A Taxonomy to Compare SPI Frameworks

Christian Printzell Halvorsen[1], Reidar Conradi[2]

[1] Galaxy Republic AS, Drammensveien 211,
N-0212 Oslo, Norway
`christian.p.halvorsen@superoffice.com`
[2] Norwegian University of Science and Technology (NTNU),
N-7491 Trondheim, Norway
`conradi@idi.ntnu.no`

Abstract. The principle behind *software process improvement (SPI)* is that product quality is strongly influenced by the quality of the associated software process for development and maintenance. A number of *SPI frameworks* have evolved from this principle. However, these frameworks are comprehensive and differ in a variety of aspects, making them difficult to compare objectively and to select between for a company. This paper discusses four comparison methods that can be used on SPI frameworks. We have explored one of them further and propose a new SPI framework *taxonomy*. Our taxonomy consists of 25 relevant characteristics, which can be used to point out framework similarities and differences on a high level. An example of how the taxonomy can be applied to six common SPI frameworks is provided.

1 Introduction

More and more people realize that the most critical challenges in software development are not purely technical. As for all engineering work, many problems lie in the interaction between technology and organization. Schedule and budget overruns are common, as are delivered software products with insufficient or unwanted functionality, poor reliability, inefficiency, and so on. These are serious deficiencies related to the development process. Evolution of software needs and requirements as well as an overall unpredictability are major challenges [14][24][25].

Software process improvement (SPI) sprung out as a solution to these process-related difficulties. The assumption behind SPI and the many SPI frameworks is that product quality is influenced by the quality of the process used to develop it. We should therefore focus our improvement efforts on the software process in order to improve software quality. This can explicitly be expressed by the following *causal relation*:

$$\textbf{Quality}(Process) \Rightarrow \textbf{Quality}(Product) \tag{1}$$

The words *quality* and *process* are vital to our discussion of SPI. The reader is encouraged to confer with other sources such as [9][14][18][29] for more information:

- *Quality* – Good quality implies happy users. However, Kitchenham states that quality "is hard to define, impossible to measure, easy to recognize" [16]. Nevertheless, most SPI frameworks specify a measurable *(process) quality indicator* from which process quality can be inferred.
- *Software process* – "...a set of partially ordered process steps, with sets of related products, human and computerized resources, organizational structures and constraints, intended to produce and maintain the requested software products" [18].

It is interesting to note that many of the differences between SPI frameworks stem from different interpretations of these two words.

1.1   Goals for this Discussion

Many people who work in the SPI domain have chosen one SPI framework as their favorite. This choice is mostly subjective and seldom based on objective evidence of appropriateness. A reason for this is that SPI frameworks are difficult to compare due to their comprehensiveness. The frameworks also differ in basic assumptions such as how the terms *software process* and *software quality* should be interpreted, making the comparison task even harder.

The goal of this paper is to make the comparison of SPI frameworks easier and more objective by presenting a *taxonomy*, which can be used to characterize the various frameworks. We hope it will clarify their similarities and differences, and act as a starting point for further information gathering.

The proposed taxonomy will be related to other comparison methods and we will end the discussion by applying it to the following SPI frameworks:

- Total Quality Management (TQM) [2].
- The Capability Maturity Model (CMM) [14][22][29].
  ISO 9000 [20].
- ISO/IEC 15504 (SPICE) [9][25][29].
- Quality Improvement Paradigm (QIP), Experience Factory (EF), Goal Question Metric (GQM) [4][5].
- SPIQ (Software Process Improvement for better Quality)[1] [26].

1.2   Related Work

There is an increasing amount of literature comparing the major SPI frameworks. Most is written in the last three years and generally covers only a small number of frameworks. A variety of comparison methods have been used, but the method of *framework mapping* seems particularly popular.

---

[1] Software Process Improvement for better Quality (SPIQ) was a Norwegian R&D project in 1997-99, aiming to increase competitiveness and profitability in Norwegian IT industry through systematic and continuous process improvement. SPIQ proposes a unified, lean and pragmatical SPI framework.

In his Ph.D. thesis Sørumgård [24] compared the Capability Maturity Model (CMM), ISO 9000 and the QIP/EIF/GQM approach. He compared the frameworks in pairs using both a textual description and a tabular list of characteristics.

Tingey [28] has dedicated a whole book to the comparison of the CMM, ISO 9000 and the Malcolm Baldridge National Quality Award (MBA). His book presents an in-depth comparison of the three approaches, including a complete mapping between statements and a quantitative analysis of framework focus.

Emam, Drouin and Melo [9] have included a mapping between SPICE and the ISO 9000 standard in their book as well as a textual discussion of their differences.

The Software Engineering Institute (SEI) has published a high-level mapping between aspects of the ISO/IEC 12207 software life cycle, SPICE, the CMM v1.1 and the CMM v2.0 Draft C [23].

Several short articles have also been published about the subject. Haase [11] compares some of the assessment concepts in ISO 9000, SPICE, CMM, Bootstrap, SynQuest and EQA. Bamford and Deibler [3], and Paulk [21] both compare ISO 9000 with CMM. Cattaneo, Fuggetta and Sciuto [6] compare CMM, MBA and ISO 9000. Ares et al. propose the ATAM method to compare assessment frameworks according to six main categories [1]. Lastly, this paper is extended from a position paper at the NASA Software Engineering Workshop in Nov. 1999 [10].

2 Comparing SPI Frameworks

One goal of comparing SPI frameworks is to provide insight into their similarities and differences. However, the kind of insight needed is highly dependent on the context. Tingey [28] discussed two organizational scenarios with different requirements imposed on the comparison method. The essential differences lie in *level of detail* and *point of view*.

First, there are organizations *without* an SPI strategy that would like to learn about different SPI approaches because of competitive pressure or certification requirements. Such organizations usually have only limited general SPI knowledge and even less knowledge about specific frameworks. This requires the comparison to be on a high level with few details.

Secondly, there are organizations *with* an organized and systematic SPI strategy that would like to learn about other frameworks. Many organizations find themselves in a situation where they have to use more than one SPI approach. As an aid in the process of learning about other frameworks one would obviously like to utilize already existent SPI knowledge. A discussion of the lesser-known frameworks from the point of view of the well-known framework could therefore be beneficial.

From our previous discussion we suggest the following four classes of comparison methods:

- Characteristics [1] [6][11] [24]
- Framework mapping (comparing kernel concepts) [9][21][22][23][28]
- Bilateral comparison (comparing textual phrases) [9][21]
- Needs mapping (comparing user needs vs. framework properties)

Each of these will be discussed shortly. It should be clear that it would be neither beneficial nor possible to define disjoint comparison methods. Thus, some overlap is inevitable. Also, differences in *level of detail* and *point of view* should be expected as they satisfy different requirements.

Our new taxonomy presented in this paper is based on the characteristics comparison method, which will be described next.

2.1 Characteristics Comparison Method

We have decided to compare the various frameworks in a taxonomy by defining an extensive list of relevant *characteristics*. Each framework is then described in terms of these characteristics and the results are presented in a tabular format.

This type of comparison is well suited for a general overview of the frameworks and it can be used as a basis for other comparison methods. A lot of information can be inferred from a table listing the characteristics of several frameworks. However, the comparison is on a high level and details must be collected elsewhere. For this purpose one of the other comparison methods can be used.

The characteristics should preferably be *objective*, *measurable* and *comparable*. One can argue what kind of measurement scale should be used, e.g. nominal, ordinal or absolute. Each has its strengths and weaknesses.

2.2 Framework Mapping Comparison Method

Much of the recent comparison work has dealt with *framework mapping*. Each framework, at least the more structured ones, consists of a more or less defined set of statements or requirements dealing with the content and focus of the framework. Framework mapping is the process of creating a map from statements or concepts of one framework to those of another. There are two distinct ways to do this:

- Mapping between existing frameworks.
- Mapping existing frameworks into basis framework established solely for the purpose of comparing. This provides a common point of reference from which the frameworks can be evaluated, compared and contrasted.

In the characteristics approach the goal was to describe key attributes of each SPI framework. However, the purpose of mapping is to identify overlaps and correlations between frameworks and create a map of these statements or correlations. There can exist strong, weak or no correlation as suggested by Tingey [28]. Furthermore, the mapping can be done on either a high or a low level depending on the amount of detail one includes. This also dictates how the mapping results can be displayed, e.g. in a matrix or some kind of visual representations such as a Venn diagram.

Framework mapping is especially useful when an organization employs two or more different SPI frameworks, as corresponding statements can be identified and redundancy reduced. The extra effort needed to employ more than one framework can therefore be minimized.

Framework mapping is definitely a more low-level and detailed comparison method than characteristics. Because mapping goes into the specifics of each framework, it is not very useful for a general overview. However, mapping into a basis framework and supplementing with a quantitative analysis can indicate overall focus and content.

Framework mapping requires that some simplifying assumptions are made. The results are necessarily biased by these assumptions as suggested by Tingey [28].

2.3 Bilateral Comparison Method

In a *bilateral comparison* two frameworks are compared textually, for example the CMM and ISO 9000 [21]. The difference between this comparison method and the two previously described is its textual nature. A bilateral comparison is often a summary or explanation of findings from other the comparison methods.

The bilateral comparison can take on the point of view of one framework and describe the other in terms of that. This is convenient for people with detailed knowledge of one framework, as they can easily get insight into another using familiar terms.

The amount of detail included in a bilateral comparison can vary widely, depending on the purpose for which it is written. Frequently the level of detail is somewhere in between that of characteristics and the mapping approach.

2.4 Needs Mapping Comparison Method

Needs mapping is not a direct comparison between frameworks. Instead, it considers organizational and environmental needs that must be taken into account when selecting which SPI framework to adopt. The requirements imposed by such needs are often highly demanding and can limit the choice of framework severely. Nonetheless, they are of utmost importance and must be considered carefully. Here are some examples:

- Certification requirements, for example to ISO 9000, often imposed on a subcontractor.
- Top-level management requires that the chosen SPI approach should be incorporated in a Total Quality Management (TQM) strategy.
- Financial limitations.

There certainly exist other examples as well, and they can vary substantially from organization to organization, or depend on the business environment. Furthermore, the needs may vary over time as the organization or environment evolves.

3 The Taxonomy

In this paper we propose a *taxonomy* based on the characteristic comparison method described previously. We present a list of 25 characteristics regarded as important for describing SPI frameworks. Sørumgård [24] and Cattaneo et al. [6] have been particularly influential on our work, but we have also tried to capture aspects generally regarded as important in SPI literature (as cited previously).

One could argue that 25 characteristics are too many and that many of them are overlapping. However, the goal is to make a short tabular presentation, which requires each characteristic to have only a short description. More general characteristics would need longer descriptions, which could not easily fit in small table cells. In addition, the overlap between some of the characteristics is important to capture small variations among the frameworks.

A description of each characteristic follows. They have been grouped in five categories to ease readability and comprehension, see Figure 1 below.

General	Process	Organization	Quality	Result
Geographic origin/spread	Assessment	Actors/roles/stakeholders	Quality perspective	Goal
Scientific origin	Assessor	Organization size	Progression	Process artifacts
Development/stability	Proc. improvement method	Coherence	Causal relation	Certification
Popularity	Improvement Initiation		Comparative	Cost of implementation
Software specifc	Focus			Validation
Prescriptive/descriptive	Analysis Techniques			
Adaptive				

Fig. 1. Categorization of Characteristics in the Proposed Taxonomy

3.1 General Category

This category of characteristics describes general attributes or features of SPI frameworks. Such attributes are often specific to each framework and frequently related to how the framework is constructed or designed.

Geographic Origin/Spread. This characteristic describes where in the world the approach originated as well as where it is commonly used today.

Scientific Origin. The scientific origin of a framework or approach is the philosophical background on which it is based. Some frameworks have originated from purely philosophical principles, while others are based on mathematical or statistical rules. Some frameworks can even be said to originate from previous frameworks, by borrowing heavily from them in terms of concepts, content and focus.

Development/Stability. Another important characteristic is the development and stability of the framework. One should expect that the stability of a framework increases as experience is gained over a number of years. Many frameworks depend on feedback from users to evolve as time goes by, e.g. identification of parts in the framework that are difficult to use or especially useful. Changes in the software engineering discipline may be reflected as well, for example new technologies or the introduction of new SPI frameworks and standards. Naturally, it is desirable to employ a framework that is both evolved and relatively stable.

Popularity. Popularity is also important for an organization selecting to employ a certain process improvement approach. One would like to be sure that support and further development of the framework is continuing in the foreseeable future. Those frameworks with a greater mass of supporters also have a tendency to receive better support.

Software Specific. Whether the framework is software specific or not reveals something about its scope. Some of the approaches are specifically directed towards software development and maintenance, while others are more general. The latter must be adapted to the software development domain and this requires an extra effort.

Prescriptive/Descriptive. A prescriptive framework, as the name suggests, prescribes requirements and processes that are mandatory. In contrast, a descriptive framework does not assign specific actions to be taken by the organization. Instead, it describes a state or certain expectations to be met without stating how they should be accomplished.

Adaptability. The degree of flexibility varies among the process improvement approaches. An adaptive approach has support for tailoring and is customizable for specific uses. A study by Kautz and Larsen [15] indicated that the use of comprehensive SPI approaches is still quite low, partly because of their rigidity. This suggests that adaptability is an important quality of a framework.

3.2 Process Category

The process category concerns characteristics that describe how the SPI framework is used.

Assessment. The improvement work is often initiated by an assessment of the organization's current software process capabilities. Frameworks that include such assessment methods specify techniques for assessing software processes, practices and infrastructure, with the result being an identification of strengths and weaknesses. The assessment often includes recommendations for process improvement and it can therefore be used as an improvement road map.
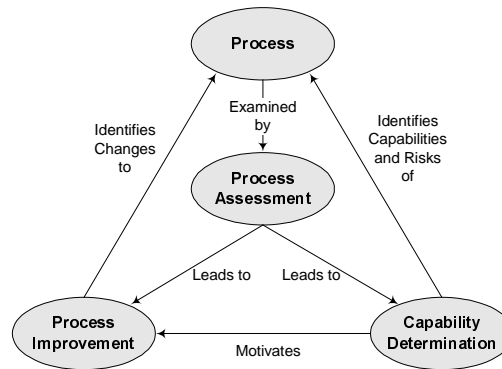
Fig. 2. Software Process Assessment (from [25])

There are at least two dimensions to assessments, namely *what* is assessed and *who* performs the assessment. The latter part is covered by the next characteristic called assessor.

Our investigation shows substantial differences between the frameworks in what their assessments are meant to cover. For example, some frameworks assess organizational maturity, others process maturity and then again others may assess customer satisfaction.

Assessor. The second dimension of assessments is *who* it is carried out by, i.e. the assessor. Generally, this is the organization itself or an external group depending on the purpose of the assessment. Some studies show that different assessors reach different conclusions when the same organization is assessed. This challenge should be addressed by the frameworks in question and a few initiatives have been made in this direction, e.g. TickIT.

In our taxonomy, assessments are characterized as either "external" or "internal" , or a combination of these.

Process Improvement Method. Some frameworks are purely assessment frameworks, while others include guidelines or process improvement methods used to implement and institutionalize the process improvement. A guideline like this is often called a process improvement life cycle. It "defines the strategy and the process by which improvement is pursued" [6]. This characteristic indicates the type of improvement method included in the framework, if any.

Improvement Initiation. Improvement initiation is related to the previous characteristic and indicates where the improvement work is initiated in the organization. In other words, it is the approach used to achieve improvement. The approach can for example be *top-down* or *bottom-up*. However, other approaches exist as well, e.g. SPICE initiates the improvement work at the level of *process instances*. A combination of these approaches could also have been used.

McGarry and Thomas have written an excellent article [19] comparing the top-down and bottom-up approach. The main difference is between *generality* (universal good practices exist and the organization's processes should be compared with these) and *specificity* (the importance of the local domain should be recognized).

Improvement Focus. Some frameworks focus primarily on experience reuse, whereas others concentrate on management processes or other areas. They differ in improvement focus, i.e. which SPI activities they regard as essential for improvement.

The difference between the focus and the goal characteristic, which will be described later, might not be clear and it deserves some extra attention. Improvement focus is related to activities and deals with the means of achieving the end result. It is therefore placed in the process category. In contrast, the goal characteristic is only concerned with the end result not how it was achieved.

Analysis Techniques. Their use of analysis techniques can also be employed to classify process improvement frameworks. Some frameworks use quantitative techniques, others qualitative and still others might not use any defined techniques at all. Examples of analysis techniques are questionnaires, statistical process control, various measurements etc.

3.3 Organization Category

The characteristics in the organization category are directly related to attributes of the organization and environment in which the SPI framework is used, i.e. who is involved and for what kind of organization is the framework applicable.

Actors/Roles/Stakeholders. Each framework has its own primary actors, which are the people, groups or organizations affected by or taking part in the improvement process. Generally, each such actor or stakeholder has a predefined role in the improvement process.

Organization Size. This characteristic indicates the organization size for which the SPI approach is suitable. Some frameworks are so comprehensive that only large corporations have the resources to use them. Other approaches are easier for smaller companies to employ. However, this relation between organization size and cost-effectiveness is hard to pinpoint and has been heavily debated [6] [8].

Coherence. Coherence is the logical connection between engineering factors and factors related to the business or organization. In an article by Cattaneo et al. [6] this was pointed out as extremely important for a successful SPI implementation, yet still insufficient in many process improvement frameworks. Cattaneo et al. stated that "most software process improvement efforts limit their focus and concern to the technical and engineering aspects of software development [...] They do not consider the relationships between these engineering factors and organizational and market variables."

There are two types of coherence and both should be maximized for the greatest benefits [6]:

- *Internal coherence* – The fit among the different parts of a company, such as organization, technology, human resources, strategy and culture.
- *External coherence* – The fit between the organization and its environment, including market, customers and competitors.

3.4 Quality Category

The quality category deals with characteristics related to the quality dimension by pointing out aspects such as how progression is measured, whose quality perspective is employed and what that means in terms of quality indicators and causal relations.

Quality Perspective. When looking at quality, it is necessary to establish a clear understanding of whose quality perspective the approach adopts. The concept of quality will naturally be different depending on who is asked. For example, the management will have a different quality perspective compared to what a customer might have.

Progression. If the SPI framework is looked upon as a road map, progression along the road can be characterized as "flat", "staged" or "continuous" as shown in Figure 3. ISO 9000 is a typical flat framework with only one step, namely certification. One cannot be more or less certified. On the other hand, in the CMM there is a discrete ladder of staged maturity levels, each indicating a specific level of process quality. There are also approaches that proclaim a continuous and evolutionary quality progression, e.g. TQM and QIP.
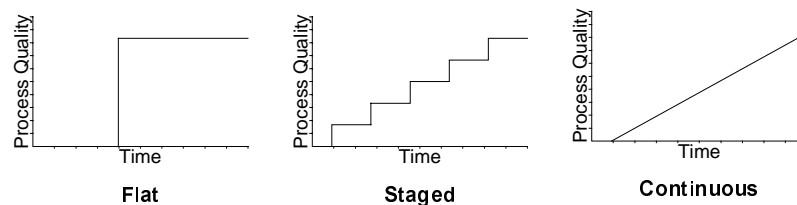
Fig. 3. The Progression Characteristic

Causal Relation. SPI frameworks assume that product quality is affected by the quality of the process used to produce it. However, process quality is difficult to determine and therefore most frameworks measure some *quality indicator* instead (cf. section 1). This suggests the following *causal relation*:

$$\mathbf{F}(\textit{Quality indicator}) \Rightarrow \mathbf{Quality}(\textit{Process}) \Rightarrow \mathbf{Quality}(\textit{Product}) \tag{2}$$

If an improvement can be detected in the quality indicator, the quality of the process is regarded as better and we should expect an improvement in product quality as well.

The quality indicator used differs among the frameworks, but is in all cases based on factors assumed to influence the process and product quality. We have used the functions $F(x)$, $F'(x)$, $F''(x)$ and so on to indicate such relationships. For example, equation (3) shows us that there is a relationship between the SPI framework and the quality indicator used. Furthermore, there is a relationship between this quality indicator and the quality of the process and the product:

$$\mathbf{F''}(\textit{SPI framework}) \Rightarrow \mathbf{F'}(\textit{Quality indicator}) \Rightarrow \mathbf{Quality}(\textit{Process}) \Rightarrow \tag{3}$$
$$\mathbf{Quality}(\textit{Product})$$

None of the frameworks explicitly states a causal relation like those above. However, they are implicit and this paper proposes this explicit representation.

Comparative. Some frameworks are comparative in nature, meaning that they can be used to compare certain aspects of different organizational units, either internally or externally. The aspect compared may vary between the frameworks, but it is supposed to represent an objective comparison criterion. For example, two organizations assessed using the CMM can be compared on the basis of their maturity levels, i.e. the one with the higher maturity level is assumed to have the higher process quality as well. In contrast, when using ISO 9000 the only comparison that can be made is whether the organizations are certified, a very coarse characterization.

Frameworks that are not comparative are only used as guidelines within a single project or organization.

3.5 Result Category

This category treats characteristics that describe the results of employing an SPI framework, but also the costs of reaching these results and the methods used to validate them.

Goal. The goal of the framework is its primary objective or end result, i.e. what is achieved by using the framework. As discussed previously it differs from the improvement focus characteristic by only treating the end result, not the means of getting there.

Process Artifacts. Process artifacts are the "things" created as a result of adopting an SPI framework. The process-related artifacts can be tangible, such as documents containing process models, improvement plans, experience data etc. However, the artifacts can also be intangible, e.g. increased knowledge in the organization. In either case, the process artifacts are created in addition to the actual product being produced.

Certification. The International Organization for Standardization (ISO) and a variety of national standard bodies have developed standards to which an organization can be certified. Only conformance with the minimum requirements is measured in the certification process. A standard to which one can be certified is comparative by nature.

Certification is closely related to the flat quality progression previously discussed, indicating that not all SPI approaches lead to certification. There has been some debate on whether a certain CMM level can be regarded as equivalent to ISO 9000 certification.

Implementation Cost. The implementation cost is naturally an important characteristic, because it often dictates whether the approach is financially feasible. There are considerable cost differences among the frameworks, making some impossible for smaller organizations to adopt. Emam et al. [8] state that "the implementation of certain processes or process management practices may not be as cost-effective for small organizations as for large ones" . However, the exact impact of organization size on cost is not obvious.

Because the cost is affected by a number of factors outside the SPI framework, it will be impossible to provide exact numbers in this characteristic. However, rough estimates should be provided where such numbers can be obtained, for example from case studies and other validation efforts. A set of articles on SPI cost/benefits is recently published by Curtis and co-workers in [7]. Some factors that can influence cost estimates are:

- *Organizations specific attributes* – For example organization size.
- *Inconsistent use of cost models* – Different cost models are used, variation in cost/benefit analysis, disagreement on measurement unit, disagreement on which SPI expenses to include in the cost estimates.

Even if the validity of the implementation costs may be questionable, a decision has been made to include it as a characteristic. However, the currently existing cost data is so limited in nature that its inclusion could have given a distorted, biased or even wrong view of reality. The reader will therefore notice that this characteristic is empty for all the frameworks. As more data is collected, some missing data can be included, e.g. as averages or medians, and with standard deviations or min/max values.

Validation. Validation is needed to evaluate whether the improvement efforts have actually worked. Such validation should distinguish between improvements resulting from SPI efforts and improvements resulting from other factors (context). The question we want answered, is whether the same benefits would have been achieved without a comprehensive SPI effort or with an alternative SPI effort. In other words, the difficulty lies in establishing a direct causal relation between the SPI effort and the achieved improvements. Figure 4 shows this problem. For instance, a major success factor may be the actual technology being used, e.g. OO, reuse, inspections etc.
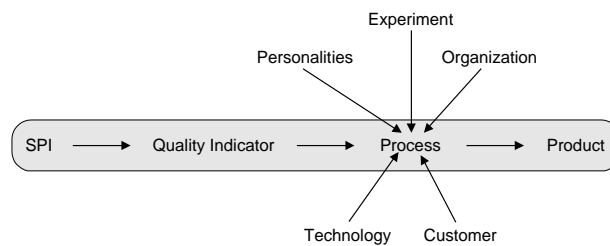
Fig. 4. Multifactor Problem

Some authors have reported that some "improvement" seems to be the outcome, regardless of the activities initiated (Figure 5: a possible "Hawthorne effect"?). Other initial improvements can be attributed to the elimination of obvious process deficiencies ("hot spots") [6][17]. Both these factors illustrate that real, causal benefits are hard to show. Often the actual improvement gains are only seen after 2-3 years ("process refinement"). However, software organizations often experience rapid evolution ("product innovation"), where products become obsolete over that time span -- cf. the term "Internet time" .
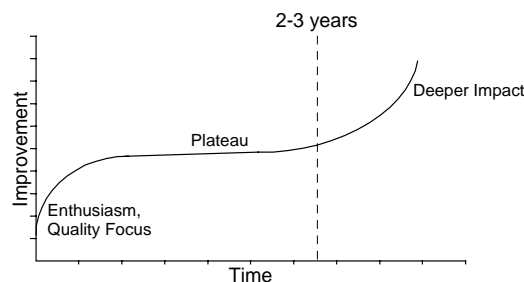
Fig. 5. Immediate vs. Deeper Impact

There have been several attempts to validate the effects of SPI frameworks, e.g. [12][13][15][17]. However, several critical questions can be raised against them:

- *Non-representative* – It is likely that successful organizations are more eager to publish their case studies than unsuccessful organizations [12].

- *Inconclusive* – Often only very vague conclusions are reached, such as "CMM seems to pay off" and "those respondents who reported a better performance tended to be from higher maturity organizations" [12]. Words like "seems" and "tended" reveal that a definitive causal relation between the SPI effort and the achieved improvement is hard to find.
- *Insufficient validation in general* – Many software engineering papers have insufficient or no validation, often based on ad hoc methods [27][30].
- *Less than favorable experiences* – A paper by Herbsleb and Goldenson [12] shows that quite a few people describe SPI as having no or very little effect (26%), or even being counterproductive (4%). To another question 49% answered that they were disillusioned over the lack of results.

However, the scope of this paper is merely to characterize and provide guidance for selection of SPI frameworks, not to demonstrate or validate their effectiveness.

4 Concluding Remarks

The proposed taxonomy has been applied to six SPI frameworks: TQM, CMM v1.1, ISO 9000, ISO/IEC 15504 (SPICE), QIP/EF/GQM, and SPIQ – see Table 2.

There are some cases where a characteristic may not be applicable for a given SPI framework. Those cases are indicated by NA for "Not applicable" . Furthermore, the scope of our taxonomy is rather extensive, so some characteristics could not be determined. The reason could be unavailability of information, or the necessity of further exploration and experimentation. In either case, these characteristic are marked with a dash (-). The reader should be aware of this difference between a characteristic that is not relevant and a characteristic that has yet to be determined.

4.1 Taxonomy Contributions

The goal of the taxonomy was stated in the introduction. It has been proposed as an objective method for comparing SPI frameworks and as a starting point for information gathering. Said differently, it is a high-level tool for use in the process of selecting which SPI framework should be employed in an organization. As such it is indirectly influencing product quality:

$$\mathbf{F'''}(\textit{Taxonomy}) \Rightarrow \mathbf{F''}(\textit{SPI framework}) \Rightarrow \qquad (4)$$
$$\mathbf{F'}(\textit{Quality indicator}) \Rightarrow \mathbf{Quality}(\textit{Process}) \Rightarrow \mathbf{Quality}(\textit{Product})$$

The taxonomy in itself does not provide enough information to enable a sound decision in this selection process. However, it points out areas of interest for someone investigating different SPI frameworks. Such a high-level comparison is needed because of the multitude, diversity and comprehensiveness of current SPI frameworks. It is a daunting and costly task to learn about all of them. Clearly it is valuable to quickly get an overview, so that one or a few frameworks can be chosen for further investigation.

A natural question to ask is whether company SPI efforts, that have been only partially successful, have employed the "wrong" SPI framework. To be successful, the framework's quality indicators must represent the relevant quality of the process and product -- cf. the characteristics "coherence" and "validation" . Typical process qualities are development time and cost, while product quality could be reliability, usability, or even fancyness. It is tempting to "over-structure" the process, and rely too much on the process quality indicators. Further, it is difficult to say that one process indicator or SPI framework is better than another, as long as the customer is satisfied. For example, Nokia has largely been assessed to CMM level 1, while its competitor Motorola has many of its departments on CMM level 5. Even so, Nokia hardly appears inferior to a customer.

The causal relation stated first in this section implies that the taxonomy ultimately affects product quality. However, this through-going causal relation is vague, and *validating* the taxonomy in this respect is close to impossible. Much of the vagueness, however, stems from the problematic causal relation between a given SPI framework and product quality, not being the subject of this paper. A validation of our taxonomy should therefore consider it *in its own right* -- whether it is fit for its purpose. We will therefore only consider the first step in the causal relation:

$$\mathbf{F'''}(\textit{Taxonomy}) \Rightarrow \mathbf{F''}(\textit{SPI framework}) \qquad (5)$$

There are two sides of this. First, it should be indicated how the taxonomy leads to the selection of the "right" SPI framework, as suggested by the causal relation between them. Secondly, the relation above implies that there are several ways that SPI frameworks can be objectively compared, i.e. the choice of SPI framework is a function of the comparison method. A validation effort should therefore include evidence that the taxonomy represents an appropriate comparison method.

With respect to *appropriateness*: The proposed taxonomy is based on an in-depth literature study of *commonly used characteristics* to describe and compare SPI frameworks. It is built on previous comparison work [6][24][28]. The taxonomy has been applied to six SPI frameworks, and we believe it can be applied to other as well.

Further, the taxonomy is very *compact* and provides a quick way of acquiring knowledge. Thus the taxonomy represents an appropriate way of pragmatically describing the frameworks on a *high level*.

Up until this point, little has been said about the *objectivity* of the taxonomy. It is possible to determine some of the characteristics objectively, but not e.g. imple-mentation cost. However, the same argument applies to other comparison methods.

Comprehensibility and *usefulness* are the most important aspects of the taxonomy, and should be used in an *empirical validation* to assess *company satisfaction*. A newcomer to SPI should have little trouble using the taxonomy, while an experienced SPI user should be able to infer even more information. It should also be useful for beginner companies that want to start a systematic SPI effort, and for more mature companies that want to combine parts of different SPI frameworks (e.g. ISO 9000 and CMM).

Table 1. SPI framework causal relations. (F' is split into F1' and F2' for three frameworks, and is similar for the last two).

SPI Framework	Causal Relation
TQM	Not applicable
CMM	**F1'**(*Key Process areas*) $\Rightarrow$ **F2'**(*Maturity level*) $\Rightarrow$ **Quality**(*Process*) $\Rightarrow$ **Quality**(*Product*)
ISO 9000	**F1'**(*Quality elements*) $\Rightarrow$ **F2'**(*Certification*) $\Rightarrow$ **Quality**(*Process*) $\Rightarrow$ **Quality**(*Product*)
ISO/IEC 15504 (SPICE)	**F1'**(*Process attributes*) $\Rightarrow$ **F2'**(*Capability level*) $\Rightarrow$ **Quality**(*Process*) $\Rightarrow$ **Quality**(*Product*)
QIP/GQM/EF	**F'**(*Experience reuse*) $\Rightarrow$ **Quality**(*Process*) $\Rightarrow$ **Quality**(*Product*)
SPIQ	**F'**(*Experience reuse*) $\Rightarrow$ **Quality**(*Process*) $\Rightarrow$ **Quality**(*Product*)

5 Further Work

We have seen that the proposed taxonomy seems reasonable from a theoretical standpoint. However, there are many issues that can only be answered through practical use and further investigation:

- *Cost* – An evaluation of the costs/benefits incurred from use of the taxonomy, for example in terms of training and competence building. We also need more cost data on implementation of the various frameworks.
- *Effectiveness* – Which parts of the framework seem to be the most useful, for whom and for what? We may apply the Goal Question Metric (GQM) method on the taxonomy itself, in order to assess its usability to help making right SPI decisions.
- *Ease-of-use* – Which parts or subsets of the taxonomy appear to be the easiest or hardest to use?
- *Taxonomy as basis for mapping* – Investigation on how the taxonomy can be used as a basis for a more detailed mapping approach.

References

1. Ares Juan, Garcia Rafael, Juristo Natalia, López Marta, Moreno Ana M., A more rigorous and comprehensive approach to software process assessment. *Software Process - Improvement and Practice*. Vol. 5, No. 1 (March 2000), pp 3-30.
2. Arthur Lowell J., *Improving Software Quality - An Insider's Guide to TQM*. John Wiley, 1993.
3. Bamford Robert C., Deibler William J. II, Comparing, contrasting ISO 9001 and the SEI capability maturity model, *Computer*, Vol. 26, No. 10, Oct. 1993.

4. Basili Victor R. The Experience Factory and its Relationship to Other Improvement Paradigms. *4th European Software Engineering Conf. (ESEC'93)*, Springer Verlag LNCS 517, 1993, pp 68-83.

5. Basili Victor R., Caldiera Gianluigi, Rombach H.-Dieter, *Goal Question Metric Approach*. Encyclopedia of Software Engineering (John J. Marciniak, Ed.), John Wiley, 1994, Vol. 1, pp. 528-532.

6. Cattaneo F., Fuggetta A. and Sciuto D., Pursuing Coherence in Software Process Assessment and Improvement. Forthcoming in *Software Process - Improvement and Practice*, 2001, 46 p.

7. Curtis, Bill, The Global Pursuit of Process Maturity, IEEE Software, Vol. 17, No. 4 (July/Aug. 2000), p. 76-78 (introduction to special issue on SPI results).

8. Emam Khaled El and Briand Lionel, *Chapter 7 - Costs and Benefits of Software Process Improvement (part of future book)*. Fraunhofer IESE, Kaiserslautern, Germany, 1997.

9. Emam Khaled El, Drouin Jean-Normand, Melo Walcélio, *SPICE - The Theory and Practice of Software Process Improvement and Capability dEtermination*. IEEE CS-Press, Nov. 1997, 450 p.

10. Halvorsen Christian P. and Conradi Reidar, A Taxonomy for SPI Frameworks, *24th NASA Software Engineering Workshop*, Greenbelt/Washington, USA, 1--2 Dec.1999, 4 p. See http://sel.gsfc.nasa.gov/website/sew/1999/program.html. Also as NTNU SU-report 19/99.

11. Haase V. H., Software process assessment concepts. *Journal of Systems Architecture*, Vol. 42, Nr. 8, Dec. 1996, pp 621-631.

12. Herbsleb James D. and Goldenson Dennis R., A Systematic Survey of CMM Experience and Results. *18th Int'l Conf. on Software Engineering (ICSE'96)*, Berlin, March1996, IEEE-CS Press, pp 323-330.

13. Herbsleb James, Carleton Anita, Rozum James, Siegel Jane, Zubrow David, Benefits of CMM-Based Software Process Improvement: Initial Results. *Tech. Report CMU/SEI-94-TR-13*, Aug. 1994, 53 p.

14. Humphrey Watts S., *Managing the Software Process*. Addison-Wesley, 1989.

15. Kautz Karlheinz and Larsen Even Åby, *Diffusion and Use of Quality Assurance and Software Process Improvement Approaches in Norway: A Survey-Based Study*. (Report no 906, 25 p. Norwegian Computing Center, Oslo, April 1996. ESPITI project.)

16. Kitchenham B. A., Evaluating Software Engineering Methods and Tools. *ACM SIGSOFT Software Engineering Notes*, 1996.

17. Løken Cecilie B. and Skramstad Torbjørn, ISO 9000 Certification - Experiences from Europe. 11 p. *First World Congress for Software Quality*, San Francisco, June 1995.

18. Lonchamp Jacques, A Structured Conceptual and Terminological Framework for Software Process Engineering. *Second Int'l Conference on the Software Process*. IEEE-CS Press, 1993, pp 41-53.

19. McGarry Frank and Thomas Martyn, Top-Down vs. Bottom-Up Process Improvement. *IEEE Software*, July 1994, pp 12-13.

20. Oskarsson Östen, Glass Robert L., *An ISO 9000 Approach to Building Quality Software*. Prentice Hall. 1996.

21. Paulk Mark C., How ISO 9001 Compares with the CMM, *IEEE Software*, Vol. 12, No. 1 (Jan 1995), pp 74-82.

22. Paulk Mark C., Weber Charles V., Curtis Bill, Chrissis Mary Beth et al., *The Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley, 1995.

23. Software Engineering Institute, *Top-Level Standards Map - ISO 12207, ISO 15504 (Jan 1998 TR), Software CMM v1.1 and v2 Draft C*. (Internal report Software Engineering Institute, Feb. 28, 1998.)

24. Sørumgård Sivert, Verification of Process Conformance in Empirical Studies of Software Development. IDI doctoral thesis 1997:14, NTNU, 252 p.

25. SPICE 1998, *Software Process Improvement and Capability dEtermination Web Site*. [Online]. Accessible from: http://www.seq.iit.nrc.ca/spice/ [Last accessed: 12.01.2001]

26. Dybå Tore (ed.): *SPIQ - Software Process Improvement for better Quality: Method Handbook (in Norwegian)* (SINTEF/NTNU/UiO), Trondheim/Oslo, ISSN 0802-6394, Jan. 2000, ca. 250 p.

27. Tichy Walter F., Should Computer Scientists Experiment More? *IEEE Computer*, May 1998, pp 32-40.

28. Tingey Michael O., *Comparing ISO 9000, Malcolm Baldridge, and the SEI CMM for software: a reference and selection guide*. Upper Saddle River: Prentice-Hall, 1997.

29. Zahran Sami, *Software Process Improvement - Practical Guidelines for Business Success*. Essex: Addison Wesley Longman, 1998.

30. Zelkowitz Marvin V. and Wallace Dolores, Experimental validation in software engineering. *Information and Software Technology*, Vol. 39, 1997, pp 735-743.

Appendix: The Taxonomy applied to Six SPI Frameworks

Table 2. The Taxonomy Applied to TQM, CMM v1.1 and ISO 9000

Category	Characteristic	TQM	CMM v1.1	ISO 9000
General	Geographic Origin/ Spread	Japan/World	U.S./World	Europe/World
	Scientific Origin	Quality control	TQM, SPC	_[3]
	Development/ Stability	Entire post-war era	Since 1986	Since 1987
	Popularity	High (esp. in Japan)	Top (esp. in U.S.)	High (esp. in Europe)
	Software Specific	No	Yes	No
	Prescriptive/ Descriptive	Descriptive	Both	Both
	Adaptability	Yes	Limited	Limited
Process	Assessment	None	Org. maturity	Process
	Assessor	NA[2]	Internal and external	External
	Process Improvement Method	PDCA	IDEAL	None
	Improvement Initiation	Top-down	Top-down	NA[2]
	Improvement Focus	Management processes	Management processes	Management processes
	Analysis Techniques	7QC, 7MP, SPC, QFD	Assessment questionnaires	ISO guidelines and checklists
Organ-ization	Actors/Roles/ Stakeholders	Customer, employees, management	Management	Customer, supplier
	Organization Size	Large	Large	Large
	Coherence	Internal and external	Internal	Internal and limited external
Quality	Quality Perspective	Customer	Management	Customer
	Progression	Continuous	Staged	Flat
	Causal Relation	NA[2]	**F1'**(*Key process areas*) $\Rightarrow$ **F2'**(*Maturity level*) $\Rightarrow$ **Q**(*Process*) $\Rightarrow$ **Q**(*Product*)	**F1'**(*Quality elements*) $\Rightarrow$ **F2'**(*Certification*) $\Rightarrow$ **Q**(*Process*) $\Rightarrow$ **Q**(*Product*)
	Comparative	No	Yes, maturity level	Yes, certification
Result	Goal	Customer satisfaction	Process improvement, supplier capability determination	Establish core management processes
	Process Artifacts	Plans, diagrams	Process documentation, assessment result	Process documentation, certificate
	Certification	No	No	Yes
	Implementation Cost	_[3]	_[3]	_[3]
	Validation	None	Surveys and case studies	Survey

---

[2] Not applicable.

[3] Not yet determined.

Table 3. The Taxonomy applied to ISO/IEC 15504, QIP/EF/GQM and SPIQ

Category	Characteristic	ISO/IEC 15504	QIP/EF/GQM	SPIQ
General	Geographic Origin/ Spread	World/World	U.S./World	Norway/Norway
	Scientific Origin	CMM, Bootstrap, Trillium, SPQA	Partly TQM	TQM, QIP/EF/GQM, ESSI model
	Development/ Stability	Under development	Since 1976	Under development
	Popularity	Growing	Medium	Norway only
	Software Specific	Yes	Yes	Yes
	Prescriptive/ Descriptive	Both	Descriptive	Descriptive
	Adaptability	Yes	Yes	Yes
Process	Assessment	Process maturity	None	Customer satisfaction
	Assessor	Internal and external	NA[4]	Limited internal
	Process Improvement Method	SPICE Doc. part 7	QIP	Two-level PDCA
	Improvement Initiation	Process instance	Iterative bottom-up	Top-down and iterative, bottom-up
	Improvement Focus	Management processes	Experience reuse	Experience reuse
	Analysis Techniques	Several (manual and automated). Required.	GQM	GQM, QFD, 7QC, 7MP
Organ-ization	Actors/Roles/ Stakeholders	Management	Experience factory, project organization	Customer, experience factory, project org., sponsoring org.
	Organization Size	All	All	All
	Coherence	Internal	Internal	Internal and external
Quality	Quality Perspective	Management	All	Customer, all
	Progression	Continuous (staged at process instance level)	Continuous	Continuous
	Causal Relation	$\mathbf{F1'}$(*Process attributes*) $\Rightarrow$ $\mathbf{F2'}$(*Capability level*) $\Rightarrow$ $\mathbf{Q}$(*Process*) $\Rightarrow$ $\mathbf{Q}$(*Product*)	$\mathbf{F'}$(*Experience reuse*) $\Rightarrow$ $\mathbf{Q}$(*Process*) $\Rightarrow$ $\mathbf{Q}$(*Product*), but feedback loops here	$\mathbf{F'}$(*Experience reuse*) $\Rightarrow$ $\mathbf{Q}$(*Process*) $\Rightarrow$ $\mathbf{Q}$(*Product*)
	Comparative	Yes, maturity profile	No	No
Result	Goal	Process assessment	Organization specific	Increased competitiveness
	Process Artifacts	Process profile, assessment record	Experience packages, GQM models	Experience packages, GQM models
	Certification	No	No	No
	Implementation Cost	_[5]	_[5]	_[5]
	Validation	Document review, trials (case studies and surveys)	Experimental and case studies	Experimental and case studies

---

[4] Not applicable.

[5] Not yet determined.