ELSEVIER

The Journal of Systems and Software 66 (2003) 23–33

The Journal of Systems and Software

www.elsevier.com/locate/jss

De-motivators for software process improvement: an analysis of practitioners' views

Nathan Baddoo *, Tracy Hall

Department of Computer Science, University of Hertfordshire, Hatfield Campus College Lane, Hatfield, Hertfordshire AL10 9AB, UK

Received 9 August 2001; received in revised form 31 October 2001; accepted 20 February 2002

Abstract

We present a study of software practitioners' de-motivators for software process improvement (SPI). The aim of this study is to understand the nature of the issues that de-motivate software practitioners for SPI, so that SPI managers can better manage these de-motivators. This study compares what the SPI literature reports as the factors that hinder SPI success with software practitioners' perception of the factors that de-motivate them. Focus groups are used to elicit the perceptions of over 200 software practitioners. Our findings show that software practitioners confirm what the literature reports as the major issues that de-motivate them for SPI. These issues are related to resistance to change, lack of evidence, imposed SPI initiatives, resource constraints and commercial pressures. Our findings also show that there are differences in de-motivators for SPI across staff groups and that these differences are related to the role that software practitioners have in software development generally. We offer these findings as insight to aid SPI managers to design more targeted SPI strategies.

Keywords: De-motivators; Software process improvement; Practitioners

1. Introduction

In this paper we present empirical findings showing what de-motivates software practitioners from supporting Software Process Improvement (SPI). Our study uses data from focus group discussions in 13 UK companies encompassing almost 200 software practitioners. This paper provides insight to managers of SPI programmes by identifying issues that practitioners find de-motivating about SPI.

Despite widespread adoption of SPI in companies, and the publication of case studies, dissertations and survey results on SPI, there is insufficient quantitative evidence of how products have been improved by process change (Glass, 1999; Gray and Smith, 1998; Kuilboer and Ashrafi, 2000). This lack of evidence suggests that SPI is not delivering the benefits promised. It also suggests that there have been too few independent evaluations of the impact of SPI on software quality. We suggest that SPI may not be delivering the benefits promised because insufficient attention has been paid to the human aspects of implementing SPI. One such human aspect is why practitioners are de-motivated to get involved in SPI. We present these findings as part of a series of examinations on the human aspects of SPI in companies. In previous studies, we have reported on practitioner motivators for SPI (Baddoo and Hall, 2002).

Our previous findings suggest that different groups of practitioners have different perceptions and experiences of SPI (Hall and Wilson, 1997) and that these differences make it difficult to develop a coherent strategy for effectively implementing SPI. In this study we present the similarities and differences of de-motivators for three groups of practitioners: developers, project managers and senior managers. Such analysis provides SPI managers with insight to develop targeted strategies for SPI implementation.

In this paper we report that even though the majority of de-motivators are common to all three practitioner groups, unique de-motivators are related to the particular roles that practitioners play in software development.

* Corresponding author. Tel.: +44-1707-286-492; fax: +44-1707-284-303.

E-mail addresses: n.baddoo@herts.ac.uk (N. Baddoo), t.hall@herts.ac.uk (T. Hall).

0164-1212/03/$ - see front matter © 2002 Elsevier Science Inc. All rights reserved.
doi:10.1016/S0164-1212(02)00060-2

So that developers tend to cite day-to-day de-motivators, whereas project managers cite tactical, medium-term de-motivators. An effective SPI strategy must address the commonality in de-motivators across practitioner groups but also reflect uniqueness within groups.

In Section 2 of the paper we provide some background to de-motivators of successful SPI implementation. In Section 3 we describe our research methodology and discuss our data collection and analysis techniques. In Section 4 we present the results of our analysis. In Section 5 we discuss the findings of this research. Section 6 summarises this paper.

2. Background to SPI de-motivators

In order to explore software practitioners' de-motivators for SPI more, we have analysed the SPI literature to ascertain the most critical barriers to SPI. The following issues are reported as the most critical barriers to SPI success. We discuss these issues in this section and follow them up when we analyse the results of our study in later sections.

2.1. Resistance, inertia and negative experience

One of the biggest obstacles to introducing any new practice is the unwillingness of practitioners who actually use the practices to take them up. This problem often arises when practitioners perceive no incentive for giving up practices with which they are accustomed and feel comfortable with. Such a reaction is not necessarily a pro-active response to the new practices being introduced, but rather a need to continue with current and established practices. It reflects the old adage: "why fix what's not broken". Overcoming such inertia is reported as critical to gaining support for new practices (Humphrey, 1998). Humphrey observes that even intelligent practitioners will not engage in practices that "logic, experience and even hard evidence suggests that they should" (Humphrey, 1998). Humphrey offers several reasons for this including (Humphrey, 1998):

- Once practitioners learn to develop programs that work they also establish some basic personal practices.
- These personal practices become ingrained the more practitioners use them.
- Previous bad experience of new tools and techniques, does not make practitioners think that new practices will improve their output.

This suggests that practitioners display inertia because they are unwilling to give up learned habits and have previous negative experience of new practices. However, it is possible that practitioners also can actively resist the introduction of new practices due to other factors. For example, practitioners may resist new practices that they perceive as a threat to their autonomy.

In this study we investigate accounts of resistance, inertia, obstacles and negative experiences in practitioner perceptions of SPI. We focus on the following research question:

RQ1: Are software practitioners resistant to SPI?

2.2. Lack of evidence of benefits

Studies investigating critical success factors of SPI indicate that providing practitioners with evidence of the benefits of SPI is a good motivator for establishing buy-in to SPI (Humphrey, 1998). Indeed our own study of the motivators of SPI indicates that practitioners want some evidence of its direct benefits to them in order to embark upon SPI (Baddoo and Hall, 2002; Rainer et al., 2002).

On the other hand, software practitioners may display intransigence towards new and better practices which will not necessarily be improved by providing them with evidence (Humphrey, 1998). Humphrey argues that practitioners will not use new practices even when there is clear evidence that these practices or methods help.

These conflicting views on the influence of evidence on practitioner motivators for SPI warrant further investigation. In this study we investigate the extent to which lack of evidence of direct SPI benefits constitutes a practitioner de-motivator for SPI. To do this, we focus on the following research question:

RQ2: Does a lack of evidence de-motivate software practitioners from supporting SPI?

2.3. Imposition

Work done on barriers to SPI identify corporate level SPI initiatives as a barrier to successful SPI implementation (Hantos and Gisbert, 2000). This is because software practitioners resist initiatives that they perceive as imposed upon them. The nature of SPI programmes initiated at a corporate level means that they are often not consultative and have not secured practitioner buy-in.

In this study, we explore software practitioners' perception of imposed practices. We find out the extent to which this approach to implementing SPI de-motivates the SPI effort. We research the following research question:

RQ3: Do imposed SPI initiatives de-motivate practitioners?

2.4. Resource constraints

Most studies on critical success factors of SPI recognise that dedicated resources are critical to success (El Emam et al., 1999). Goldenson and Herbsleb (1995) cite resources as one of the two most important factors to SPI success. Pitterman refers to the importance of dedicated resources to Telcordia's SPI success (Pitterman, 2000). Indeed our own previous findings suggest that software practitioners across all staff groups are highly motivated by dedicated people, time, experience and tools to SPI programmes (Baddoo and Hall, 2002).

However, little work has been done to examine the impact of the absence of all or a combination of these resource factors on the motivations of software practitioners. In this study, we explore the lack of such resource factors on the motivations of practitioners for SPI. We focus on the research question:

> RQ4: Does a lack of resources de-motivate software practitioners from supporting SPI?

There is a lack of research to indicate what are sufficient resources. Therefore in this study, we suggest that if software practitioners are content with their resource levels, then it indicates that they have got sufficient resources.

2.5. Commercial pressures

Anecdotal evidence suggests that pressure from customers often serves as a barrier to SPI as companies are regularly faced with commercial pressures to meet customer demands. Such pressures are brought on by the need to gain favourable market positions. SPI can often become a casualty of these commercial pressures.

> "When the customer is waving his[sic] cheque book at you, process issues have to go" (a developer in our study)

In this study, we explore how software practitioners perceive these commercial constraints on their motivations for SPI. We research the question:

> RQ5: Do commercial pressures de-motivate software practitioners from supporting SPI?

2.6. Staff groups

We are also interested in how de-motivators differ with respect to software practitioner groups. In previous work we have collected data from separate groups of software practitioners in order to ascertain a staff level perspective of issues (Hall and Wilson, 1997). In a study of 39 staff groups from some 59 companies, Kitson and Masters (1993) separated their practitioner groups into three hierarchical staff groups. They report that because they were collecting data from managers and practitioners who were confronting real issues on a daily basis, they had high confidence in the accuracy and validity of the data. Also, in other studies done on software practitioners' motivators, practitioners have been separated into staff groups in order to enable a staff group perspective of the issues being examined (Couger and O'Callaghan, 1994; Khalil et al., 1997). These group perspectives make it possible to detect differences that may exist in practitioners' perceptions within companies.

In the context of this study, we suggest that understanding the differences in de-motivators across practitioner groups can help to develop better SPI implementation strategies. As a result, we focus on the following research question:

> RQ6: Do developers, project managers and senior managers have different de-motivators for SPI?

3. Methodology

3.1. Focus groups

We used focus groups as our main approach to collecting data. Focus groups are a well documented technique in the social sciences (Morgan, 1997). They involve assembling small groups of peers to discuss particular topics. Discussion is largely free-flowing, but is directed by a researcher allowing soft issues to be explored. Indeed focus groups have been described as "a way to better understand how people feel and think about an issue" (Krueger and Casey, 2000). Also, Morgan and Krueger say that "the comparisons participants make among each other's experiences and opinions are a valuable source of insights into complex behaviours and motivations" (Morgan and Krueger, 1993). Focus groups are, therefore, an ideal vehicle for exploring motivation for SPI.

Focus groups also elicit data that allows a better understanding of the differences between groups of people (Krueger and Casey, 2000). This makes them ideal in our study where we are interested in exploring the different motivations of practitioner groups.

3.2. Implementing focus groups

From September 1999 to March 2000 we visited 13 software companies and conducted 49 focus groups.

Appendix A provides an overview of the companies. Focus groups comprised between four and six members and altogether 200 software practitioners took part in this study. Participating companies were selected from a larger sample of companies who responded to a detailed questionnaire giving broad information about their software development activities and company demographics. They were chosen to provide our research project with a cross-section of company maturity levels and software applications.

Each focus group lasted approximately 90 min. We conducted the following focus groups: 21 developer focus groups, 16 project manager focus groups, 12 senior manager focus groups.

In order to investigate de-motivators for SPI we asked focus groups to discuss:

- What are the obstacles to SPI in your company?
- What would stop SPI from happening in this company?

Practitioner responses to the questions were audio recorded and the audio recordings were transcribed.

3.3. Qualitative data analysis

We used the broad principles of content analysis (Krippendorf, 1980) to analyse each focus group discussion. We used content analysis principles to develop categories for de-motivators by placing emphasis on the meaning, the interpretation and the intentions of the data, as offered by practitioners (Krippendorf, 1980).

Below is a fuller explanation of the analysis procedure.

Stage 1: A manual protocol analysis is performed on responses to focus group questions on de-motivators for SPI. All de-motivators cited by each practitioner group are identified. A list of all the de-motivators for each group was then constructed. See Appendix B for a definition of all de-motivators. We conduct inter-rater reliability to increase confidence in this identification process:

- To check for researcher bias, a list of all de-motivators categorised is given to a colleague who is not familiar with the issues being researched. Four transcripts were selected at random and the colleague asked to record the issues that appear in the transcripts under the categories offered.
- If there are great disagreements in the recording of the categories, the two groups of researcher will discuss the categorising system and make adjustments as fit. In this instance there were no great disagreements.

Stage 2: We constructed a matrix mapping de-motivators cited to each of the three practitioner groups.

Stage 3: We weighted each de-motivator for each practitioner group by frequency. These frequencies were converted into percentages. So, for example, if out of a total of 21 developer groups *de-motivator x* was identified in six of them, *de-motivator x* is assigned an occurrence weighting of 28.6%.

4. Findings

Table 1 provides a summary of all de-motivators identified in this study. It shows how de-motivators occurred across all three practitioner groups. Frequencies show the number of occurrence across all 49 groups and the percentages reflect the proportion of groups the de-motivators occurred in.

Table 1 shows that time and resource issues, inertia and difficulties with obtaining practitioner support and buy-in for SPI are perceived as the major de-motivators for SPI by practitioners in this study.

Table 1
Summary of de-motivators in study across all groups

De-motivators	Occurrence in focus groups ($n = 49$)	
	Freq.	(%)
Time pressure/constraints	27	55
Inertia	19	39
Lack of resources	13	27
Commercial pressures	11	22
Lack of overall support	9	18
Budget constraints	8	16
Cumbersome processes	8	16
Lack of evidence of direct benefits	7	14
Negative/bad experience	7	14
Inadequate communication	5	10
Imposition	4	8
Inexperienced staff	4	8
Lack of mgt direction/commitment	4	8
Lack of SPI management skills	4	8
Low process priority	4	8
Personality clashes	3	6
Workload	3	6
Lack of feedback	2	4
Lack of standards	2	4
Staff turnover	2	4
Customers	1	2
Fire fighting	1	2
Inadequate metrics	1	2
Irrelevant objectives/deliverables	1	2
Isolated best practice	1	2
Large-scale programmes	1	2
Organisational changes	1	2
PM's lack of technical knowledge	1	2
Reduced creativity	1	2

4.1. De-motivators within practitioner groups

4.1.1. Developers' de-motivators

Table 2 shows that 62% of developers find time pressures de-motivating to SPI. When we combine this with 14% of developers who cited workload and 24% citing budget constraints as de-motivators, time and resources for SPI seem very hard to find. Nearly half of developers also think that inertia caused by practitioners' resistance to change is an obstacle to SPI. A quarter of developers also cited cumbersome processes as de-motivators. These are the only de-motivators cited by more than a quarter of developer groups.

Eight of the 19 de-motivators cited by developers were cited by single groups. A few of these seemed obscure, for example, reduced creativity. Such de-motivators could be company specific. However, many are related to de-motivators cited by other groups. For example, only one group of developers indicates that customers are de-motivating to SPI. At the same time, this issue of customers is linked to some of the views expressed about commercial pressures and external pressures on SPI. Also, other de-motivators like inadequate communication and negative experience are widely shared by other practitioner groups even though they are cited in only one developer session.

The following are additional comments made by developers about some de-motivators:

Table 2
De-motivators cited by developers

De-motivators	Occurrence in focus groups ($n = 21$)	
	Freq.	(%)
Time pressure/constraints	13	62
Inertia	9	43
Budget constraints	5	24
Cumbersome processes	5	24
Commercial pressures	4	19
Lack of mgt direction/commitment	4	19
Imposition	3	14
Workload	3	14
Lack of standards	2	10
Lack of feedback	2	10
Personality clashes	2	10
Inadequate communication	1	5
Customers	1	5
Inexperienced staff	1	5
Isolated best practice	1	5
Lack of overall support	1	5
Negative/bad experience	1	5
PM's lack of technical knowledge	1	5
Reduced creativity	1	5

Commercial pressures:

> If you want to spend a little bit of time fixing up one of your areas and there's a customer offering to write a cheque for a new piece of development, the piece of development would be done in preference to the niceties of something else.

Inertia:

> You are always going to come across people who will say, "oh I have done this for the last twenty years, I am going to carry on doing it". And I think when it comes to CMM, they are the ones that are causing us problems.

Negative experience:

> I think the biggest problem is getting people that have been de-motivated in the past, re-motivated. They will say "we have done this in the past, nothing happened. I gave you those volumes, nothing happened, we are still doing the same old thing, why should I bother? This has to be the biggest challenge I think.

4.1.2. Project managers' de-motivators

Table 3 shows that time pressures and lack of evidence of direct benefits were cited by around 40% of project managers. Other de-motivators were cited by over 25% of project managers, for example commercial pressures, inertia, lack of evidence and low process priority.

Table 3
De-motivators cited by project managers

De-motivators	Occurrence in focus groups ($n = 16$)	
	Freq.	(%)
Time pressures/constraints	7	44
Lack of evidence of direct benefits	6	38
Lack of resources	5	31
Commercial pressures	4	25
Inertia	4	25
Low process priority	4	25
Cumbersome processes	2	13
Inadequate communication	2	13
Lack of overall support	2	13
Negative/bad experience	2	13
Staff turnover	2	13
Fire fighting	1	6
Imposition	1	6
Inadequate metrics	1	6
Irrelevant objectives/deliverables	1	6
Large-scale programmes	1	6

Project managers cite more de-motivators than other groups. Project managers are often perceived by developers and senior managers as a hurdle to SPI implementation in companies. The fact that they cite more de-motivators may explain this perception that they are a barrier to SPI.

Here are some additional comments offered by project managers to describe their de-motivators for SPI:

Fire fighting:

> When we are 'fire fighting' we can't step back and think about things that can be improved.

Low process priority:

> The engineers get very disillusioned when they start working on a process activity and they have to stop and finish the project work they are doing because they know it's a low priority. Process work is not recognised as much as project work. If they don't manage to get process work done it's very much don't worry, try and do it next year.

Time pressures:

> The biggest problem that I have is that I ask people to change the way they work when they are already working flat out in order to deliver a product. The biggest issue is that people do not have time to understand a change and the benefit that change will give them so they are resistant to change.
> Officially we've been told that you can build in time for PI but as soon as you hit any deadline the first thing to go is that.

4.1.3. Senior managers' de-motivators

Table 4 shows that, overall, there is strong agreement between senior managers about their de-motivators, with nearly two out of three de-motivators cited by over 25% of senior managers. Table 4 shows that 67% of senior managers cited lack of resources as a de-motivator and 58% cited time pressures. Inertia and lack of overall support are cited by 50% of senior managers. Furthermore 33% of senior managers cited lack of SPI management skills as a de-motivator with 25% citing inexperienced staff. This indicates senior managers have a particular concern about the shortage of SPI skills.

This is how senior managers described some of their de-motivators:

Bad experience:

> We have had a situation in the past where we have put a lot of effort into something but haven't got the end result and people remember that. They always remember the things that didn't go well, not the things that did.

Lack of resources:

> The whole areas of giving the reward, the incentive and the bandwidth to engineers to go and do it. The best engineers at PI also happen to be the best engineers at getting the product out of the door, so it's very hard to free up enough to concentrate on the process stuff.

Table 4
De-motivators cited by senior managers

De-motivators	Occurrence in focus groups ($n = 12$)	
	Freq.	(%)
Lack of resources	8	67
Time pressures/constraints	7	58
Inertia	6	50
Lack of overall support	6	50
Bad/negative experience	4	33
Lack of SPI management skills	4	33
Budget constraints	3	25
Commercial pressures	3	25
Inexperienced staff	3	25
Inadequate communication	2	17
Cumbersome processes	1	8
Lack of evidence of direct benefits	1	8
Organisational changes	1	8
Personality clashes	1	8

4.2. De-motivators across practitioner groups

4.2.1. Common de-motivators

Table 5 reproduces de-motivators cited by more than one practitioner group. The percentages represent de-motivators within particular groups.

Half of the de-motivators are common to more than one practitioner group. In fact Table 5 shows that over half of these common motivators are common to all the three practitioner groups. This provides an indication of the level of agreement that practitioners across all staff groups have about what de-motivates them from practising SPI.

We also suggest that if all staff groups cite the same de-motivator then they are actually identifying important de-motivators. It also indicates a good understanding of the problems in SPI. These common de-motivators probably need to be taken very seriously.

4.2.2. Different motivators across practitioner groups

Table 6 shows de-motivators that were cited within only one practitioner group.

Table 5
Common de-motivators across practitioner groups

De-motivators	Cited (%)		
	Dev	PM	SM
Commercial pressures	19	25	25
Cumbersome processes	24	13	8
Inadequate communication	5	13	17
Inertia	43	25	50
Lack of overall support	5	13	50
Negative/bad experience	5	13	33
Time pressure/constraints	62	44	58
Lack of resources	0	31	67
Lack of evidence of direct benefits	0	38	8
Budget constraints	24	0	25
Inexperienced staff	5	0	25
Personality clashes	10	0	8
Imposition	14	6	0

Table 6 shows that both developers and project managers cite more group specific motivators than senior managers. The proportion of group specific de-motivators for developers and project managers are approximately 0.4, respectively, whereas senior manager specific de-motivators are less than 0.2. Indicating that the issues that de-motivate senior managers are less group specific.

On the one hand, of these ‘group specific’ de-motivators cited by developers and project managers there are more likely to be company specific than group specific. Company specific de-motivators are those that are more likely to be cited in only one focus group within a staff group as opposed to several times within the same staff group. For example, four out of eight developer specific de-motivators can be considered company specific. This is because developers cited customers, isolated best practice, PM’s lack of technical knowledge and reduced creativity in only one focus group session. Indicating that these issues may only be critical in those particular companies.

Table 6
De-motivators cited in only specific groups

De-motivator	Cited (Freq.)		
	Dev	PM	SM
Lack of mgt direction/commitment	4		
Workload	3		
Lack of feedback	2		
Lack of standards	2		
Customers	1		
Isolated best practice	1		
PM’s lack of technical knowledge	1		
Reduced creativity	1		
Low process priority		4	
Staff turnover		2	
Fire fighting		1	
Inadequate metrics		1	
Irrelevant objectives/deliverables		1	
Large-scale programmes		1	
Lack of SPI management skills			4
Organisational changes			1

On the other hand, there are other de-motivators that were cited across several companies but within the same practitioner group. For example, senior managers cited organisational changes whilst project managers cited low process priority and staff turnover. Such de-motivators appear to be specific to the particular practitioner groups and merit further investigation as to why they are not acknowledged by other groups. These de-motivators are probably directly related to specific functions within particular staff groups.

5. Discussion

In this section we discuss the major findings from our study and attempt to answer the research questions set out at the beginning of this study. We discuss the major issues that de-motivate practitioners and the similarities and differences in de-motivators across staff groups.

5.1. Major de-motivators

The following are the major de-motivators identified by practitioners from our findings.

5.1.1. Resources for SPI

Having analysed the data from this study, we can answer the following research questions on resources:

> RQ4: Does a lack of resources de-motivate software practitioners from supporting SPI?
> RQ5: Do commercial pressures de-motivate software practitioners from supporting SPI?

Our findings indicate that all practitioner groups are concerned about the effect of resource constraints on motivations for SPI. Developers and senior managers perceive these in terms of the constraints of project budgets whilst project managers view it as a general lack of resources. Practitioners generally view these constraints as de-motivating to the SPI effort.

Practitioners also indicate that pressure from the market to keep up with competition in terms of maintaining a company’s position in the market tends to frustrate SPI. Such pressures come from the need to satisfying time to market demands and meet customer demands. Practitioners indicate that such commercial pressures make it difficult to actively devote resources to SPI. Practitioners find that in an environment where the commercial imperative is so high, incentives for SPI are compromised.

This result also suggests that if software practitioners are not embarking upon SPI when commercial pressures are high, then they do not perceive SPI as cost and schedule effective. In effect, software practitioners perceive that SPI might impinge on commercial success.

5.1.2. Resistance to SPI

Our findings on inertia, bad experiences and general lack of support for SPI enable us to explore the issue of practitioner resistance to SPI. More specifically, we are able to answer the following research question:

RQ1: Are software practitioners resistant to SPI?

All practitioners indicate that inertia, negative previous experiences and lack of overall support de-motivates support for SPI.

Practitioners may resist SPI due to their negative experiences of previous SPI programmes. As Humphrey describes in his studies of the personal software process, past experiences can make practitioners think that the new processes will not improve their output (Humphrey, 1998). Such negative experiences can create resistance to SPI, which can prevent practitioners from supporting SPI.

Practitioners will also not support SPI if they are unwilling to give up established ways of working. Hovenden et al. recount the actions of the 'maverick practitioners' who frustrate the quality improvement effort in companies as they fear that it will threaten their established way of working (Hovenden et al., 1996). Humphrey also indicates that practices become established in practitioners themselves, thereby creating intransigence to new ways of working. Such intransigence means that practitioners fail to support SPI.

Lack of support for SPI becomes de-motivating for other practitioners who may not have necessarily resisted SPI, but become frustrated by the apathy that others have for SPI. Our findings show that all practitioner groups find that overall lack of support for SPI de-motivating.

For these reasons, practitioners are resistant to SPI.

5.1.3. Evidence for SPI

From our findings on evidence of SPI success on practitioner support for SPI we are able to answer the following research question:

RQ2: Does a lack of evidence de-motivate software practitioners from supporting SPI?

Project managers and senior managers indicate that the lack of direct evidence showing the benefits of SPI is de-motivating. Developers are the only group that does not cite a lack of evidence as de-motivating for their support of SPI. This indicates that the issue of evidence is more prominent for manager groups than it is for developers.

5.1.4. Skills for SPI

Senior mangers indicate that a lack of SPI management skills is a de-motivator for SPI. They also indicate that it is critical to have people with the expertise to drive SPI programmes. Absence of such skill de-motivates the SPI effort. This perception is supported indirectly by developers and project managers.

Developers and senior managers say that working with inexperienced staff de-motivates the SPI effort. Project managers on the other hand highlight staff turnover as de-motivating. We suggest that both sets of de-motivators are related to a lack of SPI skills. Companies may have inexperienced staff because of high staff turnover. However, having inexperienced staff could also be due to company policy on training—where little to nothing is spent on training staff for SPI.

5.1.5. Imposed SPI initiatives

In this study practitioners have identified de-motivators that are directly related to the way SPI is implemented in companies:

- imposition,
- inadequate communication.

From these findings we are able to explore the following research question:

RQ3: Do imposed SPI initiatives de-motivate software practitioners?

Project managers and developers find that imposing SPI without prior consultation is de-motivating. Even though senior managers do not perceive imposition, particularly, as a de-motivator, they do acknowledge that inadequate communication is de-motivating for SPI.

All practitioner groups seem to be suggesting that SPI initiatives that do not involve practitioners, through consultation and communication, are de-motivating and unlikely to be supported fully by practitioners.

5.2. Spread of de-motivators across practitioner groups

Finally, our findings provide results of de-motivators across staff groups that enable us to answer the research question:

RQ6: Do developers, project managers and senior managers have different de-motivators for SPI?

Our results show that there are both similarities and differences in de-motivators across software practitioner groups. Out of these, nearly 45% are common to more than one practitioner group. These common de-motivators can be categorised into the following broad areas:

- resource related,
- commercial pressures,
- the actual process constraints,
- implementation issues,
- personnel factors.

Within the de-motivators that are practitioner group specific, developers and project managers have cited, proportionately, twice the number of de-motivators as senior managers. This may suggest that whereas developers and project managers may be aware of many senior manager de-motivators, the same cannot be said of senior managers' awareness of developers' and project managers' de-motivators. This is worrying finding.

Our findings also show that the group specific de-motivators are very much related to the roles that practitioners play in their companies.

Developer specific de-motivators are

- lack of feedback and lack of standards,
- workload,
- reduced creativity,
- customers,
- lack of management direction/commitment.

Project manager specific de-motivators are

- lack of measures for controlling projects,
- fire-fighting,
- low process priority,
- staff turnover.

Senior manager de-motivators:

- organisational changes,
- lack of SPI management skills.

6. Conclusion

In this study we have collected and analysed de-motivators for SPI from groups of software practitioners. We have presented the major issues that software practitioners indicate de-motivates them. We have also analysed the differences that exist across staff groups.

We have shown that software practitioners are de-motivated from supporting SPI by the lack of resources dedicated to SPI. Practitioners find time and budget restraints de-motivating. They find that the pressure from commercial commitments compromises the incentive for SPI.

We have also shown that software practitioners are resistant to SPI for a variety of reasons. Previous negative experiences of SPI prevent software practitioners from supporting SPI and software practitioners are resistant to SPI because they are unwilling to change their tried and established practices.

We have also shown that software practitioners are de-motivated by lack of evidence of the direct benefits of SPI to their practices. Practitioners are more likely to support SPI if they know how it will benefit them directly.

Software practitioners are more receptive to consultative initiatives. Therefore they find SPI initiatives that are imposed upon them de-motivating.

We have shown that there are common issues that de-motivate all groups of practitioners. These issues are varied and cover factors relating to resources, processes, implementation and people. There are, also, differences in the de-motivators for SPI amongst different staff groups of practitioners. These differences are often related to the role that practitioners play in software development.

Our findings also show that software practitioners are de-motivated from supporting SPI when there is a lack of appropriate SPI skills in their companies. Developers and senior managers find it de-motivating when project managers lack SPI skills. Project managers on the other hand see the problem of shortage in SPI skills as an indirect result of high staff turnover.

We present these findings as a guide to SPI managers to enable them to design more effective strategies for deploying SPI. These findings offer SPI managers insight into how to manage practitioner de-motivators for SPI. We suggest that SPI managers can improve SPI uptake in companies if they reflect these findings in their SPI strategies.

Acknowledgements

The work we report in this paper is part of a wider study into SPI. We are sincerely grateful to all the companies and practitioners (who, for reasons of confidentiality, must remain anonymous). This project is funded by the UK's Engineering and Physical Sciences Research Council under grant no. GR/L91962. We would like to thank Austen Rainer and Sarah Beecham of the University of Hertfordshire, UK and David Wilson of University of Technology, Sydney, Australia, for their input and contribution to this work.

Appendix A. Companies in study

Company number	HW/SW producer	UK or Multi-national?	Size (people)	SE size (people)	Age (years)	SW type
1	HW/SW	MN	>2000	>2000	>50	RT/EM
2	SW	UK	100–500	100–500	20–50	Bus
3	HW/SW	MN	>2000	500–2000	>50	RT/EM
4	HW/SW	MN	>2000	500–2000	>50	RT/EM
5	SW	MN	>2000	>2000	10–20	RT
6	SW	MN	>2000	>2000	10–20	RT
7	SW	MN	>2000	>2000	20–50	Packs
8	SW	UK	10–100	10–100	5–10	Bus
9	SW	MN	10–100	10–100	10–20	RT/EM
10	SW	MN	>2000	10–100	10–20	Sys/EM
11	HW/SW	MN	500–2000	11–25	20–50	RT/EM
12	HW/SW	UK	100–500	<10	20–50	EM
13	SW	UK	100	40	10–20	Bus

RT = Real time; EM = Embedded; Bus = Business systems; Packs = Packages; Sys = Systems software.

Appendix B. Definition of de-motivators

De-motivator	Definition
Budget constraints	Budgets do not allocate resources specifically to SPI, therefore SPI work becomes a drain on overall budget
Commercial pressures	Pressure to satisfy commercial/financial objectives of company
Cumbersome processes	Processes that are bureaucratic and difficult to implement
Customers	Direct interference from customers
Fire fighting	A policy of tackling problems as they occur as opposed to a proactive long-term strategy for tackling problems
Imposition	Imposing SPI as a dictate, without prior consultation with practitioners
Inadequate communication	Lack of communication between different levels in a company and between different functional areas
Inadequate metrics	Not collecting sufficient metrics to guard improvement
Inertia	Resistance to practices new
Inexperienced staff	New and temporary staff who are not sufficiently knowledgeable of company processes
Irrelevant objectives/deliverables	SPI objectives are not tailored to “real” needs that practitioners can identify with
Isolated best practice	Best practices are kept within departments/teams/groups and not shared within the company
Lack of evidence of direct benefits	Practitioners do not have or are not provided evidence of the success of SPI
Lack of feedback	Practitioners are not given feedback of the SPI outcomes, or of contributions they make towards SPI
Lack of mgt direction/commitment	Senior management do not demonstrate understanding nor commitment to SPI
Lack of overall support	SPI is not overwhelming supported by the practitioner involved in it. There is apathy amongst certain groups
Lack of resources	The company does not have the resources—staff, time, tools—to properly fund SPI. This is dissimilar to *Budgets*
Lack of SPI management skills	There are insufficient personnel with the appropriate skills to drive (manage) SPI in the company

Lack of standards (different platforms)	The software development function operates across different platforms. There are no overall standards to SW development
Large-scale programmes	The SPI initiative is too big for the company. To many facets going on at the same time. It creates co-ordination problems
Low process priority	SPI is given low priority with respect to other project activities
Negative/bad experience	Previous negative experiences of SPI negates against SPI uptake amongst practitioners
Organisational changes	Organisational changes that imply re-allocation of staff and responsibilities impact negatively on ongoing SPI programmes
Personality clashes	Individuals and personal politics frustrates the SPI effort
PM's lack of technical knowledge	Project managers do not possess technical knowledge of SW production hence are unable to appreciate the merits of SPI
Reduced creativity	Practitioners perception of SPI procedures is that it takes away their individual creativity and flair
Staff turnover	High staff turnover frustrates the nurturing of SPI culture. Ever having to teach new people the established processes
Time pressure/constraints	Pressure to deliver product on time frustrates SPI initiative
Workload	Practitioners have too much work, thus are unable to devote sufficient effort to SPI

References

Baddoo, N., Hall, T., 2002. Motivators of software process improvement: an analysis of practitioners' views. Journal of Systems and Software 62, pp. 85–96.

Couger, J.D., O'Callaghan, R., 1994. Comparing the motivations of Spanish and finish computer personnel with those of the United States. European Journal of Information Systems 3 (4), 285–291.

El Emam, K., Fusaro, P., Smith, B., 1999. Success factors and barriers for software process improvement. In: Messnarz, R., Tully, C. (Eds.), Better Software Practice for Business Benefit: Principles and Experience. IEEE Computer Society, Los Alamitos, CA, pp. 355–371.

Glass, R.L., 1999. The reality of software technology payoffs. Communications of the ACM 42 (2), 74–99.

Goldenson, D.R., Herbsleb, J.D., 1995. After the appraisal: a systematic survey of process improvement, its benefits and factors that influence success. Software Engineering Institute, CMU/SEI-95-TR-009 ADA302225. http://www.sei.cmu.edu/publications/documents/95.reports/95.tr.009.html.

Gray, E.M., Smith, W.L., 1998. On the limitation of software process assessment and the recognition of a required re-orientation for global process movement. Software Quality Journal 7, 21–34.

Hall, T., Wilson, D., 1997. Views of software quality: a field report. IEE Proceedings on Software Engineering 114 (2), 111–118.

Hantos, P., Gisbert, M., 2000. Identifying software productivity improvement approaches and risks: construction industry case study. IEEE Software 17 (1), 48–56.

Hovenden, F.M., Sharp, H.C., Walker, S.D., Woodman, M., 1996. Building quality into scientific software. Software Quality Journal (5), 25–32.

Humphrey, W.S., 1998. Why don't they practice what we preach? Annals of Software Engineering 6 (4), 201–222.

Khalil, O.E.M., Zawacki, R.A., Zawacki, P.A., Selim, A., 1997. What motivates Egyptian IS managers and personnel: some preliminary results. SIGCPR 97.

Kitson, D.H., Masters, S.M., 1993. An analysis of SEI software process assessment results: 1987–1991. In: 15th International Conference on Software Engineering, Baltimore, Maryland, May 17–21.

Krippendorf, K., 1980. Content Analysis: An Introduction to Its Methodology. SAGE Publications Ltd., London.

Krueger, R.A., Casey, M.A., 2000. Focus Groups: A Practical Guide For Applied Research, third ed. Sage Publications.

Kuilboer, J.P., Ashrafi, N., 2000. Software process and product improvement: an empirical assessment. Information and Software Technology 42, 27–34.

Morgan, D.L., 1997. Focus Groups As Qualitative Research, second ed. Sage Publications.

Morgan, D.L., Krueger, R.A., 1993. When to use focus groups and why? In: Morgan, D.L. (Ed.), Successful Focus Groups: Advancing The State of the Art. Sage Publications, pp. 3–19.

Pitterman, B., 2000. Telcordia Technologies: the journey to high maturity. IEEE Software 17 (4), 89–96.

Rainer, A., Hall, T., Baddoo, N., 2002. Persuading developers to 'buy into' software process improvement: an exploratory analysis. Computer Science Technical Report CS-TR-367, University of Hertfordshire.

Nathan Baddoo is a researcher in the Department of Computer Science, University of Hertfordshire, UK. Nathan has spent the last three years investigating the motivators and de-motivators for SPI in UK software companies. His general interests are in the areas of software quality and software process improvement. He has published and presented material at seminars, workshops and conferences, both in the UK and in the US. He holds a Ph.D. from the University of Hertfordshire.

Tracy Hall is a principal lecturer at the Department of Computer Science, University of Hertfordshire, UK. Tracy is an active researcher in the area of software measurement and software process improvement. She has many publications in this area. She holds a Ph.D. from the Centre for Software Reliability, City University, London.