Targets, drivers and metrics in software process improvement: results of a survey in a multinational organization

Software Qual J (2007) 15:135–153
DOI 10.1007/s11219-006-9007-y

Targets, drivers and metrics in software process improvement: Results of a survey in a multinational organization

Jos J. M. Trienekens · Rob J. Kusters · Michiel J. I. M. van Genuchten · Hans Aerts

Published online: 24 January 2007

Abstract This paper reports on a survey amongst software groups in a multinational organization. The survey was initiated by the Software Process Improvement (SPI) Steering Committee of Philips, a committee that monitors the status and quality of software process improvement in the global organization. The paper presents and discusses improvement targets, improvement drivers, and metrics, and the degree to that they are being recognized in the software groups. The improvement targets 'increase predictability' and 'reduce defects' are being recognized as specifically important, joined for Capability Maturity Model (CMM) level three groups by 'increase productivity' and 'reduce lead time'. The set of improvement drivers that was used in the survey appears to be valid. Three improvement drivers that were rated highest were: 'commitment of engineering management', 'commitment of development staff, and 'sense of urgency'. Finally, it could be seen that metrics activity, both in size and in quality, increases significantly for CMM level three groups. However, no consensus regarding what metrics should be used can be seen.

Keywords Survey amongst software groups · Improvement targets · Improvement drivers · Usage of metrics · Software process improvement

1 Introduction

This research was carried out in the software groups of Philips. In these software groups, software is developed for a large product portfolio, ranging from Shavers to TVs to X-Ray equipment. Software size and complexity are increasing rapidly and the total software staff is growing continuously, from less than 2000 in the early 90s to more than 5000 now. More and

J. J. M. Trienekens (✉) · R. J. Kusters · M. J. I. M. van Genuchten · H. Aerts
Faculty Technology Management TU/e, University of Technology Eindhoven and Philips Eindhoven, Building: Paviljoen TM, D.15, Den Dolech 2, 5600 Mb Eindhoven, The Netherlands
e-mail: j.j.m.trienekens@tm.tue.nl

more software projects require more than 100 staff-years located on multiple development sites. The software groups also have to deal with the fact that the share of software developed by third parties is rapidly increasing.

Software Process Improvement (SPI) in Philips started in 1992. Capability Maturity Model (CMM[1]) is used for software process improvement (SPI), because CMM is well defined and the measurement of the process improvements is done by objective assessors, supported by jointly developed CMM interpretation guidelines (SEI, 1995). Although Philips is in the process of switching to the successor model CMM Integration (CMMI), the data discussed in this paper refer to the original CMM. Within Philips, the software capabilities and achievements are measured at corporate level by registering the achieved CMM levels of the company's software groups. As a result the measurements are comparable across the organization and the software process capability (CMM-level 1, 2, 3, 4 or 5) of each individual software group is known. Philips developed a cross-organizational infrastructure for SPI. From the organization perspective SPI is organized top-down as follows: Core Development Council, SPI Steering Committee, SPI Coordinators and Software Groups. From the communication perspective the following events and techniques can be recognized, respectively a formal Philips SPI Policy, a bi-annual Philips Software Conference, temporary Philips SPI Workshops, a Philips SPI Award, an internal SPI Web Site, and so-called SPI Team Room sessions.

The ultimate objectives of achieving higher CMM-levels are: increased productivity, shorter lead times, and higher product quality. However, these parameters are not measured and tracked at corporate level. This is because software groups cannot be compared on these parameters, as the productivity level is highly dependent on the type of software being developed, lead times cannot be compared without taking the business and development context into account, and a certain post-release defect density can be acceptable in one business while being totally unacceptable in another business. Although CMM addresses the measurement of process improvement activities on its different levels, it does not prescribe how software measurement has to be developed, and for example what kind of metrics should be used. In accordance with experiences from practice, software measurement requires well-defined improvement targets and metrics, stable data collection, and structured analysis and reporting processes (Trienekens, 2004; El-Emam et al., 2001). To investigate the status and quality of software process improvement targets and metrics in the different software groups at Philips, its Software Process Improvement (SPI) Steering Committee decided to perform an empirical research project in the global organization. It was also decided to investigate the way improvement drivers, also known as critical success factors, were recognized in the different software groups. If possible, relationships had to be identified and defined between improvement targets, drivers and metrics and the level of maturity (CMM-level) of particular software groups. Consequently, the following research questions have been elaborated:

– regarding improvement targets:

  - what are considered to be important improvement targets for software groups,
  - are these achieved, and
  - are these achievements related to particular CMM-levels?

[1] For further information on SPI, CMM and related notions such as CMMI, and KPA see the website of the Software Engineering Institute (www.sei.cmu.edu).

- regarding improvement drivers:
  - what are considered to be important improvement drivers for software groups, and
  - are they related to particular CMM-levels?
- regarding metrics:
  - what is:
    - the level of metrics activity
    - the quality of resulting data
    - the usage of resulting data, and
  - is there a relation between this and particular CMM-levels?

In Section 2 of this paper some demographics of the survey are presented, whilst Section 3 addresses the structure and background of the survey. Section 4 presents and discusses the results of the survey in three subsections. Results pertaining to improvement targets are discussed in Section 4.1, those regarding improvement drivers in Section 4.2, and finally the role and importance of metrics is looked at in Section 4.3. Section 5 presents the conclusions.

2 Survey demographics

This section gives briefly some survey demographics. The survey was aimed at group management and was completed by them. The number of responding software groups was 49 out of 74. This means a, very satisfactory, response of about 67%. Table 1 shows the global distribution over the continents of the responding software groups. Table 2 presents the CMM-levels that have been achieved in the organization.

In the remainder of the analysis the three software groups with CMM-level five are excluded from the analysis. The reason for this is that the total number of software groups on the higher levels was too small, in comparison with level one and two, to allow separate analysis of software groups for each of these higher levels. An alternative solution would have been to combine the higher level groups into a single category (CMM-level 3–4–5).

Table 1 Distribution of responding software groups over the continents

Continent	Number of software groups
Europe	32
Asia	8
America	9
Total	49

Table 2 Number of software groups on the different CMM-levels

CMM- level	Number of groups	Valid percentage	Cumulative percentage
1	20	43.5	43.5
2	13	28.3	71.7
3	10	21.7	93.3
4	0	0	93.3
5	3	6.5	100.0
Not reported	3		
Total	49		

However this category, consisting of a larger number of level three groups and some level five groups, is less coherent. Previous work (e.g. Beecham et al., 2003) suggests that level three groups differ in their behavior from level five groups. Therefore this option was not taken, albeit at the cost of three data points.

Table 2 shows that more than 50% of the software groups that completed the survey succeeded in leaving the lowest level one (initial software development) of the CMM.

3 The survey: Structure and background

The questionnaire is structured in three sections as follows:

1. the attempted and achieved SPI improvement targets;
2. the improvement drivers of SPI programs;
3. the usage of measurement data and metrics.

Each of these will be discussed subsequently in Sections 3.1, 3.2 and 3.3. All questions used in the survey a quoted verbatim, so as to facilitate reuse.

3.1 Improvement targets

The definitions of the improvement targets used in the questionnaire have been derived from earlier research of the SPI Steering Committee in the global Philips organization. This earlier work consisted of a series of structured brain-storming sessions based on publications on the subject (e.g., Paulish and Carleton, 1994). In the latter publication, effects of software process improvement that are mentioned are respectively: ‘fewer product defects’ (in the list below ‘reduce defects’), ‘faster time to market’ (in the list below ‘reduce lead time’) and ‘better predictability’ (in the list below ‘increase predictability’). The following list shows that the SPI Steering Committee took improvement targets from literature but also added the improvement targets ‘increase productivity’, ‘improve cooperation’, ‘improve staff motivation’ and ‘increase reusability’ to the list, in conformance with their own experiences with software process improvement. The resulting improvement target list:

- increase predictability
- reduce defects
- increase productivity
- reduce lead time
- improve cooperation
- improve staff motivation
- increase reusability

For each of these improvement targets a software group had to indicate “to what degree has SPI attention and effort been aimed at each target during the last 2 years”. Similarly they were asked “ to what degree has performance improved in terms of this goal during the last 2 years”. The answers were required on a five point Likert-type scale with end-anchors “little or no attention and effort(1)” to “main focus of attention and effort (5)” for the improvement targets (attention given) and “little or no improvement (1)” to “major improvement (5)” for performance improvement achieved. A drawback is that the interpretation of the meaning of intermediate values is more difficult, although, Cummins and Gullone (2000) conclude that “the end-defined format seems not to bias the data in any particular way”. However, since the scale is clearly one-dimensional, and the anchors (“little or no” versus “main”/“major”) indicate clear contrary (or polar opposite) ends, it is reasonable to assume that respondents view the scale as being symmetrical (see e.g. Skipper and Hyman, 1993). The middle value (3) can thus be interpreted as a halfway point between better and worse. We will use the adjective ‘average’ to indicate this and assume that values over three indicate more than average attention/performance. Moreover, the use of end-anchors greatly facilitated survey lay-out.

3.2 Improvement drivers

The concept of improvement drivers had not been previously discussed with the software groups and had to be explained and elaborated. These drivers, also known as success factors, can be of different types, such as organizational (e.g. commitment of management), human (e.g. resistance), technical (e.g. lack of tools), and financial (e.g. restricted budget). Over the years various papers have been written about these drivers. In the literature factors or drivers are addressed from different perspectives, for example: factors that affect organizational change (Stelzer and Mellis, 1998); factors on different levels of maturity (Rainer and Hall, 2002); factors in large and small organizations (Dybå, 2003); factors that affect software processes (Rainer and Hall, 2003). The objective of these articles is to identify and define drivers so that they can be taken into account when setting up and/or carrying out an SPI program. For this research we discussed with representatives from the SPI Steering Committee the drivers that were addressed in each of the papers mentioned. The resulting list of improvement drivers is as follows:

1. Commitment of business management
2. Commitment of engineering management
3. Sense of urgency and perceived need to improve
4. Commitment of development staff
5. Cooperation of other engineering disciplines
6. Confidence in results of SPI
7. Availability of engineers’ time for SPI
8. Availability of qualified SPI resources
9. Sufficient investment in SPI training
10. Proper tooling to support the processes
11. Use of an accepted improvement framework, such as CMM
12. Clear relation between SPI goals and business goals
13. Clear and quantified improvement targets
14. Visibility of intermediate results
15. Integration of SPI in general improvement activities such as PBE

The identified drivers were included in the questionnaire. The question used was: “Please rate the following improvement drivers on a five-point scale from very unimportant (1) to very important (5). Improvement drivers are defined as those variables whose presence or absence had a key impact on your department’s SPI results over the last 2 years”. Here again a five point Likert-type scale is used, this time with end-anchors “very unimportant (1)” and “very important (5)”. For identical reasons again the assumption is made that the value ‘3’ represents a natural midpoint and that values above ‘3’ indicate an ‘above average’ degree of importance.

3.3 Metrics

The research group used brainstorming sessions and discussions to develop a number of questions regarding metrics usage. Since usage presumes availability of data of sufficient quality these issues were also included in the survey. Questions related to metrics were divided into three categories. A first category focuses on the level of metrics activities in the participating software groups with the overall objective of finding out if metrics are actually collected. An obvious one is "do you have a formal software measurement program in place", answerable on a yes/no scale. Questions "what percentage of software projects do project evaluations" and "what percentage of software projects report quantitative data" provide an indication of the actual level of activities in this area. Both questions are asked on a four point scale (0–25%; 25–50%; 50–75%; 75–100%). Finally we asked: "which of the following quality measurements/metrics do you currently use". For this a list of 17 predefined metrics was provided (with a yes/no scale) that has been derived from earlier research sponsored by the SPI Steering Committee of the global Philips organization. To this list software groups could add their own metrics.

A second category of questions looks at the quality of the data obtained. This was asked indirectly by the question "how closely are your measurements/metrics aligned with business objectives" which was answered on a five point Likert-type scale ranging from "no connection between metrics and business objectives (1)", via "some consideration of business objectives in choosing metrics (3)", to "tight linkage between metrics and business objectives (5)". This is supported by two more direct questions, "do you validate the data you collect" and "are data reliable at holding level" which in the survey was formulated as "are these "quantitative data" reliable enough to be reported at Philips level (like nowadays the CMM-levels)". Both these questions were answered on a yes/no scale.

A third and final category of questions looks at actual usage of the resulting metrics. These questions were asked twice. Directly: "do the measurements/metrics guide you in the SW operation" with a yes/no scale. Also in more detail: "how important are measurements/metrics to your organization for managing software projects", with answer categories:

1. Minimal or no importance;
2. Ad-hoc metrics collection;
3. General performance indicators available for the department (not for the individual projects) and communicated outside the department;
4. Detailed metrics are used for estimating and tracking by some projects and communicated outside the department;
5. Measurements/metrics are used for estimating and tracking by all projects and for making organizational decisions.

4 Results of the survey

This section introduces and discusses the results of the survey. The improvement targets of the software groups are presented in Section 4.1. Section 4.2 reports on the identified improvement drivers of the software groups and Section 4.3 deals with results of the investigation on metrics used.

Table 3 Results for SPI targets

	Mean: attention given				Mean: perceived performance			
Improvement target	All	CMM1	CMM2	CMM3	All	CMM1	CMM2	CMM3
Increase predictability	3.8	3.6	3.7	4.3	2.9	2.3	3.2	3.5
Reduce defects	3.5	3.1	3.4	4.4	2.7	2.3	3.1	3.0
Increase productivity	3.0	3.1	2.5	3.3	2.3	2.2	2.1	2.8
Reduce lead time	2.9	2.7	2.8	3.2	2.1	1.9	2.2	2.3
Improve cooperation	2.7	2.7	3.2	2.4	2.4	2.4	2.4	2.5
Improve staff motivation	2.4	2.1	2.7	2.5	2.4	2.3	2.5	2.4
Increase reusability	2.1	1.8	2.2	2.6	2.0	1.7	2.4	2.1

4.1 SPI improvement targets

This section will look at the following research questions:

- what are considered to be important improvement targets for software groups,
- are these achieved, and
- are these achievements related to particular CMM-levels?

4.1.1 What are important imrovement targets

The result of these questions regarding SPI targets are depicted in Tables 3–5. Table 3 shows average scores for both ‘attention’ and ‘performance’.

The second column (‘mean attention given’; ‘all’) in this table shows that in absolute terms only ‘increase predictability’ (statistically significant for CMM level two and three groups) and ‘reduce defects’ (statistically significant for CMM level three groups) warrant more than average (above the midpoint) attention, while ‘increase productivity’ and ‘reduce lead time score approximately average (on the midpoint) attention. At CMM level three all four of these improvement targets are given a level of attention that is above the midpoint in absolute terms.

The three targets mentioned by Paulish and Carleton (1994) that were taken as the basis of this survey are all found in this list of four. Of the targets added by the SPI steering Committee only ‘increase productivity’ receives a higher level of attention. These results are also found in case studies on the effectiveness of software process improvement (e.g., Iversen and Ngwenyama, 2006). This could mean that other software groups have the same kind of focus in their SPI efforts. In the latter publication, besides ‘productivity’, ‘quality’ (defects) and ‘adherence to schedule and budget’ (in our terms ‘predictability’) also ‘customer satisfaction’ and ‘employee satisfaction’ are mentioned. ‘Employee satisfaction’ is clearly related to our improvement target ‘improve staff motivation’ which scored relatively low. In our research no improvement target ‘customer satisfaction’ has been defined. This opens up an interesting area for further research.

4.1.2 Attention vs. performance (is performance achieved)

When looking at performance, only the improvement targets ‘increase predictability’ and ‘reduce defects’ achieve a a somewhat higher score, and for CMM level two and three groups succeed in exceeding the midpoint score. In general the level of attention appears to be higher than the associated level of performance reported. This is confirmed in Table 4 which

Table 4 The relationship between attention and performance

Improvement target	Significance in difference	Correlation (significance)
Increase predictability	**0.00**	0.51 (**0.00**)
Reduce defects	**0.00**	0.68 (**0.00**)
Increase productivity	**0.00**	0.63 (**0.00**)
Reduce lead time	**0.00**	0.74 (**0.00**)
Improve cooperation	**0.03**	0.71 (**0.00**)
Improve staff motivation	0.32	0.68 (**0.00**)
Increase reusability	0.65	0.61 (**0.00**)

shows the result of a statistical test on the difference between attention and performance for each factor. The test used is a two-tailed *t*-test[2] on paired samples. A value in the column 'significance in difference' below 0.05 (indicated in bold in the table) suggests that there exists a difference between level of attention and the achieved perceived performance for that factor. Apart from 'improve staff motivation' and 'increase reusability', which incidentally are the lowest scoring factors overall for 'attention', all factors show a degree of attention that is significantly larger than the associated resulting performance.

However, also shown in Table 4 is the result of a correlation analysis for each factor between degree of attention and performance. The results presented are those of a Spearman correlation. In all cases a clearly statistically significant positive correlation exists between the degree of attention paid to a specific target and the performance achieved. We might conclude that aiming attention at a specific target generally has a positive impact on the associated performance. However the degree of performance is usually lower than the degree of attention. An issue that should be kept in mind here is that the data provide a snapshot in time. Proper analysis of the impact of 'attention' on 'performance' of course requires a longitudinal study. On the other hand, the Software Engineering Institute shows on its website that the average time required to move up a level is between one and a half and two years. This suggests that the average software group will have spent up to a year on its current level, justifying this look at the relationship between 'attention' and 'performance'.

4.1.3 Relation to CMM level

It is interesting to see if any relationship can be seen between the CMM-level reported and the score on improvement target attention or performance. Table 5 contains in the columns 'ANOVA' and '*t*-test' the significance results of statistical tests carried out to investigate this. The column 'ANOVA' gives the results of a standard analysis of variance between each individual improvement target and the CMM-level. The '*t*-test' columns for each improvement target give significance data resulting from a *t*-test for equality of means between groups on CMM-level one and two in the column '1 2', between groups on levels two and three in the column '2 3', and between groups on levels one and three in the column '1 3'. Data in bold indicate a relationship that is significant with 95% probability. Data in italic indicate a weaker relationship that is significant with 90% probability.

Looking at the factor 'reduce defects', the analysis of variance test indicates the existence of a relationship between CMM-level and degree of attention. A further look at the *t*-test

[2] Please note that in this paper parametric statistics are used for Likert-type scale data. Meltzoff (1998) provides a summary of the discussion in statistical literature that justifies this decision.

Table 5 Tests on differences for SPI targets across CMM-levels

	Attention				Performance			
	ANOVA	*t*-test			ANOVA	*t*-test		
Improvement target		1 2	2 3	1 3		1 2	2 3	1 3
Increase predictability	0.33	0.79	0.14	0.17	**0.00**	**0.02**	0.29	**0.00**
Reduce defects	**0.01**	0.48	**0.03**	**0.01**	0.10	*0.06*	0.88	0.10
Increase productivity	0.22	0.18	*0.08*	0.63	0.24	0.71	*0.09*	0.19
Reduce lead time	0.68	0.74	0.60	0.40	0.61	0.48	0.76	0.38
Improve cooperation	0.38	0.32	0.20	0.60	0.97	0.99	0.81	0.82
Improve staff motivation	0.37	0.18	0.70	0.41	0.81	0.54	0.75	0.79
Increase reusability	0.19	0.27	0.50	*0.06*	0.16	**0.05**	0.56	0.27

data shows that this is largely due to a significant increase of attention from level two to level three. No such relationship was found overall for the performance achieved for this issue. It can be seen that a weak case can be made for an increase in performance between CMM-level one and level two.

When discussing these results, the overall high level of attention for defect reduction is not very remarkable, given that CMM is a quality related program. The high increase of attention for this aspect on groups on level three and higher is interesting. This might be linked to CMM-level four Key Process Area (KPA) ‘software quality management’ and level five KPA ‘defect prevention’. Software groups at level three are supposed to concentrate at these next level KPA’s. All the more noticeable is that the performance achieved is less than satisfactory (below the midpoint). These data suggest that the focus on project management related to level two provides some ‘quick wins’ in the area of defect reduction. A more significant impact on defect reductions is apparently more difficult to achieve and would seem to require more targeted actions. Identifying these is an interesting area for further research.

ANOVA results show that the performance for the factor ‘increase predictability’ increases with the CMM-level. The *t*-tests show that this increase mainly results from the transition from level one to level two. No overall significant relationship between level of attention and CMM-level can be shown to exist, but a weak increase between level two and three is found in the *t*-test.

When looking at these results, such an increase in performance is not unexpected given the focus on project management activities (e.g. the KPA ‘software project planning’). Looking at the related level of attention for this subject, it can be seen that the overall level of attention is already relatively high, making a significant further increase unlikely. A possible explanation for the (not statistically significant) increase of the level of attention between level two and three is a tie in with the level four KPA ‘quantitative process management’ that level three groups are supposed to strive for next. However, from this increased level of attention as yet no significant effect on performance can be proven to exist from these data.

A significant relationship between level of attention/performance and CMM-level could not be established for any of the other factors. We can see however, that without any specific attention being paid to it, a significant increase of reusability performance can be observed between CMM-level one and two. Again a possible explanation is that this (in absolute terms limited) result is a ‘quick win’ resulting from the added degree of project control allowed in level two groups. No further increase was observed, hinting that more specific action in this area is required if more results is to be expected.

Another interesting change can be observed for the factor ‘productivity’ between CMM-levels two and three. A (weak) significant increase in attention here (after a puzzling but not significant decrease between level one and two) is matched by a (weak) significant increase in performance. This is remarkable, given that productivity is next to quality a main business objective for process improvement activities. A more significant result would have been appropriate. Given the similar result for defect reduction performance, this presents an overall picture that dictates a careful expectation management with respect to the effectiveness of software process improvement efforts.

4.2 SPI improvement drivers

Research questions to be looked at for improvement drivers are:

- what are considered to be important improvement drivers for software groups, and
- are they related to particular CMM-levels?

Based on literature research a list of improvement drivers has been identified, see Section 2. The results are presented in Table 6. The ANOVA and *t*-test results presented in this table are obtained in a similar way to that in Table 5. Again data in bold indicate strong statistical significance and those in italic indicate weak statistical significance.

Table 6 shows that most of the drivers can be considered to be of at least average importance in explaining SPI effectiveness. A first impression is that this set of drivers appears to be of value. Table 7 allows a closer look at this. In this table for each improvement driver the results of a one sample *t*-test of means against midpoint value ‘3’ is shown. Results are shown overall, but also separately within each CMM-level.

Looking at the overall results, it can be seen that three drivers differ significantly from the mid point value and for another a difference can be seen with a slightly weaker significance. The driver ‘commitment of engineering management’ scores highest in all columns. This result needs to be looked at with some reservation, given that the survey was answered

Table 6 Results for improvement drivers

	Mean				ANOVA	*t*-test		
Improvement driver	All	CMM1	CMM2	CMM3		1 2	2 3	1 3
Commitment of engineering management	4.0	3.7	4.0	4.7	*0.08*	0.47	*0.08*	**0.04**
Commitment of development staff	3.8	3.6	3.8	4.0	0.54	0.55	0.63	0.31
Sense of urgency	3.5	3.6	3.2	3.5	0.63	0.38	0.58	0.75
Availability of engineers time for SPI	3.4	3.6	3.4	3.1	0.63	0.66	0.62	0.36
Commitment of business management	3.3	3.3	3.0	3.5	0.69	0.55	0.39	0.75
Availability of qualified SPI resources	3.3	3.2	3.0	3.9	0.24	0.69	*0.06*	0.17
Clear/quantifiable improvement targets	3.2	2.8	3.2	3.7	*0.08*	0.33	0.15	**0.04**
Use of accepted framework such as CMM	3.1	2.5	3.6	3.9	**0.00**	**0.01**	0.54	**0.01**
Clear relation between SPI/business goals	3.1	2.7	3.8	3.3	**0.02**	**0.01**	0.22	0.17
Confidence in SPI results	3.0	2.9	2.8	3.6	0.12	0.85	**0.02**	*0.09*
Visibility of intermediate results	2.9	2.8	2.7	3.4	0.26	0.88	*0.07*	0.16
Sufficient investment in SPI training	2.8	2.6	2.5	3.7	**0.03**	0.84	**0.01**	**0.02**
Proper tooling to support the processes	2.8	2.6	3.1	2.7	0.57	0.32	0.41	0.86
Cooperation other engineering disciplines	2.6	2.3	3.0	2.6	0.43	0.19	0.54	0.58
Integration SPI in general improvement actions	2.5	1.9	3.2	3.0	**0.02**	**0.01**	0.79	**0.02**

Table 7 The difference between the improvement driver score and the midpoint value '3'

Improvement driver	Sig. (2-tailed)			
	All	CMM1	CMM2	CMM3
Commitment of engineering management	**0.00**	**0.03**	**0.00**	**0.00**
Commitment of development staff	**0.01**	**0.04**	**0.01**	**0.00**
Sense of urgency	**0.00**	**0.03**	0.55	**0.05**
Availability of engineers time for SPI	*0.07*	*0.08*	0.31	0.81
Commitment of business management	0.16	0.37	1.00	0.27
Availability of qualified SPI resources	0.19	0.53	1.00	**0.00**
Clear/quantifiable improvement targets	0.32	0.43	0.50	**0.05**
Use of accepted framework such as CMM	0.54	*0.09*	**0.01**	**0.04**
Clear relation between SPI/business goals	0.41	0.25	**0.00**	0.39
Confidence in SPI results	0.88	0.69	0.44	**0.02**
Visibility of intermediate results	0.77	0.46	0.34	0.10
Sufficient investment in SPI training	0.31	0.18	0.14	**0.01**
Proper tooling to support the processes	0.29	0.23	0.78	0.39
Cooperation other engineering disciplines	**0.04**	**0.02**	1.00	0.37
Integration SPI in general improvement actions	**0.02**	**0.00**	0.71	1.00

by engineering management. All the other drivers apart from the last two do not differ significantly from this midpoint value, which, given the symmetry of the scale, would indicate at least average impact. Finally, only two drivers score fairly low. If we look at the results per CMM-level, it can be seen that the low scoring drivers originate from level one groups. Both for level two and level three groups these drivers can be considered to score at least 'average'. In particular level three groups indicate they rate eight of these drivers significantly higher than average, while the rest does not differ significantly from the midpoint value. This confirms the first impression that this set drivers does seem to be valid. At least, this set seems to be valid for the investigated software groups within Philips.

An interesting question is how these results compare to those reported in literature. The most recent study in this area is by Niazi et al. (2005). Their work is based on a combination of literature research and interviews in practice. Although this work was not yet available when this survey was developed it will be used here as a reference since we can assume it contains state of the art information. We looked at the six main drivers identified by Niazi et al. and compare these to the drivers used in this survey. The results are shown in Table 8.

Such a comparison is not without difficulties. It is a face value comparison without any statistical backing. Also interpretation problems arise. We judged that 'staff and involvement' could be compared to 'commitment of development' and that 'SPI awareness' and 'sense of urgency' were also related concepts. Especially the latter comparison might not convince

Table 8 Comparing the drivers used to literature

Drivers from Niazi et al.	Drivers used in this research (with overall ranking)
Senior management commitment	Commitment of engineering management (1) Commitment of business management (5)
Staff involvement	Commitment of development staff (2)
Training and mentoring	Sufficient investment in SPI training (12)
SPI awareness	Sense of urgency (3)
Staff time and resources	Availability of engineers time for SPI (4)
Formal methodology	Use of an accepted framework such as CMM (8)

everyone. However, accepting this, there does seem to be a large degree of similarity between the results. Of the top six ranked in this survey only ‘availability of qualified SPI resources’ is missing. However, Niazi et.al. mention a more general driver ‘experienced staff’ that might match this. An interesting difference is observed for the difference in ranking of ‘training’. This issue is scored higher in the reference paper than in our results.

In our list of improvement drivers we decided, in collaboration with Philips, to make a distinction between two types of management commitment (see Section 3). Table 6 shows that the improvement driver ‘commitment of business management’ seems to behave differently compared to ‘commitment of engineering management’. A *t*-test on paired samples confirms this (significance 0.001). This distinction, which is also not found in Niazi et al., appears to be an interesting addition. An interesting research question for further research resulting from this, could be whether respondents are confident that SPI can be achieved by the engineering teams themselves (and without additional help from (external) business management). If true, this could be preliminary interpreted as an acceptance of CMM as an industry ‘best practice’ approach that contrasts with the ‘it is not my problem/its not my fault’ attitude that software was in at the start of the SPI activities in the early nineties (Schorsch, 1996).

Another interesting issue is the behavior of a number of drivers (7 to be exact) where we see a small ‘dip’ going from level one to two, followed by a large increase when moving to level three. The dip is small, and not supported by statistical evidence. The increase is supported by data from Table 7, where for four out of seven of these drivers in level three a significant difference from the midpoint value ‘3’ can be concluded while this is not possible for level two groups. Also, for four of these (‘availability of SPI resources’, ‘confidence in SPI results’, ‘sufficient investment in SPI training’, and ‘proper tooling to support the process’), this is supported by the results from the *t*-tests presented in Table 6. Remember that the highest scoring driver (commitment of engineering management) also showed a statistically significant increase to level three.

In contrast, of the remaining drivers only three show a significant relationship between the driver and the CMM according to the ANOVA model. For all of these, the *t*-test shows that this increase can be allocated to the change from CMM-level one to three. None show a steady increase over the CMM-levels.

All this suggests that moving across the CMM-levels is, seen from the point of view of improvement drivers, not a continuous process, but that somehow CMM-level three is different, requiring more attention. This might be explained by the complexity inherent to the higher CMM-levels, as opposed to e.g. the more basic project management skills required to obtain level two. This also is an interesting issue for further research.

4.3 Usage of metrics

As was previously discussed in Section 3, in this section the results of three research questions on metrics will be presented. These questions deal with:

- level of activity with regards to metrics;
- the quality of resulting data;
- the usage of resulting data.

4.3.1 Metrics, level of activity

The results regarding the level of metrics activities are presented in Tables 9 and 10. In Table 9 the over all questions are presented while Table 10 contains information on individual metrics

Table 9 Metrics: level of activity

	Score					*t*-test		
Question	All	CMM1	CMM2	CMM3	ANOVA	1 2	2 3	1 3
% Groups with formal metrics program	48.0%	26.3%	38.5%	100.0%	**0.00**	0.48	**0.00**	**0.00**
Number of metrics	7.37	6.16	6.31	11.50	**0.00**	0.92	**0.00**	**0.00**
% of projects with evaluation*	3.06	2.45	3.31	3.90	**0.00**	**0.05**	**0.05**	**0.00**
% of projects report quantitative data*	2.62	1.90	2.62	4.00	**0.00**	0.12	**0.01**	**0.00**

*Please note that these questions were scored on a 4-point scale: (1 = 0–25%; 2 = 25–50%; 3 = 50–75%; 4 = 75–100%).

Table 10 Actual usage of metrics by software groups that perform on different CMM-levels

	% of groups using the metric					*t*-test		
Metrics	All	CMM1	CMM2	CMM345	ANOVA	1 2	2 3	1 3
Actual effort spending	82.6%	57.9%	100.0%	100.0%	**0.00**	**0.00**		
Size	69.6%	63.2%	53.8%	90.0%	0.18	0.61	*0.07*	0.13
Lead time	65.2%	57.9%	69.2%	90.0%	0.22	0.53	0.25	*0.08*
Schedule metrics	41.3%	21.1%	38.5%	80.0%	**0.01**	0.30	**0.05**	**0.00**
Staff competence level	39.1%	31.6%	38.5%	60.0%	0.35	0.70	0.33	0.15
Staff attrition	37.8%	31.6%	53.8%	33.3%	0.43	0.22	0.36	0.93
Test coverage % requirements related	37.0%	42.1%	15.4%	50.0%	0.18	*0.10*	*0.08*	0.70
Fault density pre-release	32.6%	31.6%	7.7%	70.0%	**0.01**	*0.08*	**0.00**	*0.06*
Fault severity distribution	30.4%	31.6%	7.7%	50.0%	*0.08*	*0.08*	**0.02**	0.35
Fault density post-release	28.3%	21.1%	15.4%	60.0%	**0.04**	0.70	**0.03**	**0.04**
Cumulative failure profile	28.3%	31.6%	7.7%	50.0%	*0.08*	*0.08*	**0.02**	0.35
Test coverage % code related	21.7%	26.3%	15.4%	30.0%	0.69	0.48	0.42	0.84
Re-use metrics	17.8%	15.8%	23.1%	22.2%	0.86	0.62	0.97	0.69
Mean time to failure	17.4%	15.8%	0.0%	30.0%	0.13	*0.08*	**0.04**	0.39
Requirements metrics	15.2%	10.5%	7.7%	30.0%	0.27	0.80	0.18	0.20
Time to spec	13.3%	10.5%	23.1%	10.0%	0.57	0.35	0.44	0.97
Cyclomatic complexity	0.0%	0.0%	0.0%	0.0%				

usage. In the survey 17 metrics were included by name. Data on these are included in Table 10. A large number of widely different additional metrics was added by the respondents. Variety was such that these are not included in Table 10; however, they were added to the total count reported in Table 9. Please note that in Tables 9 and 10 the ANOVA and *t*-test columns have a meaning similar to those presented before in Table 5.

In Table 9 we see that on the higher CMM-level three the average number of metrics in use almost doubles to 11.5. This table also shows that 100% of CMM-level three groups have a formal measurement program in place (as opposed to 39% for level 2 and 26% for level one groups). These observations are in conformance with (Jalote, 2002) in that it is stated that '. . .high maturity organizations are expected to use metrics heavily. . .'. Overall in Table 9 all four questions show a significant relationship with the CMM-level, and in all cases the increase towards level three plays a significant part. Only the more general "% of projects with evaluation" also shows an earlier improvement towards level two, possibly related to the level two KPA's 'software project tracking and oversight' and 'software quality assurance.'

Table 10 shows that only three metrics, 'effort', 'size' and 'lead time,' have a score higher than 50% in the first column (all software groups). In Jalote (2002), the organizations studied on the use of metrics, also collected data on these three metrics. That research, focused at high maturity organizations, also showed a preference for the collection of data on 'defects'. Default related metrics also scored quite high in the level three groups included in our research, where all four fault/failure associated metrics were used by at least 50% of the software groups. Apart from these there seems to be little consensus in the software groups that we studied, about the specific type of metrics that should be used. Moving towards CMM-level 3 the increase in the number of metrics used that could be seen in Table 9 is mirrored here. Eight metrics show a significant increase towards level three (six strong and two weak), while ten metrics score at 50% or more. The impression arising from studying Table 9, that the level of metrics activities significantly increases at level three groups, is strengthened by these results.

An interesting phenomenon can be observed if we look at the changes between level one and level two software groups. Although the average number of metrics used does not change from level one to level two, in no less than nine cases we see that usage of a specific metric actually decreases. In five of these, this decrease is statistically (albeit weak) significant. A possible explanation might be, that in the move from level one towards level three software groups focus their attention on a more limited number of issues, resulting in a more focused use of metrics. Our research shows that in level three software groups the number of metrics used increases strongly, and that quite different metrics are used in the different software groups. This is remarkable and not in-line with (Paulk, 1999) in that it is concluded that there exist many similarities in particular metrics used in high maturity organizations. This also warrants further research.

4.3.2 The quality of the resulting data

The results for the three questions related to this issue are presented in Table 11. In this table the ANOVA and *t*-test columns have a meaning similar to those presented before in Table 5.

Table 11 shows that mainly at the higher CMM-levels a quite high percentage of software groups are convinced of the quality of the data collected. For all questions a significant relationship between CMM-level and data quality can be observed (weak significance for 'fit metrics to business objective'). However, for all questions this increase focuses on the step from level two to level three. No significant differences between level one and level three groups can be observed. This result is in line with what was observed in the previous section.

Table 11 Metrics: data quality

	Score					*t*-test		
Question	All	CMM1	CMM2	CMM3	ANOVA	1 2	2 3	1 3
Fit metrics to business objectives	2.80	2.44	2.62	3.75	*0.06*	0.71	*0.08*	**0.03**
Are data validated	31.8%	23.5%	23.1%	80.0%	**0.00**	0.98	**0.01**	**0.00**
Data are reliable at holding level	34.1%	23.5%	27.3%	80.0%	**0.01**	0.83	**0.01**	**0.00**

Table 12 Usage of data

Question	Score				ANOVA	*t*-test		
	All	CMM1	CMM2	CMM3		1 2	2 3	1 3
Degree of importance of metrics	3.00	2.42	2.77	4.30	**0.00**	0.35	**0.00**	**0.00**
Metrics guide SW operation	63.6%	52.9%	61.5%	100.0%	**0.03**	0.65	**0.02**	**0.00**

4.3.3 The usage of resulting data

The results for the questions related to this issue are presented in Table 12. Again, the ANOVA and *t*-test columns have a meaning similar to those presented before in Table 5.

Table 12 shows that again the pattern observed for level of metrics activity and data quality holds here. No significant changes occur from level one to level two, while a marked and statistically significant increase in usage can be seen between levels two and three, with even a full 100% of groups at level three claiming to use metrics data in guiding software development. Both questions show similar results. Furthermore, these results are in line with was observed in the previous two subsections. Apparently increased metrics activities, increased data quality and increased data usage go hand in hand.

These data strongly suggest that in a CMM program collection and usage of metrics is an activity that is closely linked to level three. This is possibly connected to the level four KPA 'quantitative process management' that level three software groups should take as a next target. These data also seem to confirm critique of the way measurement is handled in the CMM model, especially on the lower levels. Given that the CMMI contains a more explicit treatment of measurement, it will be interesting to see if CMMI organizations will be able to show an earlier adoption of sound metrics usage practices.

5 Conclusions

The survey results provide interesting findings on SPI in practice and complements and extends previous research from several perspectives. Of course, the research resulted also in additional and new questions for further research to be done. In the following we summarize the main conclusions regarding respectively improvement targets, improvement drivers and metrics usage.

Regarding *improvement targets* the questions were:

- what are considered to be important improvement targets for software groups,
- are these achieved, and
- are these achievements related to particular CMM-levels?

With respect to the identified improvement targets, from the point of view of the degree of attention spent, only 'increase predictability' and 'reduce defects' warrant more than average attention, joined for Capability Maturity Model (CMM) level three groups by 'increase productivity' and 'reduce lead time'. The first two of these improvement targets are also the ones that score highest in perceived actual performance.

Regarding the performance achieved, for all improvement targets a clearly statistically significant positive correlation exists between attention paid and performance achieved. Although the degree of performance achieved is usually lower than the degree of attention paid, we might conclude that aiming attention at a specific improvement target generally has a positive impact on the associated performance. Regarding the fact that the impact on performance achieved is relatively low, it could be beneficial for software groups to carefully carry out some form of expectation management regarding the effectiveness of their software process improvement activities.

The high increase of attention for ‘reduce defects’ of groups on level three and higher is interesting. We think that this might be linked to CMM-level four Key Process Area (KPA) ‘software quality management’ and level five KPA ‘defect prevention’. Organizations at level three are supposed to concentrate at these next level KPA’s. Noticeable is that the performance achieved is less than satisfactory. A more significant impact on defect reduction is apparently more difficult to achieve and would seem to require more targeted actions (in the context of paying attention). Identifying these additional targeted actions is an interesting area for further research

Regarding *improvement drivers* the questions were:

- what are considered to be important improvement drivers for software groups, and
- are they related to particular CMM-levels?

Looking at the overall results, the set of drivers used in our research does seem to be valid, at least for Philips. Three improvement drivers differ significantly from the average score, and for three others a difference can be seen with a slightly weaker significance. An interesting question is how these results compare to those reported in literature. We looked at the six main drivers identified in a related reference and compared these to the drivers used in our survey. We concluded a large degree of similarity between the results.

In our list of improvement drivers, we made a distinction between two ‘management commitment’ drivers. Our research shows that the improvement driver ‘commitment of business management’ seems to behave differently compared to ‘commitment of engineering management’. Although both commitment drivers score high, an interesting issue for further research could be whether respondents are confident that SPI can be achieved by the engineering teams themselves (and without additional help from business management).

Further, statistical analysis shows that moving across the CMM-levels is, seen from the point of view of improvement drivers, not a continuous process. Somehow CMM-level three seems to be different and seems to require more attention. This might be explained by the complexity inherent to the higher CMM-levels, as opposed to e.g. the more basic project management skills required to obtain level two. This is also an interesting issue for further research.

Regarding metrics usage the questions deal respectively with:

- level of activity with regards to metrics;
- the quality of resulting data;
- the usage of resulting data.

We can state that on CMM-level three the average number of metrics in use almost doubles. Of CMM-level three groups, 100% have a formal measurement program in place. There is consensus in all the software groups about three specific metrics, respectively ‘actual effort spending’, ‘size’ and ‘lead time’, while level three groups also seem to focus on defect metrics. Apart from this there seems to be little consensus about the specific type of metrics that should be used.

Mainly at the higher CMM-levels a quite high percentage of software groups are convinced of the quality of the data collected. From the research a significant relationship between CMM-level and data quality can be noticed. This relationship counts in particular for the step from CMM level two to CMM level three.

Further it became clear that in an CMM environment, collection and usage of metrics is an activity that is closely linked to CMM level three. To what extent this is caused by the fact that level three groups have as next target level four KPA's such as 'quantitative process management' is also an interesting subject for further research.

References

Beecham, S., Hall, T., Rainer, A. 2003. Software process improvement problems in 12 software companies: An empirical analysis. Empirical Software Engineering **8**(1):7–42.

Cummins, R.A., Gullone, E. 2000. Why we should not use 5-point Likert scales: The case for subjective quality of life measurement. Proceedings, Second International Conference on Quality of Life in Cities, Singapore, National University of Singapore, pp. 74–93.

Dybå, T. 2003. Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context. ESEC/FSE, pp. 1–10.

El-Emam, K., Goldenson, D., McCurley, J., Herbsleb, J. 2001. Modeling the likelihood of software process improvement: An exploratory study. Empirical Software Engineering **6**(3):207–229.

Iversen, J., Ngwenyama, O. 2006. Problems in measuring the effectiveness in software process improvement: A longitudinal study of organisational change at Danske data. International Journal of Information Management **26**:30–43.

Jalote, P. 2002. Use of metrics in high maturity organisations. Software Quality Professional **4**(2).

Meltzoff, J. 1998. Critical thinking about research. American Psychological Association, APA Service Center, Washington, DC, USA, p. 300.

Niazi, M., Wilson D., Zowghi, D. 2005. A maturity model for the implementation of software process improvement: An empirical study. The Journal of Systems and Software **74**(2):155–172.

Paulish, D.J., Carleton, A.D. 1994. Case studies of software-process-improvement measurement. IEEE Computer **27**(9):50–57.

Paulk, M. 1999. Practices of high maturity organisations. SEPG Conference Proceedings, Atlanta, Georgia.

Rainer, A., Hall, T. 2002. Key success factors for implementing software process improvement: A maturity-based analysis. Journal of Systems and Software **62**(2):71–84.

Rainer, A., Hall, T. 2003. A quantitative and qualitative analysis of factors affecting software processes. Journal of Systems and Software **66**(1):7–21.

Schorsch, T. 1996. The capability im-maturity model. Crosstalk 9(11):27–30.

SEI: Software Engineering Institute, Carnegie Mellon University. 1995, June. The Capability Maturity Model: Guidelines for Improving the Software Process. Addison Wesley Professional, Boston, USA, p. 464.

Skipper, R., Hyman, M.R. 1993. On measuring ethical judgments. Journal of Business Ethics **12**:535–545.

Stelzer, D., Mellis, W. 1998. Success factors of organizational change in software process improvement. Journal for Software Process—Improvement and Practice **4**:227–250.

Trienekens, J.J.M. 2004. Towards a model for managing success factors in software process improvement. Proceedings of the 1st International Workshop on Software Audit and Metrics (SAM) during ICEIS 2004, Porto. Portugal, pp. 12–21.

Jos J.M. Trienekens (1952) is an Associate Professor at TU Eindhoven (University of Technology—Eindhoven) in the area of ICT systems development. His current research interests include software process improvement, software quality and software metrics. He is responsible for a research program on ICT driven business performance and is an associate member of the research school BETA at TUE that focuses at operations management issues. Jos Trienekens published over the last ten years various books, papers in international journals and conference proceedings. He joined several international conferences as member of the organisation committees and PC's. He is also an experienced project partner in European projects.

Rob J. Kusters (1957) obtained his master degree in econometrics at the Catholic University of Brabant in 1982 and his Ph.D. in operations management at Eindhoven University of Technology in 1988. He is professor of 'ICT and Business Processes' at the Dutch Open University in Heerlen where he is responsible for the master program 'Business process and ICT'. He is also an associate professor of 'IT Enabled Business Process Redesign' at Eindhoven University of Technology where he is an associate member of the research school BETA which focuses at operations management issues. He published over 70 papers in international journals and conference proceedings and co-authored five books. Research interests include enterprise modelling, software quality and software management.

Michiel van Genuchten holds a Masters (1987) and a Ph.D. (1991) from the Eindhoven University of Technology. He has worked in industry since 1987, among others at Philips Electronics and GroupSupport, a software company he founded. His focus of attention is software as a technology, software as a business and group support systems. Results of his research work have been published in journals such as IEEE Software, Journal of MIS, IEEE Transactions on Software Engineering and IEEE Transactions on Professional Communications. Michiel van Genuchten has been appointed as a professor of software management at Eindhoven University of Technology in 2002.

Hans Aerts is responsible for Product Creation Process (PCP) support in Philips Semiconductors, co-responsible for maintaining the PCP definitions, and supporting the process deployment in the various

Business Lines, disseminating best practices, and executing assessments. He joined Philips in 1981 and has experience in developing embedded software for telecommunications systems, test and measurement equipment, car navigation systems, and consumer electronics products. Since 1988, he was involved in starting the software process improvement (SPI) activities throughout Philips and introduced the CMM and software process assessments. Hans received a MSc in mathematics and computer science from the University of Technology, Eindhoven.