Services Provider Ltd” (CSP), one of the national infrastructure providers in Norway. The improvement plan is product-driven, focusing on *Reflector*, which is implemented by CSP’s sub-contractor “Software Factory Ltd” (SF) and operated by CSP’s IT department, “Centralized IT Services” (CIS). *Reflector* is a data intensive and well established customer relation management (CRM) system in use by CSP and its customers. It has been in use by CSP for three years. The SPI plan relies mainly on PROFES methodology, established standards (for process and product quality) and tools.

The SPI programme will be carried out by an SPI working group, in tight collaboration with SF and SCP at both managerial and operational levels.

The objective of this SPI programme is to address the currently prevailing issues related to change management of *Reflector*. The main quality attributes the SPI focuses on are *Reflector*’s maintainability and lead time.

The improvement goals set early in the SPI plan are systematically focused on and thoroughly planned, with respect to preparation, execution and evaluation of the improvement measures. The SPI programme will take eight months in total.

Desired *Reflector* quality in addition to team building, better knowledge management and improved understanding of the related processes, products and roles are among the expected achievements.

This report includes context information (regarding the *Reflector* software system, the parties involved and the related processes), an SPI plan based on PROFES methodology and finally a rationale and discussion of the SPI plan proposed. The focus has been directed to the parts of the SPI which are most critical for this particular case, with respect to ensuring the validity of the results.

GLOSSARY

CSP: Critical Services Provider Ltd

CIS: Centralized IT services (CSP’s IT department)

SF: Software Factory Ltd

SPI: Software Process Improvement

GQM: Goal Question Metric

PPD: Product-Process Dependency

KPI: Key Performance Indicator

KLOC: A measure of the size of a computer program. The size is determined by measuring the number of lines of source code a program has.

Reflector: A software application developed by SF, operated by CIS and used by CSP employees and customers

Defect: An imperfection that causes inadequacy or failure; a shortcoming [17]

Failure: Nonperformance of what is requested or expected; omission [17]

Lead time: The time interval between the initiation and the completion of a production process [17]

TABLE OF CONTENTS

1. Introduction

This report serves as a software process improvement (SPI) plan. It is developed on behalf of "Critical Services Provider Ltd" (CSP), one of the national infrastructure providers in Norway. The improvement plan is product-driven, focusing on a crucial software product called "*Reflector*", provided by CSP's sub-contractor "Software Factory Ltd" (SF) and operated by CSP. The process improvement plan relies mostly on PROFES [1] methodology, although only a fraction of the methodology is exploited in form of a tailored version of PROFES.

The objective of this software process improvement initiative is to address the currently prevailing issues related to change management of *Reflector* – a business critical software product whose development is outsourced by CSP to SF. *Reflector* is a data intensive and well established customer relation management (CRM) system in use by CSP and its customers. It has been in use by CSP for three years.

The current modification approaches on *Reflector* are error prone and time consuming. The time between new releases is considered to be too long, while number of defects due to the modifications, is escalating.

The demands on *Reflector*'s maintainability and adaptability are however constantly increasing, as CSP is facing changing needs in form of new workflow patterns, new services, new specialized systems based on different technologies that need to be integrated with *Reflector*, new regulations, new standards etc. It is crucial that *Reflector* can timely comply with the changes. As a result, management of CSP is concerned that the currently unsatisfactory quality of *Reflector* will become a hinder to meeting the new needs, and as such might degrade the company's ability to provide optimal services.

Due to its being established and adopted by CSP, and tightly integrated with the CSP's overall systems, *Reflector* is not a candidate for exchange with another similar product.

Instead, a group of professionals has been asked to assist CSP in developing an SPI plan which would address the central software process factors having an impact on those system quality attributes which have been identified to be crucial, namely maintainability and lead time. Key personnel from both CSP and SF are involved. While both CSP's and SF's objective is improvement of system quality, SF also expects achieving general efficiency increase in own organization, by deploying this SPI plan. There is a broad agreement within CSP that the quality of *Reflector* is unsatisfactory, and that the problem originates from process related issues. CSP management has openly expressed concern about the system's representing a possible future obstacle for optimally running and adapting an otherwise flexible and evolving organization. This has resulted in an organization-wide commitment within both CSP and SF, for addressing the issue in the nearest future possible. CSP is SF's far most important customer, and any threats to the cooperation are undesirable.

This report proposes a software process improvement plan for CSP's challenges regarding *Reflector*. It provides the relevant background information, a detailed improvement plan and finally a discussion of the approach.

The report is organized as follows. Section 2 presents the current situation, including the process, the parties involved, the quality properties and the application areas of *Reflector*. Thereafter we focus on the major issues and select the priorities for the improvement plan. Section 3 concentrates on the improvement plan itself and proposes how to undergo planning, execution, analysis and packaging phases of the PROFES method, in our context. Section 4 substantiates and discusses the choices made throughout the phases from Section 3, before concluding and summarizing possible initiatives for further work in Section 5.

2. The current situation

This Section outlines the current situation regarding processes within and among the parties involved, properties of the system in focus, its use areas, processes for its specification, development, maintenance and operation, its quality properties, evolution prospects and its interactions with overall systems and end users. The first sub-section characterizes the context and the scope, the second sub-section outlines the present issues, while the third one specifies the focus areas by defining the main priorities of this SPI plan.

2.1. Scope and context characterization

The subject of analysis is *Reflector* application and the processes related to its quality attributes of concern. This sub-section outlines the background, i.e. the parties involved, their relationship, the products and the processes in use. It provides the wider overview, before defining the focus more specifically, in the successive sub-sections.

2.1.1. The parties involved

Figure 1 shows the organizational setup among the parties involved. Critical services provider (CSP) is the main stakeholder and the initiator of this SPI. CSP is one of the national providers of the critical services infrastructure and the associated services. They operate:

- 30% of the national electrical power network
- 25% of the entire tele/internet network in Norway
- The software services provided to their employees and customers of the network services. (CRM for billing, ordering, service portfolio configuration, operational info, user management etc.)

Reflector is but one of their specialized software applications. *Reflector* is a CRM with end-user customized service portfolio. It contains functionality for:

- Ordering
- Billing
- Operational info to end-users (capacity, errors, usage…)
- Authentication and authorization

CSP has 40 regional offices (RO1 - RO40 on Figure 1) across the country. ROs are responsible for their respective geographical areas, and manage area-specific:

- Infrastructure development/maintenance
- Customer queries
- Local administration.

Service portfolio selection within *Reflector* may be area or customer group specific. Each RO is responsible for 800-1300 customers (C1, Cx… on Figure 1), depending on the size of the local community.

Centralized IT Services (CIS) is CSP's IT department. It is co-located and well integrated with top management of CSP. CIS has 40 employees – 90% with university degree or more as well as four or more years of relevant experience. The department enjoys international reputation for their exemplary achievements and publications on integration and maintenance of dependable software services aimed at supporting society critical infrastructures. CIS is responsible for providing the IT services to headquarters, regional offices and the end customers. CIS manages:

- operational environment
- sub-contracting of the outsourced software development services
- operation and maintenance of middleware for integration and monitoring
- data warehousing
- security policies and documentation maintenance
- user training.

CIS is organized through projects, so that each employee may be affiliated to several projects, i.e. a matrix organization. The development processes are based on SCRUM [2] [7] methodology. Some of the specialized application development services are beyond the core business area, and therefore outsourced to external partners. CIS relates to a number of sub-contractors, one of which is Software Factory (SF).

SF is a software development company specialized on a major CRM application, called *Reflector*. SF consists of 25 employees, all of which are heavily involved in development, while five team members have all managerial responsibilities in addition to architecture planning and some implementation. The process methodology applied within SF has never been formalized, but resembles mostly to a waterfall model. The SF team members have a culture of tight and informal collaboration. SF delivers a customized version of *Reflector* to CSP, via CIS.

Reflector comes with monthly releases of latest versions, after implementation of customer change requests or necessary upgrades. *Reflector* is developed on .NET platform, and each release is tested in-house by SF, prior to its release to CIS. *Reflector* is integrated with CIS's overall systems via their middleware platform. The integration is done by CIS and covers message exchange, authentication, authorization, provisioning, archiving and document exchange, accounting, billing etc. Thus, in relation to each release, integration tests are undertaken by CIS in order to verify *Reflector*'s interoperability with the integrated specialized software and middleware, all operated by CIS. Each integration test takes 2-4 days in average. Approximately two times a year, the integration testing takes almost two weeks, before CIS can confirm that the upgraded *Reflector* is compatible with the overall systems it is integrated to, before it can be deployed. The test results have, however, in approx. 80 % of the new releases, been misleading and failures (due to previously unknown defects in several indirectly related modules of *Reflector*) have occurred after deployment.

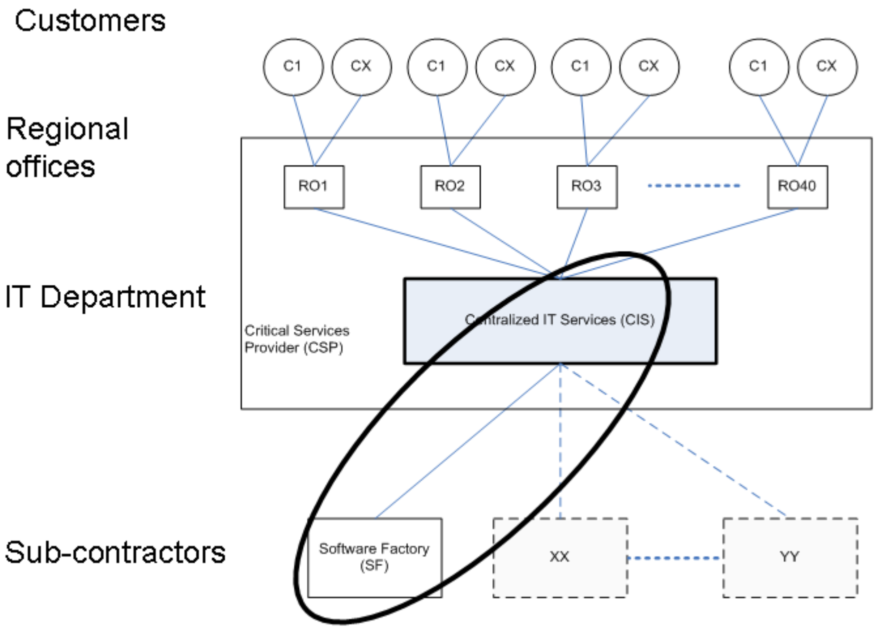

Figure 1: The organizational setup

2.1.2. Processes and products

Reflector has both front-end (www-based) and back-end (db, internet information server etc.) modules. Different functionalities are used by different user groups (employee groups, end user groups etc.). The authentication is managed by a separate system integrated with *Reflector*, user provisioning is handled partially by *Reflector* and partially by middleware, while authorization is managed by *Reflector*'s own functionality. *Reflector* is a licensed software, giving CIS limited insight into source code and thus limited possibility for undertaking own maintenance activities.

The relationship between CIS and SF is based on a contract, which contains regulations on pricing, delivery deadlines, as well as functional and technical properties of *Reflector*. The contract is informal and descriptive, and contains no demands on the development process at SF. CIS does not currently enforce use of any particular development methodology at SF.

Reflector is integrated with CSP's overall systems and operated by CIS. Thus, all the end users (customers and CSP employees) relate to CIS only. Similarly, SF relates to CIS only. Each new release of *Reflector* is delivered to CIS, which then integration tests and deploys it.

Reflector has been in use by CSP and its customers for three years. The current contract is valid for four years, with a renewal option for additional four years.

Top, middle and project management of CSP are all committed to applying the SPI in order to achieve a satisfactory level of quality of *Reflector*. The SF team shares this objective and is fully committed in achieving the common goal, namely achieving an acceptable level of *Reflector*'s quality with respect to maintainability and lead time. In addition, both the management and the development team at SF have expressed their full interest and commitment in trying out the improvement measures towards becoming more efficient in general. Although *Reflector* (in its respective customized versions for the different contractors) is delivered by SF to numerous customers across the country, CSP is SF's far biggest and most appreciated customer. Income from CSP represents nearly 40% of SF's turnover. The initial costs SF had in order to satisfy CSP's initial requirements and win the tender, represented two years turnover. CSP is therefore unquestionably SF's most demanding and most important customer, at the same time.

Both CSP and SF management have explicitly assured the total resources anticipated necessary for carrying out the improvement activities. CSP and SF want to make sure that *Reflector* is capable of meeting CSP's new working processes, regulations, integration needs etc. Based on this, the following quality attributes are, unanimously by both SF and CSP, identified to be most relevant for *Reflector*: maintainability and lead time.

Maintainability is defined as "the capability of software to be modified". Modifications may include corrections, improvements or adaptation of the software to environmental changes, and in requirements and functional specifications. Maintainability can be further categorized into analyzability, changeability, testability and stability. [1]

2.2. Identified issues

Reflector is a business critical system of unsatisfactory quality. Each release contains a number or modifications in order to comply to the necessary changes in regulations, working processes or new requirements. The result of each release are, however, new and unforeseen defects due to various modifications. The modifications are inevitable in order to adapt to the new requirements. *Reflector* has so far shown to be incapable of adequately meeting CSP's new business needs (new services, regulations, technologies, additional integrations etc.). *Reflector* is gradually becoming more critical and complex, a development which is expected to escalate. Change management is however already unsatisfactory and will most probably degrade further, given the development trend expected. It is therefore crucial to improve the processes related to contracting, development, maintenance, quality assurance and operation of *Reflector*, so that the system achieves the quality level necessary in the long term.

Maintenance of *Reflector* is currently handled through monthly upgrades. SF develops, tests, and delivers a new release to CIS, who then integration tests, deploys and operates the new release.

The unpredicted defects, originated by the modifications in the new releases may take any time to uncover (even when deployed) and repair. The reparations can in turn cause new failures, which again propagate almost anywhere. This scenario occurs so frequently, that CSP management considers the system's low dependability as a threat to trust from customers and the company's ability of operating efficiently.

Currently, *Reflector*'s availability is 95% of the total time, and the number of defects reported after deployment of each release is between 10 and 15. The causes of the defects originate both from *Reflector* itself, or its integrations with CIS' overall systems (due to dependencies across the systems). The distribution of the two cause categories is approximately equal. Furthermore, the one month lead time consists of approx. three weeks of SF's development of the new release, and almost a week for CIS' integration implementation, integration testing and deployment. The dependencies within *Reflector* are many and strong. There is a significant number of circular dependencies. Modularity is extremely low, as the system is designed and implemented as a whole. In addition, the integrations of *Reflector* with the overall applications operated by CIS are tight and proprietary in terms of both interface technologies and semantical properties of the data exchanged. No standards are applied, although numerous standards for communication, formatting and orchestration for the work processes in question, exist. Hence, the system's high complexity, low modularity, high coupling and low cohesion are assumed to be the main causes for unsuccessful change management with many related defects and long lead time.

SIC is more specifically concerned about:

- Unstructured system design which prevents from adaptability in the long run.
- Escalating security, availability and scalability issues, low modularity, high coupling and low cohesion within *Reflector*.
- Exponential fault propagation due to changes.

These issues are almost certain to increase in their intensity in the future, due to the system's becoming more complex and critical.

CIS does not know the particulars of the processes used within SF, but notices that very little formalism exists, as the SF team members communicate tightly and informally, claiming that they rely on waterfall-like development methods.

CIS believes that intrusion into SF's processes (by imposing requirements on them) is necessary in order to control development process of *Reflector* and thus make the product's quality more predictable.

2.3. Main concerns of the improvement plan

CIS has an opportunity to address some of the issues in relation to renewal of their contract with SF. The current contract can be re-negotiated and become operative in a year. The new requirements may address:

- Processes within SF, particularly adoption of a specific development methodology
- Conformance to standards
- Quality assurance demands, particularly testing routines prior to each version's release.
- Mutually agreed quality metrics and their acceptable values regarding system's modularity, cohesion, coupling and complexity and availability measures

- Incremental/modular upgrades.
- Undertaking of a security analysis prior to major releases.
- Improved documentation regarding system design (including structure, dependencies, dataflow, integration interfaces, setup of operational environment, configuration) and user manuals.

Possible overall measures include CIS' own processes regarding requirement specifications, integration testing and operation (including monitoring). More specifically, CIS may introduce:

- A dedicated integration testing environment
- Some metrics as KPIs in the monitoring environment.
- More precise and formal requirement specifications.

Thereafter, the effects of the measures enforced can be measured on the successive releases of *Reflector*.

This case is considered from CIS's perspective. This SPI is the first one of the kind, for both CIS and SF. The bold ellipse on Figure 1 indicates the scope of this analysis.

3. Improvement plan

This SPI plan is a tailored version of PROFES, for this particular context. It focuses on maintainability measures, as means to improved change management of a system whose changes cause substantial defects, increase lead-time and generally degrade quality.

The initial step is provision of an organizational infrastructure necessary for developing and carrying out this SPI plan. The organizational infrastructure consists of dedicated human resources, competence, organizational support, reporting mechanisms, support for process assessment and measurement programmes concerning both SF and CIS/CSP. In addition, the following tools for support of different PROFES activities, have been made available:

- Bootsample – for software process assessment
- GQMAspect – for building GQM plans
- MetriFlame – for measurement plans and data analysis

The external SPI (PROFES) experts are familiar with all the three tools, while one SPI project group member from SF and another from CIS have considerable experience with use of MetriFlame. No overall people involved know how to use the three tools mentioned above.

The commitment of top and middle management level as well as of project members of both CIS/CSP and SF is verified. PROFES method and its possible gains have, by the external PROFES experts, been presented through two workshops simultaneously held for CSP and SF management and overall SPI participants (interested or likely to be involved in the SPI). Application of PROFES in practice was explained and expected benefits were outlined. The second workshop was also the official kick-off of the SPI pilot project whose primary objective is development of an SPI plan.

The SPI working group founded shortly before the kick-off consists of eight members: (PROFES team roles are given in the parentheses below)

- Two external PROFES specialists (having PFOFES facilitator and lead assessor role, respectively)
- The managing director of CIS (also GQM expert)
- *Reflector* project manager from CIS (measurement facilitator)
- Quality assurance and integration manager from CIS (also assessor)
- Technical manager and board member of SF (also assessor)
- One project manager from SF (also GQM expert)
- One senior developer from SF (also measurement facilitator)

The group enjoys full support of their respective top management and boards, and commands over already budgeted resources according to the stipulates of the external PROFES specialists (now SPI working group members). As the initial written input, the group has received *Reflector* requirements specifications, specification of operational environment, user manuals for *Reflector*, the contract between SF and CSP and a risk analysis of *Reflector*, recently conducted by CIS.

The overall plan schedule, which the working group has obtained a mandate to carry out, is shown by Figure 2. The SPI programme will take eight months in total and is organized in 10 stages.

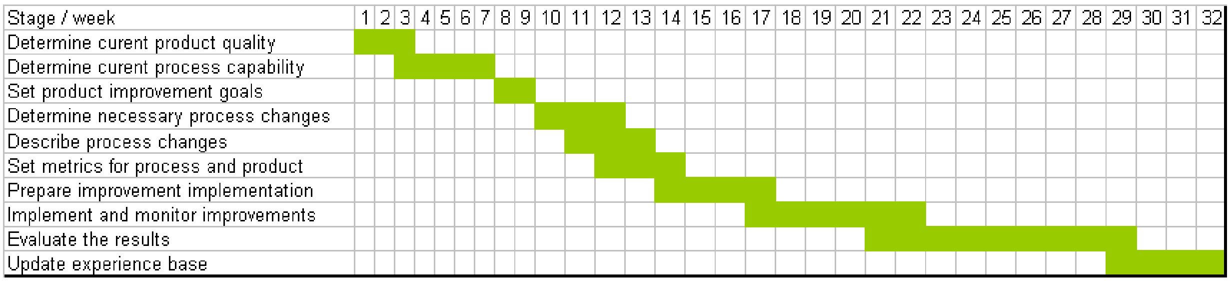

Figure 2: An overall plan schedule for improvement activities

The information dissemination and feedback is a continuous activity which will mainly involve two regular monthly workshops (similar to the ones presented above), and two regular monthly updates of a webpage with news on the SPI work, during the entire SPI. The SPI proposes a two-way information dissemination between all the SPI programme participants, during the entire SPI programme. Feedback sessions, web pages, workshops and meetings are among the methods which will be actively applied. Considerable written documentation will be exchanged for contributions, feedback and reporting purpose. The effort reporting (based on the PROFES templates) will take place regularly.

Table 1 below provides a plan overview for the 10 stages from Figure 2. Each stage is summarized with respect to the main activities, responsible individuals, the participants involved, the tools needed and the main outcomes. Each stage will have only one responsible individual – a member of the SPI working group who will coordinate the activities. In most of the stages this role will be assigned to one of the two PROFES experts from the SPI group.

Stage	Main activities	Responsible	Involved	Tools needed	Main outcomes
Determine the current product quality	• Interview developers and users • Retrieve historical data	*Reflector* project manager from CIS	• three CIS developers • three admin. interface users (CIS) • three end users • *Reflector* project manager (CIS) • quality assurance and integration manager(CIS)	MetriFlame	A presentation of current quality of *Reflector*
Determine the current process capability	• Develop a reference specification • Interviews with key personnel • Review available documentation • Document processes • Quantify the reference specification	A PROFES expert	• The SPI working group • Key personnel from CIS and SF	ABC-Flowcharter Bootsample	A process capability report
Set product improvement goals	• Interviews • Discussions • Evaluate external metrics • Rank product improvement goals • Tag workflow process models	A PROFES expert	• company management • project management • developers • end users • SPI working group	CIS' monitoring environment	Summary of the success criteria for this SPI Improvement goals
Determine necessary process changes	• Retrieve relevant PPDs • A feasibility study to identify the candidate processes which can be modified • Classify the processes	A PROFES expert	• SPI working group • experts and project managers from SF and CIS	PPD repository	A classification of candidate processes for improvement
Describe process changes	• Describe the processes selected for improvement • Update process models • Develop a reference process model • Interview CIS process engineers • Information dissemination • Training	A PROFES expert	• SPI working group • project personnel from both SF and CIS • CIS process engineers	ABC Flowcharter	Description of process changes and a reference process model
Set metrics for the process and product	• Set formal measurement goals • Structured interviews • Assemble a GQM plan	A PROFES expert	• SPI working group • Project team members at SF and CIS	• PROFES templates for measurement	GQM plan Measurement

Stage	Main activities	Responsible	Involved	Tools needed	Main outcomes
improvements	•Develop a measurement plan •Review the GQM plan and measurement plan •Reach an agreement w.r.t. GQM plan			goals •Abstraction sheets •GQMaspect	plan
Prepare improvement implementation	• Arrange progress meetings • Detailed planning of SPI steps • A kick-off meeting • Install an online help desk • Form pilot groups	A PROFES expert	• SPI working group •Top and project management at SF and CIS (a committee) • SF and CSP employees	•Abstraction sheets •PROFES improvement plan template	All conditions for SPI plan implementatio n fulfilled
Implement and monitor improvements	• Activate pilot groups at SF and CIS • Implement software changes • Collect measurement data • Prepare data for feedback sessions • Hold feedback sessions • Quantify the reference specification • Process assessment	A PROFES expert	• All involved SF and CIS personnel • SPI working group • Project quality managers from SF and CIS	•Monitoring env. at CIS •Databases •MetriFlame •MS Excel •PROFES feedback templ. •Bootsample	Improvement implementatio n and monitoring Feedbacks
Evaluate the results	• Analyze the data collected at previous stage • Evaluate product improvement goals and GQM plan • Evaluate the reference specification values • Evaluate process changes • Evaluate PPDs • Post-mortem analysis • Document evaluation results	A PROFES expert	• SPI working group • Quality engineers at SF and CIS	•MetriFlame •MS Excel	SPI implementati on results evaluated
Update experience base	•Build the infrastructure for the experience base •Document all relevant results, methods used, approaches, findings •Assure quality of the information •Disseminate the packaged info.	*Reflector* project manager from CIS	•Technical managers of SF and CIS •SPI programme participants	A proprietary multimedia database on intranet (shared by SF and CIS)	The lessons learned documented in experience base

Table 1: Plan overview

The version management of this SPI plan document will be supported by Subversion [8], which all the SPI working group members have gotten access to. Each one of the stages of Table 1 is dedicated a sub-section below.

3.1. Determine current product quality

The first task is determination of the current product quality of *Reflector*. To this end, representative members of three stakeholder groups will be interviewed:

- three system developers (involved in integration, testing and operation of *Reflector*) form CIS
- three end users of *Reflector* administrative interface from CSP
- three CSP customers (end users of the www client of *Reflector*)

about their experiences with the *Reflector*, with respect to the change management. Prior to this, a questionnaire will be designed with clear and consistent questions about kinds and number of undesired experiences regarding testing, integration and use of new releases of *Reflector*. This questionnaire will then be used during the interviews. The results will be analyzed and summarized.

In addition, some historical data will be retrieved from the monitoring environment of CIS. The data needed are: downtime, access denials, overloads, time between each incident and latest upgrade, service request frequency and service delivery frequency. Equipment for acquiring the product quality data will be assembled in form of a separate server with a data warehouse which will collect a copy of all monitoring data. This will be performed by the *Reflector* project manager from CIS. The metrics for data analysis will entirely be based on the quantitative definitions of maintainability sub-characteristics provided by ISO9126 [5]. The findings will be analyzed using MetriFlame tool and compared with the results of the interviews.

Based on the interviews and data analysis, the current quality values of *Reflector*, will be determined and a summary made. MetriFlame tool will be used for presenting the results. Data analysis and presentation of the analysis results will be performed by *Reflector* project manager from CIS and quality assurance and integration manager from CIS, over a period of three weeks, as shown by Figure 2. The results will be presented on open workshop, and on intranet of both SF and CSP.

3.2. Determine the current process capability

Next, the current process capability regarding the processes within CIS, SF and between them, will be determined. A reference (a high-level, comprehensible set of maintenance issues, with intuitive and informal terminology, appropriate for quickly being reused at a later stage of the SPI programme or later) specification will identify five of the most representative (with respect to scope, importance and frequency both previously and in long tern) maintenance tasks. The tasks will involve both SF and CIS ay operational level. Then, for each task, the SPI working group will seek to quantify:

- The total time passed before the task has been completely solved, without manifestation of any direct or indirect failures (due to the system changes performed in relation to this task) being reported for one week
- Number of defects caused by the changes related to the task solving
- Number of failures caused by the changes related to the task solving
- Number of modules affected by the failures caused by the changes related to the task solving
- Total downtime of the system due to the above mentioned failures. Include:
    - Lack of service continuity (e.g. interruptions, unstable system)
    - Reduced quality of service in any module
- Effort (number of work hours) in solving the task without any successive failures (caused by the changes related to the task solving) being reported in one week

To quantify the reference specification, the SPI working group will:

- perform interviews of key personnel at CIS and SF
- evaluate available documents
- observe some of the processes
- analyze logs

In addition, they will, based on own knowledge and experience, contribute with process models and written statements of the processes within and across the two organizations. Finally, the evaluation will be summarized in a process capability report structured at three sections:

- Process capability within CIS
- Process capability within SF
- Process capability between CIS and SF

Only the processes related to *Reflector* will be analyzed. The main focus will be put on the processes within SF. The approach will be founded on the BOOTSTRAP [9] methodology.

The report will contain both organization level and project level assessments, combining textual descriptions and workflow models. The workflow models will be developed in ABC-Flowcharter tool, while Bootsample will be used for the process assessment.

3.3. Set product improvement goals

The SPI working group will conduct separate interviews and discussions with company management, project management and developers at their respective organizations (SF and CSP), as well as with the end users. The interviews and discussions will seek view points on the current and desired system quality (both positive and negative aspects), market scenarios, business expectations, future development trends (functional and technical), previous current and future improvement initiatives etc. The group members will thereafter analyze and discuss the outcomes of the interviews within the group. The result will be a statement containing a summary of the success criteria for this SPI.

The available preliminary input, however, already provides considerable relevant information. Based on it, the following success criteria are currently among the possible candidates which will be discussed within the SPI working group:

- *Reflector* is available 99% of the total time
- Each available version of *Reflector* is defect free after its deployment
- *Reflector* release cycles are reduced from one month to one week

- The weekly releases of *Reflector* are capable of meeting all the currently relevant requirements, so that no modification need is pending for more than ten days
- All modifications are specified and implemented independently from one other.
- *Reflector*'s level of modularity is well defined, measurable and kept constant.

Estimated values of the current and desired level of maintainability, according to its ISO 9126 maintainability metrics definitions, will be assigned, by the working group.

ISO9126 (part 2) defines external metrics of all sub-characteristics regarding each dimension. In case of maintainability dimension, the standard defines each metric both qualitatively and quantitatively, in terms of:

- metric name
- purpose of metric
- method of application
- measurement, formula and data element computation
- interpretation of measured value
- scale type
- measure type
- input to measurement
- ISO/IEC 12207 SLCP reference
- Perspective beneficer

Table 2 shows an example of *Reflector*'s assumed current and target (desired) values of each external metric regarding sub-characteristics of maintainability, according to ISO 9126 (part 2) definitions. The values will be verified by analysis of the historical data retrieved from the monitoring environment. A measurement scale will be chosen for each product quality characteristic. Those metrics which are not measurable (approx. 20 per cent of metrics from Table 2), will be estimated based on expert judgment made by SF and CIS employees. The target values will thereafter be deduced by a brainstorming and discussions within the SPI working group. The definitions of the values and the units of measure for the values provided in Table 2, are given by ISO 9126. The dimensions themselves (analyzability, changeability…) are not quantifiable, but all their sub-characteristics are (see Table 2).

For example, stability metrics is one kind of external maintainability metric. One of the stability metrics is "Localisation of modification (Emerging failure after change)" – see Table 2. Its ISO 9126 definition is:

- metric name: Localisation of modification (Emerging failure after change)
- purpose of metric: Can user operate software system without failures after maintenance? Can maintainer easily mitigate failures caused by maintenance side effects?
- method of application: count failure occurrences after change, which are mutually chaining and affected by change?
- measurement, formula and data element computation: chaining failure emerging per resolved failure. X= A/N; A: number of failures emerged after failure is resolved by change during specified period; N: number of resolved failures. Note: it is recommended to give precise measure by checking whether cause of current failure is attributed to change for previous failure resolution, as possible.
- interpretation of measured value: 0<X. The smaller and the closer to 0 is the better.
- scale type: X=Abs.
- measure type: A: count; X: count; N: count.
- input to measurement: problem resolution report. Operation report.
- ISO/IEC 12207 SLCP reference: 5.3. qualification testing; 5.4. operation; 5.5. maintenance.
- Perspective beneficer: developer, maintainer, operator.

Based on the current and target values of the metrics, feasibility and expected payoffs of their improvements, a brainstorming within the SPI working group will be conducted and improvement priorities set. An example is shown by the third column of Table 2. Only the top ten priority sub-characteristics are denoted in the rightmost column of Table 2. Commitment of the overall management for the priorities will then be ensured. Moreover, percentages of the discrepancies between the current and the desired quality values will be deduced (not included in Table 2). The product improvement goals (as defined by ISO 9126) for *Reflector* will, by the SPI working group, at this point be ranked, based on the metrics improvement priorities set above. An example/possible ranking for *Reflector* is:

1. Maintainability (according to the priorities from Table 2)

a. Localisation of modification
b. Time spent to implement the change for user's satisfaction
c. Change recordability
d. Readiness of change
e. Time spent to implement the change by the maintainer
f. Less encountering failures after change
g. Effortless testing
h. Readiness of built-in test function
i. Failure analysis time
j. Satisfaction coverage of compliance items relating to maintainability

2. Lead time reduction (which is considered to be indirectly affected by the maintainability related improvement measures).

Analyzability	**Current (average)**	**Target (average)**	**Priority**
Diagnostic function support	0,6	0,9	
Data recording during operation	0,2	0,8	
Failure analysis time	0,8	0,1	9
Finding results of failure case	0,5	0,9	
Status monitoring during operation	0,6	0,9	
Changeability			
Change recordability	0,6	0,9	3
Ease of parametrisation	0,7	0,9	
Readiness of change	1	0,25	4
Time spent to implement the change for user's satisfaction	1	0,2	2
Time spent to implement the change by the maintainer	1	0,2	5
Stability			
Less encountering failures after change	0,7	0,1	6
Localisation of modification (emerging failure after change)	0,8	0,2	1
Testability			
Effortless testing	0,7	0,1	7
Readiness of built-in test function	0,3	0,9	8
Test restartability	0,4	0,8	
Compliance			
Satisfaction coverage of compliance items relating to maintainability	0,5	0,8	10

Table 2: *Reflector*’s current and target values of external metrics for maintainability

The working processes (and their respective parts of the workflow models) that would be influenced by each one of the sub-characteristics having priorities 1 – 5 on Table 2, will be identified and mapped to the external maintainability metrics affecting them. This will be done by tagging the relevant parts of the workflow models (developed at the previous stage), with kind and degree of impact of the maintainability metric improvement on the working process.

3.4. Determine necessary process changes

The product goal setting will be followed by identification of the process changes that can contribute to achievement of the product improvement goals, ranked at the previous stage. A preventative rather than a corrective quality approach to process changes will be undertaken.

The relevant PPDs will be retrieved by the external PROFES experts (and members of the SPI working group), from the PROFES PPD repository that they have access to. The suitability of the proposed PPDs will then be evaluated based on the context information provided in the PPDs, relative to the context of this case. One example PPD (regarding the software testing process) selected from the repository is shown by Figure 3. An example of the overall processes (as preliminary candidates for improvement), which will potentially be identified from the selected PPDs, is shown by the leftmost column of Table 3.

PPD model	
Product quality	**Maintainability**
Process	**Software testing**
Technology	**Hudson continous integration engine**
Context	
Overall time perssure in project	**low average** high
Experience of project team	low **average high**
Management commitment to the technology	low **high**

Figure 3: example of a selected PPD regarding software testing process

Table 3 provides an example of the processes assumed to be selected for improvement. The final version will, by the SPI working group, be deduced specifically for *Reflector*. The PROFES experts will propose a selection of the processes (based on the context information and the goals set at the previous stage), and look for an agreement by the rest of the SPI working group. Further analysis of the selected PPDs and discussions within the SPI working group will classify the selected processes with particular product quality impact into another two groups: processes with highest improvement potential (the highest improvement potential selection, i.e. a subset of the leftmost column of Table 3), and processes that can actually be modified (a selection of the feasible ones, i.e. a subset of the leftmost column of Table 3), respectively. The selection of the 2. column of Table 3 will be based on:

- the PPDs that the process list originally was retrieved from
- a cost-benefit analysis and
- the original problem analysis which revealed the main weaknesses of *Reflector*.

Processes with particular product quality impact	*Processes with highest improvement potential*	*Processes that can actually be modified*
Conformance to standards	√	
Software requirements analysis (with formal quality metrics)	√	
Software architecture design (with modularity etc.)	√	
Software detailed design		
Software implementation		
Software testing		√
Software integration methodology (incremental techniques, atandard technologies, middleware etc.)	√	√
Software integration testing	√	
Software operation (with configurability, recovery mechanisms, a takeover environment, fault reporting etc.)		
Risk analysis	√	
Configuration management	√	
Enforcement of a particular delelopment methodology		
Monitoring (with metric based KPIs)		√

Table 3: the perspective processes for changes

Finally, a feasibility study based on PROFES experts' survey of experiences from similar SPIs and interviews of technical experts and project managers from SF and CIS will provide indications with respect to which processes actually can be modified (3. column of Table 3). Table 3 provides an example which summarizes the potential results regarding the process improvements related to both SF and CIS.

Similarly, CIS' measures on own processes will be outlined. The possible outcome is, for example:

- A dedicated integration testing environment as a clone of the rest of their operational environment
- Inclusion of some of the metrics (to be identified at a later stage as a part of the GQM plan) as KPIs in the monitoring environment.
- Development of more precise and formal *Reflector* change requirement specifications which are to be delivered to SF'

SF should simultaneously adopt the Agile methodology, based on model driven development (MDD) in relation with both architecture design and implementation.

3.5. Describe process changes

Based on the output (summarized in Table 3) from previous stage, the processes and practices that need to and can be changed, replaced or adapted in the existing process model, will, by the SPI working group, be marked on the existing (previously developed) workflow models. This will be done by first refining the processes that can actually be modified (for example the ones from the third column of Table 3 or similar which have been deduced), and then tagging and extending the ABC Flowcharter models presented in Subsection 3.2 in order to point to the affected parts.

Then, a desired (objective) process model will be developed, by modifying the existing process model with the chosen process changes and adding work instructions on how to perform the processes, by whom and with which tools/resources. This will be done by the SPI working group, together with selected project personnel from both SF and CIS. Additional interviews about the current form of the processes selected, will be conducted with CIS process engineers. Moreover, the available information (provided by both SF and CIS) from two earlier (after release) post-mortem analysis, test and quality reports regarding *Reflector*, will be used. The overall process participants (everybody at SF and CIS who will be using the new process model directly or indirectly) will thereafter be:

- informed about the changes proposed and their assumed implications, and

- trained to use the new (objective) process model. Training will be provided in form of tutorials and manuals.

Refinement of the objective process models will proceed until they are fully understood and accepted by all the parties involved.

SF's adoption of Agile methodology will involve that the designers and developers work tightly together, both of them using abstract modeling tools to solve their tasks, and that each iteration results in a working system with additional functionality (in relation to the previous iteration).

3.6. Set metrics for the process and product improvements

The formal measurement goals will, by the SPI working group, be defined on the PROFES template for measurement goals. The measurement goals will be based on the earlier deduced product quality goals, process capability goals and product-process dependencies. An example measurement goal is provided by Table 4.

The project team members at SF and CIS will be consulted by the SPI working group and a number of measurement goals will be specified after an approx. four hours long brainstorming session among the project team members and the PROFES specialists from the SPI working group. Then, the SPI group will prioritize and select measurement goals.

Object	System testing
Purpose	Evaluation
Quality focus	Correctness, defect removal
Viewpoint	System tester (SF), quality assurance manager (CIS)
Context	Software Factory on Reflector

Table 4: an example measurement goal

GQM [13][14] questions and metrics will thereafter be deduced (based on the above identified goals) by performing structured interviews of project team members from both SF and CIS. The interviews will be conducted by the PROFES specialists from the SPI working group. Abstraction sheets will be used during the interviews. Interview reports will be made and analyzed, by the PROFES specialists from the SPI working group. The identified measurement goals will be refined into questions at a level suitable for interpretation of goal achievement. For each question, expected answers will be formulated as hypothesis. Then, both hypothesis and questions will be reviewed.

Thereafter, the PROFES experts will refine the questions into quantitative metrics which identify the measurements. Some metrics will be directly measurable, while others may be deducible (indirect metrics) from the measurable ones. Throughout this phase, checks of consistency and completeness of goals, questions and metrics (GQM), in relation to the process and product (design) models of *Reflector*, will be performed in form of inspections. In case of inconsistency of GQM plan with the process and product models, adjustments of GQM will be made.

Based on the so far provided goals, questions, metrics and hypothesis, a GQM plan will be assembled, as a basis for measurement and analysis planning. GQMaspect tool will be used for formulating the final GQM plan. A small extract of an example GQM regarding Monitoring with metric based KPIs is shown by Table 5. The metrics should correspond to the ones used in 3.3.

Goal	Question	Metric (or KPI)
Monitoring (with metric based KPIs)	What is the quality of the inspected software?	Average faultes detected per KLOC
		Average inspection rate
		Average preparation rate
	To what degree did the staff conform to the procedures?	Average inspection rate
		Average lines of code inspected
	What is the status of the inspection process?	Total KLOC inspected

Table 5: a small extract of a GQM regarding Monitoring with metric based KPIs

The measurement plan will specify (for each measurement) who should collect it, when and how. In addition, the measurement plan will include a specification of all tools for automatic data retrieval and templates for manual data collection. An example part of a measurement plan describing an indirect measurement identified in the GQM plan, is given in Figure 4.

Formal definition:
X: avg. time to test after failure resolution
T: time spent to make sure whether the reported failurewas resolved
Number of resolved failures
Possible outcomes of the measurement: 0 - ∞
Measurement collected by: Rick Hall, maintainance engineer at SF and Eli Piani quality engineer at CIS
Measurements to be collected during weeks 17-20
Measurements to be collected using Hudson tool (SF and CIS) and CIS' monitoring environment

Figure 4: an example part of a measurement plan

Finally, the GQM plan and the measurement plan will be reviewed by all involved project members (who will potentially be involved in the data collection) at SF and CIS. The PROFES experts from the SPI working group will lead the review process, and focus on reaching an agreement with respect to goals, questions, metrics, definitions, data collection procedures and tools.

3.7. Prepare improvement implementation

This particular stage aims at putting all the conditions and prerequisites for the actual implementation of the improvements and the measurements, in place. A committee represented by top and project management from FS and CIS and one PROFES expert (who is appointed the committee chairman) from the SPI working group, will be established. Their task will be to arrange progress meetings and follow up the process improvement implementation, at the following stage.

Next, the practicalities regarding process improvement steps from 3.5. and measurements from 3.6., will be planned in further detail, by the SPI working group. The PROFES improvement plan template will be used for this purpose. A detailed schedule containing process improvement progress meetings, milestones, feedback sessions, training, promotion, assistance, guidance, GQM measurement, partial implementation etc., will be made. Each task will be allocated the necessary resources and assigned the responsible individuals.

Pilot project groups will be formed at SF and CIS, and provided a separate infrastructure.

The implementation of changes will be started by a kick-off meeting. The progress meetings committee, the SPI working group and numerous SF and CSP employees will participate. An online help desk will also be installed.

All the involved SF and CIS personnel will reserve necessary time and resources for daily implementation and measurement tasks, and give the SPI a generally high priority.

3.8. Implement and monitor improvements

At both SF and CIS, a pilot group responsible for checking whether the SPI actions are feasible individually and mutually, will be activated first. This group will initiate and observe the each one of the originated SPI actions at a separate environment for two days, before the actual implementation of the SPI actions starts.

The process changes identified, specified and modelled in Subsections 3.4 and 3.5 will, step-by-step, be implemented according to the process model. The process changes will, by the SPI working group, be coordinated with the now shortened software development iterations at SF and more frequent integrations of new releases, at CIS.

Expected quantifiable impact of each modification will be analyzed and documented prior to the implementation and measured thereafter. Similarly, the impact of a set of modifications will be analyzed and documented for each release – thereafter measured. This procedure will be followed by both SF (in relation to development of a new version) and CIS (in relation to integration of the new version or modified integration of an existing version of *Reflector*).

Measurement data according to the GQM measurement plan developed in 3.6., will be collected. Most of the measurement collection will be implemented by the PROFES experts from the SPI working group and project personnel from SF and CIS, using existing monitoring environment and databases. The project quality managers from the two respective organizations will be responsible for the measurement and communication with the SPI working group (through the *Reflector* project manager from CIS – having measurement facilitator role in the SPI working group).

The reference specification developed in Subsection 3.2. will, once again, be quantified (see the six quantifiable bullet points in 3.2, for each task specified).

The data collected will, by the SPI working group, be prepared for the feedback sessions. The most useful measurements will be focused on. They will first be gathered and then processed using MetriFlame and MS Excel toolsets. The confidential parts of the data will be protected. Different views of the measurements will be created (charts, histograms and statistical operator presentations) and their correlations examined. MetriFlame will assist collecting the data, analyzing them according to the defined metrics and presenting them. Excel will be used for additional charts and special layout productions. The analysis results to be presented at the each forthcoming feedback session, will be selected. Some of the data may be incomplete or unreliable, and therefore omitted. A slide presentation will summarize the analysis.

A preliminary process assessment (of the changed processes) based on Bootstrap and supported by Bootsample will, once again, be undertaken, by the approach similar to the one described in 3.2.

Feedback sessions will be organized weekly, after each iteration. Findings and results of the collected measurements will be discussed and interpretations of the collected data derived. Experiences from process changes will also be shared. Project teams from SF and CIS, as well as the SPI working group will participate at each feedback session. Comments will be provided and recorded in the session minutes. The measurements themselves will be discussed, in order to reveal difficulties and weaknesses. Some of the PPDs will be evaluated during the feedback sessions, using data from product measurement and the preliminary process assessment. Moreover, the impacts of the improvements will be frequently discussed and process changes modified accordingly. PROFES feedback session templates will be used.

3.9. Evaluate results

When the implementation of the improvements is completed and most measurement data and lessons learned are available, evaluation of measurement results will start.

The data collected will, by the SPI working group and the quality engineers of SF and CIS (who also previously retrieved some of the data), be analyzed through charts, tables, statistical operators etc. The product improvement goals exemplified in the third column of Table 2 will, upon completion of the SPI implementation and measurement, be compared with the actual measurements. The current values will be compared with the ones before the implementation of the improvements (column 2 of Table 2) and the targeted ones (column 3 of Table 2). In addition, the measurement data will be evaluated according to the GQM plan. The dependencies expressed by the GQM will be evaluated together with the project personnel of SF and CIS.

Next, the PPDs selected will be evaluated. The dependencies assumed by the PPDs will be evaluated through the measurement results. PPD models used will, based on the measurement results, be confirmed correct, modified or rejected, depending on the evaluation of the measurement results. MetriFlame and MS Excel will be used for data analysis support.

Moreover, the values regarding the reference specification from Subsections 3.2. and 3.8., respectively, will be compared and analyzed for their validity.

The process assessments undertaken in 3.2. and 3.8., respectively, will be compared. The overall process improvements will be related to the *Reflector* quality improvements, and the overall process-product dependencies evaluated, relative to the underlying dependency assumptions and validity.

General experiences (not available through the measurement data) with use of the PPDs will be included in the evaluation. A post-mortem analysis will be conducted with the management of SF and CIS, to uncover the experiences regarding the SPI programme and the preliminary results. The results of the evaluation session will be documented in a final report.

3.10. Update experience base

Since no experience base exists at SF or CIS from earlier, the infrastructure will be built (by technical managers of SF and CIS, respectively) in form of a non-commercial multimedia database with advanced search opportunities and user friendly editing of new contents.

The lessons learned will then be documented in an experience base for future use. All the SPI programme participants will contribute with their knowledge, data, and the relevant approach and methodology information. The experiences will be described explicitly and include information at all appropriate levels of detail. The documentation will include: problem analysis, process models, PPDs, GQM plan, measurement programme, measurement data, interpretation of the measurement data, data quality analysis, suggested improvements of products/processes, informal lessons learned etc. The documentation will be packaged and extended with various meta-information (for better accessibility) before being stored in the final version in the experience factory. The PROFES expert responsible for this stage will perform quality assurance of the information provided. He will remove overlapping and confidential information, check whether all information has been provided, make references among the documents, supplement with explanations, keywords, metadata, additional figures, author info, general SPI programme info, etc. The packaging will also involve describing application domain and structuring the various artifacts.

The packaged information from this SPI will then be stored and disseminated throughout SF and CIS/CSP by sending an email with a link to an intranet-address to all the employees.

4. Rationale of the improvement plan

The improvement plan addresses maintainability directly and lead time indirectly, of an outsourced system with separate development and operational environments. In addition, the users have different background and needs, and are distributed across the country. In such a complex setting it has been necessary to provide a detailed SPI, not deviating too much from the validated and established PROFES methodology. Such an approach will contribute to the predictability of the results and a controlled SPI, without risks of major deviations or delays. This section presents the rationale behind some of the decisions made and discusses the uncertainties faced in the SPI plan presented in Section 3. More specifically, the specific choices made in the SPI plan, the role of the initial problem analysis, data analysis, meta evaluation and threats to validity, will be substantiated and discussed in separate subsections below.

4.1. Specific choices and decisions made

The underlying assumption of this SPI plan is that maintainability improvement and lead time reduction are the main measures towards enhanced change management. These aspects of product quality are further related to the underlying processes adopted by the parties involved. The SPI plan is customized to the context presented in Section 2. Maintainability (according to its ISO 9126 definitions) and lead time are tightly inter-related, as the lead time depends on the efficiency of the maintenance.

In 3.1, both interviews and data analysis are planned, in order to relate high-level user interpretations, with system data and logs. Moreover, 3.1 suggests using a separate server, in order not to overload the production environment. Data analysis and presentation is suggested to be performed by the *Reflector* project manager from CIS and quality assurance and integration manager from CIC, since CIS is the customer, and therefore more likely to be critical and open about system weaknesses. ISO9126 is the source of quantitative and qualitative definitions, since it is unambiguous and known by most of the people involved.

3.2 suggests focusing on SF's process capability, since most of the defects are assumed to originate from the design and implementation stage, which are solely handled by SF. The reference specification introduced in 3.2. is motivated by its practicality for easy comparison and high-level analysis both during the SPI programme and in the future. It is informal and comprehensible, while covering many of the relevant aspects which quantitatively indicate the status of the system quality and process capability. Some of the measurements (bullet points quantified for each task of the reference specification) are product, rather than process related. Still, they are suggested to be conducted at this stage for convenience reason, and due to readiness of the setup.

In order to additionally verify the conclusions from determination of product quality needs, current product quality and current process status, the working group will conduct separate interviews and discussions planned in Section 3.3. The example provided in 3.3. serves as an aid for understanding the expected form and contents of the outcomes of the stage.

Since predictability of the system quality is an important aspect, 3.4 suggests a preventative, rather than a corrective quality approach to planning process changes.

One of the assumed goals regarding the lead time is reduction of *Reflector* release cycles from one month to one week. 3.4. suggests SF's adopting Agile methodology, based on model driven development (MDD). Agile methodology based on MDD is expected to shorten iterations, improve documentation and increase consistency between design and implementation. This will make the two organizations more compatible in terms of methodological thinking and practical collaborations, since CIS already uses the SCRUM methodology. SF's use of MDD will most likely result in much shorter cycles which could be more synchronized with CIS' change requests and software management processes. The adoption of MDD is expected to solve many of the testing issues, as SCRUM (used by CIS) pre-defines tests as a part of each development cycle – an approach called continuous integration which is possible to synchronize with the iterations of the Agile approach. CIS's use of continuous integration approach (supported by Subversion [8] database environment and Hudson [6] testing application relying with each revision's predefined test rules) can harmonize their *Reflector* integration testing into the overall development process (by SF) and thus be able of uncovering defects much earlier.

The process changes description in Section 3.5. is a refinement of the feasible processes deduced during the stage presented in 3.4. The desired process model will integrate all the changes and present the objective model. The additional interviews conducted in 3.5. aim at sufficiently understanding the relationships and the effects of the changes.

During goal interpretation in 3.5, the hypothesis on measurement results will be compared to the actual measurement results, in order to identify and analyze the underlying reasons for deviations or confirmations with respect to the expectations.

In 3.7., a pilot group is formed at both SF and CIS, in order to check the impacts of the SPI actions.

In particular, the pilot group will in 3.8. focus on testing the interdependencies of several actions being deployed simultaneously. The interdependencies (with respect to both processes and the product quality) of the actions, are not possible to retrieve from existing PPDs and therefore difficult to predict. Both monitoring data and subjective data (from feedback sessions) will be collected and documented during the implementation, in order to allow for their relating and comparison during the forthcoming (evaluation) stage. Measurement data, according to the GQM measurement plan developed in 3.5, will be collected in 3.8. The purpose is to monitor the improvement actions and their effects, and to take corrective actions if necessary.

An additional process assessment will be undertaken (see 3.8.) in order to allow for comparison with the initial process capability determined in 3.2.

The evaluation presented in 3.9. will make comparisons of the achieved state of the processes and the system quality with both the state determined prior to the SPI programme realization, and the desired state – stated in terms of product and process improvement goals.

The purpose of the evaluation of the PPDs in 3.9 is to evaluate whether the product improvement goals have been achieved by the changes made to the processes, methods and tools. The dependencies assumed by the PPDs will be evaluated through the measurement results. The SPI working group will evaluate the collected product and process related data in order to find out whether the PPD models used were appropriate in the *Reflector* context. The dependencies can normally only be substantiated, not entirely refuted or proved. Further, the SPI working group will gather and analyze the data and experiences in order to improve management of product quality-based process improvement projects in the future.

The experience base presented in 3.10 collects both short term (current, *Reflector* specific) and long term (generic) information, in order to provide a double-loop feedback, for future use in similar environments. Of particular interest is the information on the mutual dependencies of the SPI actions, which is not stored in similar repositories.

4.2. A through initial problem analysis

A controlled improvement of a system is dependent on the understanding of what is currently present, as changes in, for example architecture design and technology, are affected by the prior situation (context, operational profile, system quality, process capability etc.). The problems that occur when changes are being made to poorly understood, communicated or specified problems have been highlighted by e.g. [3]. An important problem is architectural erosion: “A system that is being changed when the architecture has not been understood erodes into an entity where new change requests are becoming harder and harder to fulfill, and eventually a change request is either impossible to accommodate, or it results in more new errors than those potentially being fixed. This is a serious problem in domains where lead-time is an important issue, since lead-time accelerating activities such as product-family or product-line reuse requires a strong grip of the architecture. [4]”

The above presented arguments emphasize the importance the initial problem analysis. The contents of the initial determination of system quality and process capability, provide the rationale for many of the choices made in the remainder of the SPI plan. A through problem analysis helps focus on the prevailing issues, the issues having highest impact and the feasible ones. This is why our characterization section is dedicated a significant part of this SPI plan, and why many of the specific actions and choices will depend on the results of 3.1. through 3.3.

4.3. Data analysis and information exchange

Measurement in PROFES is goal-oriented, according to GQM paradigm, which means that metrics are derived from goals via questions. Interpretation of the collected data will be done in a bottom-up way. Due to CIS’ very good and established monitoring environment (containing the data collected during the three years of *Reflector* operation), the data mining techniques will be exploited, in terms of both prediction and description.

The feedback sessions are essential for analysis and interpretation of the measurement data. They can encourage creativity of the participants and often result in immediate process change, without delays.

Feedback sessions are also the key for achieving positive results. This case has particularly many stakeholders, which makes efficient communication among the parties involved even more critical for success of the SPI. The feedback sessions will facilitate coordination and unite the parties involved in SPI. The feedback sessions will be followed up by the SPI working group. The minutes will be written, open questions resolved and the findings disseminated. All identified problems and suggestions will be followed up. The feedback session meetings will seek to provide valuable and relevant input to the audience, in order to maintain the motivation for meeting up.

4.4. Perspectives of a meta evaluation

"According to the principle of continuous assessment, measurements collected during development project implementation are used to evaluate process."[1] The data collected should therefore be applicable for performing a meta analysis of this SPI. A complete plan for a meta analysis is beyond the scope for this SPI plan, but could be useful for the organizations involved, particularly with respect to the documentation of the overall results, general lessons learned, cost-benefit analysis and as a basis for the future decision making with respect to whether and under which conditions a new SPI should be launched in the future.

The overall evaluation criterion should be the cost-benefit relationship of the SPI, from point of view of the methodology user. To what degree do the benefits of the results of SPI outweigh the costs of planning and conducting it? And what are the alternatives to this particular SPI plan, in order to achieve comparable benefits? In our case, exchange of *Reflector* is not an option, so improvements on the current practices and tools have to be imposed. In addition, the importance of *Reflector* substantiates to a large degree the investments that will be made. Measurement of the actual benefits is difficult to do as a part of the improvement process, as the long-term benefits would be neither measurable nor predictable. The PROFES method itself has been evaluated (see appendix 5 of [1]) and the results have shown the significant outweigh of the benefits compared to the inherent costs. This in itself is a significant argument for the initial decision making on whether to start up an SPI programme, in the future.

When evaluating (see 3.9), it is important to repeat a process assessment just on the processes affected by the implemented improvement actions. The measurement and the variation factors should be kept as controlled as possible. Only the measured aspects should be varied at a time, then combined in order to reveal possible interactions between the independent variables.

An analysis across SPIs and other (case) studies could be possible. [16] proposes metrics for cohesion, coupling and visibility in order to evaluate high-level design of software systems. These are among the means for measuring and assessing maintainability which, with our extensive access to the system design and implementation (within SF) during the SPI programme, could be tried out in order to evaluate overall maintainability and check whether the findings would be similar to the ones presented by [16]. Another highly relevant approach is presented in [15]. It uses actor-dependency modeling to model and analyze a large scale maintenance organization. It captures properties of organizational context and maintenance process, in order to understand flaws found in the process. In our case with many actors, this could be an approach worthwhile trying and testing whether similar experienced and results to the ones presented in [15] would be acquired.

4.5. Threats to validity and reliability

This SPI can be considered a case study. When conducting an empirical study, it is important to be aware of potential threats to the validity and reliability of the obtained results and derived conclusions. Three types of validity threats presented in [18] should be analyzed: construct validity, internal validity and external validity.

Construct validity concerns whether we measure what we believe we measure. The GQM plan, the well formulated questionnaires, comprehensiveness of the goals and measurement tasks, multiple and differentiated data retrievals and interview subjects, are among the measures imposed towards construct validity. Data quality is essential in this context and therefore dedicated a subsection below.

Internal validity (which concerns causal relationships within the study) is a concern in this study since we establish a relationship between the product quality and the process capability. The former one is improved through improvement of the latter one. Several interviews and data retrievals, as well as use or the already validated PPDs are among measures used to preserve the internal validity. We still, however, need to question representativeness of the system quality determination that will be done and the processes selected to be influences through the SPI actions.

Both construct validity and internal validity are issues in design, quantification and evaluation of the reference specification introduced and measured in 3.2 and then measured again in 3.8. In that context, one should question the representativeness of the tasks, as well as both the internal and the construct validity of the measurements.

Being able to trust that the quality improvement is result of the specific actions, and not just a side effect, is necessary for the long term use of the experiences gained. External validity concerns the generalization of findings of this case study to other contexts and environments than the one studied. Our findings will necessarily be limited to the context, unless a meta analysis across different studies can deduce relationships independent of the *Reflector* context.

The reliability is concerned with demonstrating that the operations of the case study can be repeated with the same results. It should be possible to repeat many aspects of this study, regarding the context independent parts for which the repeated measurements will lie within acceptable thresholds. The SPI plan is however customized for the particular setting presented in Section 2. Many SPI aspects which are irrelevant to this particular case (but potentially relevant in other settings) have been omitted. This SPI plan on this particular case will most likely provide similar results, if repeated with the same starting point. The context relevant and the subjective input may however threat the reliability. The analysis participants should in each case be highly competent and involved, in order to provide similar input. In addition, a necessary degree of estimation (interview objects) and measurement (data) certainty should be demanded. PROFES (which this SPI is based on) provides generic parts of approach which contribute to reliability.

4.5.1. Data quality

The two main forms of input in this SPI will be interviews and empirical measurements. One can and should question the data quality of this kind of input. Data quality is one of the major threats to the validity of results. A possible source of bias relates to the data used, e.g., its measurement accuracy and representativeness if the results are to be generalised. Despite the general suitability of the data, the sampling procedure might bias the results and prevent generalisation. Selection of the acquisition method and the model describing relationship among the data may be another possible source of bias. The way statistical methods are used, for example, may invoke numerous issues regarding validity. The study design, the choice of appropriate algorithms, the fitness of the input, statistical power, representation of the analysis and definition of null hypothesis, are among the examples of possible threats to validity.

Despite full access to data, the measurements within CIS and SF will have to rely on a selection of the available ones, for practicality reasons. This naturally constrains the generality of observed results as well as replication by other practitioners. Furthermore, different accuracy indicators may be used across data retrievals, possibly leading to contradictory results, especially if varying data acquisition quality results in inconsistent and invalid input. In a case with several data sources and parties involved, this can be a threat. Therefore, tight coordination of both data retrieval and analysis, is needed. Moreover, when interpreting the measurement results, statistically insignificant result must be omitted and use of too small data samples to analyse correlations must be avoided. The context needs to be made explicit, in order for the results to be evaluated and reused in the future SPIs.

The subjects measured or interviewed must be representative, significant, and obtained by random sampling within the target group. The sample should be of sufficient size for the model being tested. Statistical techniques for determining minimum sample size exist. They are calculated with basis of probability for rejecting null hypothesis with a specified significance level. Inclusion and exclusion criteria for both subjects and input must be clearly defined. Subjects should be allocated to treatments in an unbiased manner, preferably by randomisation. Although blinding is common in statistical methods, it is often unfeasible in software engineering, since human subjects are aware of the technology they are using, and since redundant systems with and without testing objects, are too costly to build and deploy. Calculating and complying to a reliable sample size may be difficult, but combining a judgemental approach with a normative one, makes this accomplishable in a sufficiently accurate manner.

Non-randomisation is in some cases more appropriate if sample size is too small. This will be appropriate in relation to interviews at SF and CIS. Similarly, some adjustment after the input acquisition can help deal with biased groups. A potential mistake is however use of analysis and interpretation techniques which only apply to randomised samples. Bias is often related to small sample size, which makes randomisation unfeasible. The GQM measurement plan should specify how bias will be detected and handled. Particularly significant is dealing with confounding variables and missing data. In the latter case, insertion of estimated values (based on expert judgement) being within a threshold of uncertainty, is an option. Finally, all treatments and outcome measures should be fully defined and justified.

Statistical power is the probability that a hypothesis test will correctly reject a false null hypothesis. It is the probability that an effect which exists, will be found. High statistical power level, low significance criterion, large effect size (the degree to which the phenomenon under study is present in the population) and large sample size (number of subject studied) decrease the probability of incorrectly rejecting hull hypothesis. Omitting statistical power analysis is most likely to produce results that are inconclusive, incorrect (due to acceptance of false null hypothesis) or uninteresting (due to small effect size). Thus, incorrect or missing statistical power analysis in empirical software engineering may fatally flaw the research, as profoundly elaborated in [12]. It is a known fact that statistical power is difficult to apply to the design of a software engineering empirical study. However, any hypothesis test without sufficient statistical power is meaningless, since it fails to provide enough information to draw a reliable conclusion. This obviously makes meta analysis impossible as well. The data collected should be significant, complete and sound. These aspects can be threatened by inadequate metrics, imprecise measuring instruments, frequency of acquisition or semantic properties of the data. All measures, including ratings, attributes, factors, entities, characteristics, units and counting rules, should be fully defined. Additionally, any measurement performed has to be defined in an unambiguous manner. These aspects will be relevant for GQM evaluation.

Completeness and accuracy of data collection should be ensured by a quality control method. One also has to make sure that data about the subjects (data and individuals) that drop from study, are recorded. If ignored, dropouts may contain significant information and their absence may bias the result.

The data should be analysed in accordance with the study design. One should be aware of the limitations of multiple analysis of the same datasets. The number of positive or negative results should be considered in relation to the total number of tests performed.

Data may be biased, noisy, insignificant due to changes over time, inconsistent etc. A step towards revealing this and model inconsistencies is sensitivity analysis. It will uncover presence of outliers and impact of the individual parameters on the overall model. Moreover, one has to make sure that the assumptions made on data prior to their acquisition, are not violated at a later stage. For example, if the design suggests a certain distribution, then one has to confirm that the data conform to that distribution.

Data screening and visualisation may be useful prior to detailed analysis in order to gain an early, coarse overview. Most analysis are performed by statistical software tools. One should however not blindly trust the tools or the correctness of the provisioning process. Therefore, the tool based analysis should be complemented by another analysis method, e.g., a thought experiment.

The data quality is, for example, the rationale for the many additional evaluations in this SPI, such as the one in 3.3. If not having a sufficient quality, the measurements (for evaluating metrics presented in 3.6) may bias the answers to the questions within the GQM plan.

4.6. Discussion

One of the main success criteria of this SPI plan is its applicability and effectiveness. Relying on PROFES, which is a verified method, is one of the factors contributing to the predictability of the outcomes. PROFES' being based on the well-established and developed approaches to improvement, is another factor contributing to the method applicability and predictability of the outcomes. In addition, the SPI's being goal-driven and modular helps focus on the right aspects, structure efficiently and choose the measures with most impact on the product quality. The modularity of PROFES allowed for easy adaptation of the SPI plan to suit the resources and needs of this particular case. Time did not need to be spent on deducing steps or definitions, as the methodology already offered a comprehensible and applicable approach.

The improvement goals set early in the project plan have been systematically focused on. The SPI programme will take eight months in total (see Figure 2). The question is however to what degree the quantified goals will remain unchanged in the mean time. The SPI working group and the feedback sessions have to raise this question during the SPI programme.

This case uses ISO 9126 as the kernel for characterization of the current and the aimed product quality, as well as the basis for defining metrics. This is in order to avoid any definitional ambiguity and the since the definitions were well applicable in this context.

One of the measures proposed is reduced iterations by an agile approach within SF, supported by model driven development. This was partially motivated by [10] which argues for suitability of PROFES for use in an Agile process.

The effort reporting should be emphasized since the effort information about the actual improvement cycles allows to perform better estimates in the future and is necessary to perform an evaluation of the improvement costs against improvement benefits. Effort data collection should be supported by the PROFES templates.

It is crucial that the GQM is complete and consistent in relation to process and product (design) models from 3.6. Traceability between GQM and the process and product models is a prerequisite for meaningfulness of the evaluation through measurable metrics.

In order to reuse the experiences from this project in the forthcoming ones, the lessons learned will, as presented in 3.10, be documented in an experience factory for future use. It is important that experience factory includes both positive and negative aspects, findings and dependencies revealed throughout the project. For future use, there is a value in being aware of the failed assumptions and improvement attempts. Post mortem analysis should also address both the positive and the negative experiences from the SPI.

One should question how and to what degree the improvement measures proposed in this SPI will influence quality attributes of *Reflector* beyond maintainability and lead time? There may be both short term and long term, positive or negative, indirect effects on overall quality aspects of the system, which should be examined. This is however beyond the scope of this particular study.

After completing the SPI, continuous assessment can be used in the future for less extensive adaptations, since the *Reflector* system is undergoing frequent modifications. This will be feasible since iterations are expected to be significantly shorter and since the necessary infrastructure and experiences already exist and can be utilized in further SPI of *Reflector*.

The SPI working group will, during the entire process, hold most of the information of the SPI plan, progress and experiences completely open for all the employees of SF and CSP. This is a strategic decision, which will facilitate: trust to the SPI group and the SPI programme in general, an open dialog, high participation and generally a positive cooperation within and across SF and CSP.

For example, 3.5. suggests that project personnel is involved in development of the perspective process model, in order to ensure its technical quality and maintain the motivation of the project personnel, while in 3.9. suggests that the project personnel of SF and CIS are involved in the evaluation on order to ensure ownership of the new processes and disseminate the lessons learned.

5. Conclusions and further work

A detailed SPI plan for *Reflector* has been developed and its rationale presented. Maintainability and lead time have been focused on, as the main measures towards improved change management. This report includes context information (regarding the application, the parties involved and the related processes), an SPI plan based on PROFES methodology and finally a rationale and discussion of the SPI plan proposed. The focus has been directed to the parts of the SPI which are most critical with respect to ensuring the validity of the results.

The achievements of the SPI programme are expected to be beyond the pure product quality. Both SF and CSP will most likely, as a result of the SPI, and in addition to improved *Reflector* quality achieve:

- advanced team knowledge on software development and measurement programmes
- better understanding of the role of the processes and persons involved
- improved knowledge management
- increased team building and organizational cultures.

Given the nature of the *Reflector* system (which undergoes frequent changes) and all the artifacts expected to be produced during this SPI, application of continuous assessment will be an efficient approach for further constant evaluation of incremental system modifications. Any possible problems in the future could then be identified early on and resolved quickly, due to the rapid progress and short feedback cycles in both PROFES and Agile model driven development (to be adopted by SF and already present at CIS). One can integrate the advanced analysis techniques (such as Classification and Regression Tree, for example) into the monitoring environment and provide both single loop (short term, system specific) and double loop (more general, long term) feedback (more on the two feedback forms in [11]) from measurements and evaluations. Such an approach and infrastructure would also allow for continuous cost-benefit analysis.

Further work should encompass a cost-benefit analysis of the SPI and a meta analysis with focus on generalizing the results and customizing the SPI plan to the future needs of *Reflector* and the parties involved in its development, operation and use.

The contractual issues (between SF and CSP/CIS) should also be discussed, planned in detail and formalized. Results from this SPI may provide valuable input for a new, formal contract regarding CSP’s outsourcing of *Reflector* development and maintenance.

References

1. PROFES - User Manual, Profes Consortium. 1999

2. Rising, L. and N. S. Janoff (2000). The Scrum software development process for small teams. Phoenix, AZ;, AG Commun. Syst.

3. Bartthall, L. and P. Runeson (1999). Architecture Design Recovery of a Family of Embedded Software Systems. Proceedings of the TC2 First Working IFIP Conference on Software Architecture (WICSA1), Kluwer, B.V.

4. Dewayne, E., A. Perry, et al. (1992). "Foundations for the study of software architecture." SIGSOFT Softw. Eng. Notes 17(4): 40-52.

5. International Organisation for Standardisation: Iso/iec 9126:2004 - Software engineering - Product quality. ISO 2004. Retrieved on 21.11.2008.

6. "Hudson: an extensible continuous integration engine." Retrieved 19.04.2009, 2009, from https://hudson.dev.java.net/

7. Schwaber, K. and M. Beedle (2001). Agile software development with Scrum, Prentice Hall PTR Upper Saddle River, NJ, USA.

8. Pilato, M. (2004). Version Control with Subversion, O'Reilly & Associates Inc. Sebastopol, CA, USA.

9. Similä, J., Kuvaja, P. and Krzanik, L. (1995): BOOTSTRAP: A Software Process Assessment and Improvement Methodology, International Journal of Software Engineering and Knowledge Engineering, Vol. 5, No. 4, 1995, pp. 559-584.

10. Jedlitschka, A., D. Hamann, et al. (2005). Adapting PROFES for use in an agile process: An industry experience report. PROFES 2005, LNCS 3547, pp. 502–516, 2005.

11. Birk, A. and D. Pfahl (2002). A systems perspective on software process improvement, Lecture notes in computer science, 2002 - Springer-Verlag Berlin Heidelberg 2002.

12. Miller, J., et al., "Statistical power and its subcomponents - missing and misunderstood concepts in empirical software engineering research ". Information and Software Technology 1997. Vol.(4): p. 285-295

13. Basili, V., Caldiera, G. and Rombach, H. The goal question metric approach. (Encyclopedia of Software Engineering), 1994. Retrieved on 26.02.2009.

14. Basili, V. (1992). Software modeling and measurement: the Goal/Question/Metric paradigm, University of Maryland at College Park.

15. Briand, L., Walcélio, M., et al. (1995). Characterizing and assessing a large-scale software maintenance organization. Proceedings of the 17th international conference on Software engineering. Seattle, Washington, United States, ACM.

16. Briand, L. C., S. Morasca, et al. (1993). Measuring and Assessing Maintainability at the End of High Level Design. Proceedings of the Conference on Software Maintenance, IEEE Computer Society.

17. The free dictionary. Retrieved on 10.05.2009 from http://www.thefreedictionary.com/

18. W. R. Shadish, T. D. Cook, and D. T. Campbell. Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Houghton Miffin Company, 2 edition, 2001