University of Technology, Dresden

Department of Systems Engineering

Pattern-based Software Development
An Empirical Study

- Summary of Results -

Thoralf Czichy (thoralf@czichy.org)

Helsinki, 03. December 2001

Abstract

This paper presents the results of a survey on the use of patterns in software development that was conducted by the University of Technology, Dresden, Germany. After a short introduction to the basic theoretical foundations of this survey, seven areas of interest are presented:

1. Understanding of patterns
2. Patterns in an organisation
3. Different pattern types - usage and tool support
4. Pattern usage and pattern mining in the process model
5. Characterising patterns - pattern descriptions and pattern organisation schemes
6. Experience with patterns
7. The value of patterns for an organisation

Based on these areas a web questionnaire was designed. In the second part of this paper, the results from the survey are analysed. Each area is discussed separately.

Table of Contents

0. Introduction *1*

1. Patterns in Software Development *2*

1.1 A Classification of Patterns based on Software Development Activities **2**

1.2 A Microprocess for Patterns **4**

1.3 Pattern Automation and the Microprocess for Patterns **7**

2. General Aspects of the Survey *10*

2.1 Areas of Interest (AoI) **10**

2.2 Related Work **12**

2.3 Design and Realisation of the Survey **13**

3. Summary of Results *16*

3.1 General Information **16**

3.2 Significance of Results **18**

3.3 Areas of Interest **20**

3.3.1 Understanding of Patterns (AoI1) 20

3.3.2 Patterns in an Organisation (AoI2) 21

3.3.3 Different Pattern Types – Usage and Tool Support (AoI3) 26

3.3.4 Pattern Usage and Pattern Mining in the Process Model (AoI4) 32

3.3.5 Characterising Patterns – Descriptions and Organisation Schemes (AoI5) 35

3.3.6 Experience with Patterns (AoI6) 38

3.3.7 The Value of Patterns for an Organisation (AoI7) 45

4. Conclusion *47*

Table of Figures ..... *I*

References ..... *III*

Appendix 1: List of Pattern Qualities ..... *VII*

Appendix 2: Table of Data for Question 6A ..... *VIII*

Appendix 3: Open Answers to Question 6A ..... *X*

0. Introduction

Except for several experience reports, only few studies about the use of patterns in software development have been conducted (see [4], [36], [37], [48]). Most of them were limited to an analysis of design patterns.

While the origin of the reuse approach "patterns" is based on the development of design patterns [14], many other types of patterns have emerged since the mid-90s. Popular examples are analysis patterns, architectural patterns, and process patterns [2], [22], [28].

The goal of this paper is to give an overview of the usage of patterns in practice. This not only includes design patterns, but also other types of patterns will be analysed. New insights are based on an empirical study conducted during September and October 2001. The following areas of interest (AoI) are included in this study:

AoI1. Understanding of patterns

AoI2. Patterns in an organisation

AoI3. Different pattern types - usage and tool support

AoI4. Pattern usage and pattern mining in the process model

AoI5. Characterising patterns - pattern descriptions and pattern organisation schemes

AoI6. Experience with patterns

AoI7. The value of patterns for an organisation

In the first chapter a short summary of the theoretical foundations being used in this study is given. Based on common activities in software development, eleven pattern types are presented. Within the survey these pattern types build a framework for an analysis of familiarity, frequency of use, and tool support in practice.

Furthermore a Microprocess for Pattern Application and Pattern Discovery is presented. Four approaches for pattern automation are characterised and put into the context of the

Microprocess presented. The respective questions about pattern automation approaches in the survey are based on this classification.

In the remaining part, following a short introduction to the survey, the results for each area of interest are presented one by one.

1. Patterns in Software Development

1.1 A Classification of Patterns based on Software Development Activities

Like many other terms in the realm of software, the term "pattern" has been used in many other application domains. This makes it even more difficult to find a common understanding of the term, since somewhere back in everyone's mind there is always a connection to real world applications of patterns. On one hand, this is good because a well chosen metaphor helps to grasp the basic principles behind a certain concept. On the other hand, even an excellent metaphor will not cover all aspects of the underlying concept correctly. Some aspects might be even wrongly explained.

In the remainder of this article, a commonly accepted definition by ALEXANDER is used:

"Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution" [1].

Even though this definition has its background in architecture, it can be applied to patterns in software development as well.

The use of patterns in software development is closely linked to design patterns which are applied to typical problems in the detailed design of software systems. Along the same lines one can apply the concept of patterns to other activities in software development as well. An initial classification is based on the phases of the waterfall model [41] and their respective activities:

PT1. Patterns for requirements analysis

PT2. Analysis patterns

PT3. Architectural patterns

PT4. Design patterns

PT5. Idioms (Implementation patterns)

PT6. Patterns for testing

PT7. Patterns for maintenance

According to the classification scheme presented here, patterns of these pattern types (PT) are known as activity-based. These patterns are closely linked to a certain activity. The distinction between architectural patterns and design patterns needs further clarification. While architectural patterns focus on architectural design, which defines the relationship between major structural elements of the software [38], design patterns emphasise less abstract problems closer to hardware/software implementations [28].

Contrary to *activity-based* patterns, there are other *activity-independent* patterns:

PT8. Process patterns

PT9. Organisational patterns

Process patterns describe general techniques, activities, and tasks in software development. Process patterns and organisational patterns are closely related. Organisational patterns describe organisational structures that are useful in software development [2]. The distinction between activity-based patterns and activity-independent patterns is sometimes difficult, for example in the case of process patterns which focus on only one activity. This leaves room for interpretation [15].

Additionally there are two more pattern types that have their roots in software development. The area of application for these patterns is outside the scope of the immediate software development process.

PT10: Educational patterns

PT11: Pattern writing patterns

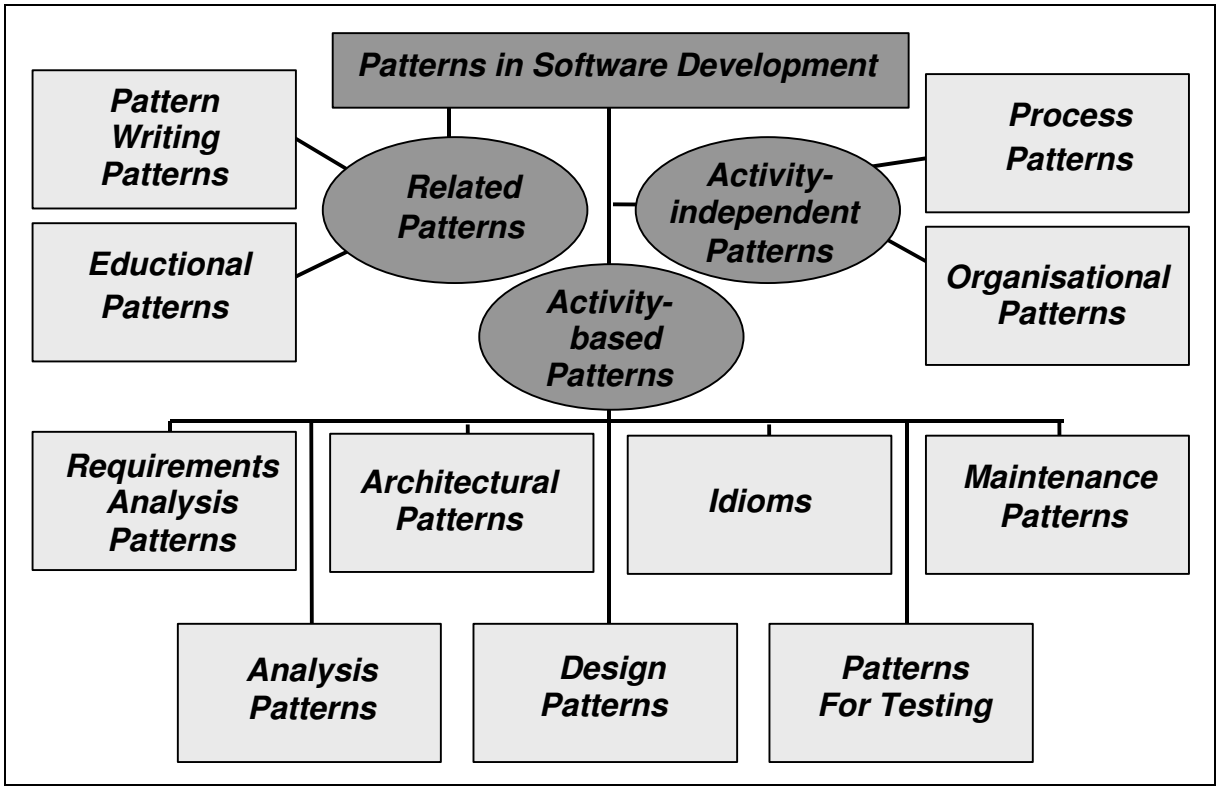

Figure 1: A classification of patterns

1.2 A Microprocess for Patterns

Patterns help to document the knowledge of experts [13], [23]. NONAKA distinguishes two types of knowledge: formal and systematic *explicit knowledge* which is easily communicated and shared, and *tacit knowledge* which is embodied in individuals. The transitions between both types are *externalisation* (from tacit knowledge to explicit knowledge) and *internalisation* (from explicit knowledge to tacit knowledge) [33].

The use of patterns will not replace existing processes or software development methods. They will only supplement existing methods [42]. In the style of BOOCH the following approach is called Microprocess for Patterns (Mp). While the macroprocess describes the general process in use, the Microprocess's focus is on the daily activities of software engineers.

Software developers, whether working individually or in teams, can take two roles within the Microprocess: the producer role and the consumer role. These two roles correspond to NONAKA'S distinction between externalisation and internalisation.

The Consumer's Viewpoint (MpCon)

The main purpose of the Mircoprocess for Pattern Consumers is to support existing activities of the macroprocess. Based on the problem at hand, matching problem-solution pairs are chosen and the most appropriate solution is selected and applied [11], [51]. This leads to the following four phases:

MpCon1: (Micro-)problem specification

MpCon2: Search for applicable patterns

MpCon3: Selection of the most appropriate solution

MpCon4: Pattern application

"Many non-experts simply don't know what problems they have"[26].

Typically these problems are related to non-functional qualities, such as maintainability. Patterns explicitly address these properties of software systems. Therefore, it makes sense to search for applicable patterns even though there is no specific problem specification. MpCon1 would be omitted.

The Producer's Viewpoint (MpPro)

There are many possibilities to develop new patterns. Mostly they are independent of the actual software development process [40]. The search for new pattern ideas can be incorporated into the Microprocess for Patterns. It consists of three phases in which the first two are iterative:

MpPro1: Document pattern idea

MpPro2: Evaluate pattern idea

MpPro3: Write pattern

"Patterns explicitly capture knowledge that experienced developers already understand implicitly" [13].

This implicit understanding makes it difficult to recognise new patterns. VLISSIDES suggests to keep track of experiences and gained insights while developing software systems [49]. This is reflected in the iterative nature of phases MpPro1 and MpPro2.

In an evaluation of a new pattern idea one should look for similar patterns as well as for other application areas of the same pattern. An evaluation of a new pattern should also pay attention to the "Rule of Three" which states that a pattern was successfully used more than once. It is only after this evaluation that the actual pattern life cycle starts. In most cases the pattern is developed independently of the software development process.

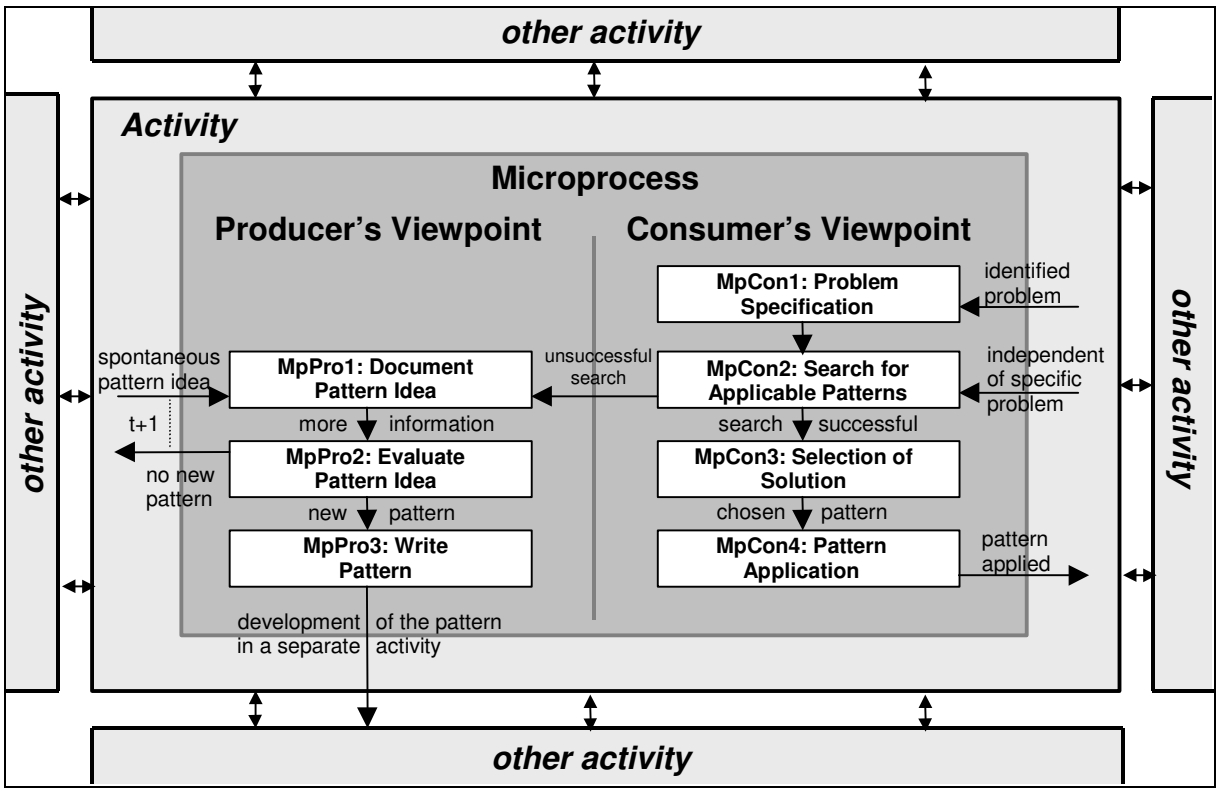

Figure 2: The Microprocess for Patterns

One open question is, when to start with the producer's and consumer's Microprocess. From a producer's viewpoint it is required to document pattern ideas immediately. The subsequent activities, evaluation and writing could be postponed till the end of a particular activity.

In the consumer's viewpoint a problem-based Microprocess, which starts with MpCon1, should begin immediately after a problem has been discovered. A Microprocess that starts independent of actual problems with MpCon2 could be initiated at the end of an activity. This would guarantee that experts, who apply patterns implicitly, are not unnecessarily disturbed by the required documentation of patterns in the activity's artefacts.

1.3 Pattern Automation and the Microprocess for Patterns

Pattern descriptions, for example formalised in the pattern form by GAMMA et al., support the evaluation of new pattern ideas (MpPro1). They also support the selection of the best solution (MpCon3) and the application of patterns (MpCon4).

Pattern organisation schemes, for example a pattern language or a pattern classification, support the search for applicable patterns (MpCon2). They also help in evaluating new pattern ideas against existing patterns (MpPro2).

Which approaches for pattern automation do exist and how do they relate to the Microprocess for Patterns? Four approaches have been identified and are briefly described below.

Aut1. Computerised Pattern Descriptions

Approaches for the automated discovery of patterns (Aut3) as well as the automated introduction of patterns (Aut4) are based on computerised pattern descriptions. The use of mark-up languages such as XML or SGML is based on the idea of formal pattern descriptions. For example for structural object-oriented patterns it is possible to define their configuration using a mark-up language. This can also include instructions on how to instantiate a pattern structure [34]. Other languages have been proposed to describe the structure of patterns, for example PaL, a language based on Eiffel, that provides additional constructs for the composition and refinement of patterns [10], [31]. Basically all these approaches are limited to design patterns.

Computerised pattern descriptions can automate the search for applicable patterns (MpCon2) or the selection of a solution (MpCon3).

Aut2. Automated Presentation of Patterns in Artefacts

The presentation of patterns in artefacts is mostly limited to the structure of object-oriented analysis patterns and design patterns. Possible solutions are the use of colours for classes that are part of the same pattern or the use of frames similar to the subject layer of the object-oriented analysis by COAD et al. [16]. Since the advent of version 1.3 the UML has defined a dotted ellipse as a means to present patterns in class diagrams. UML stereotypes can also be used to describe patterns in models.

The automated presentation of patterns in artefacts can help in documenting patterns in developed models. This approach helps to transfer additional information in between different activities.

Aut3. Automated Discovery of Patterns

Based on a computerised pattern description (Aut1), it is possible to search for patterns in existing models such as a source code. A static analysis or even a dynamic analysis at runtime can identify patterns based on a pre-defined pattern repository [9], [25]. Most approaches are limited to the source code. In general it should be possible to apply a similar approach to other models, for example design models.

An analysis of source code is based on the assumption that certain patterns result in specific compositions and inheritance relationships. By comparing these relationships to the ones of patterns described in a pattern repository, it is possible to discover existing patterns in source code. The restriction on source code implies a focus on reverse engineering tasks.

The automated discovery of patterns can help in the problem-independent search for applicable patterns. It is also possible to employ automated discovery in the search for new patterns.

Aut4. Automated Introduction of Patterns into Artefacts

Most approaches apply the automated introduction of patterns to design patterns. It is possible to put analogous methods to use for other pattern types as well. There are two basic approaches for an automated introduction of design patterns: the automated generation of design models and the automated generation of source code. BOSCH distinguishes three approaches [8]. He divides the automated generation of source code further into a generative approach in which parts of the source code are automatically generated, and an approach with special programming language constructs for patterns.

An automated generation of patterns in design models starts with assigning pattern roles to certain classes in the model. If there is no existing class for a certain role, it can be automatically generated. After that the designer selects certain options that are presented in the pattern's description. Based on these options and the selection of roles, a series of mini-transformations, such as the generation of methods or the renaming of parameters, introduces the pattern into the model [47]. The automated introduction of patterns corresponds to the pattern application phase of the Microprocess (MpCon4).

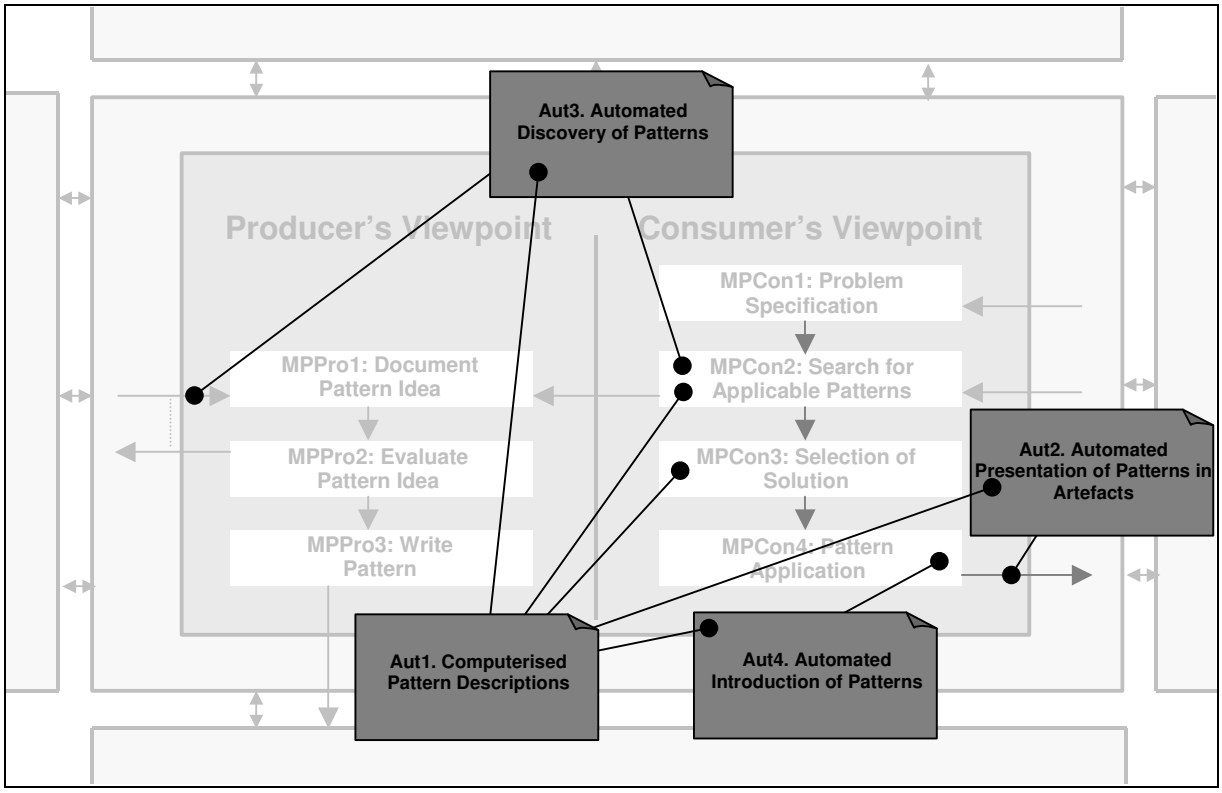

Figure 3: Pattern automation in the Microprocess for Patterns

2. General Aspects of the Survey

2.1 Areas of Interest (AoI)

The following general areas of interest have been identified. Some of them, such as the characterisation of patterns or the usage of different pattern types, are very detailed. Others, such as pattern usage and pattern mining in an organisation only scratch the surface of the subject matter.

AoI1. Understanding of Patterns

The goal of this analysis subject is to gain a common understanding of patterns as they are used in software engineering. Pragmatically, commonly identified pattern properties could be used to separate patterns from other related concepts.

AoI2. Patterns in an Organisation

This analysis subject provides a ratio between pattern consumers and pattern producers. In addition it will provide information about how patterns are introduced in an organisation, how they are discovered, and how this is encouraged, for example with incentives.

AoI3. Different Pattern Types – Usage and Tool Support

Based on the questions of this subject one can derive statements about the usage of different pattern types. The classification into four approaches for automation is being used to investigate the degree of pattern automation for different pattern types. This might be affected by the respondents' familiarity with different pattern types. An additional question measuring the number of known patterns for each type is targeted towards these familiarity aspects.

AoI4. Pattern Usage and Pattern Mining in the Process Model

Only respondents who are using a process model in their organisation are preselected for this area. Its aim is to provide an indication whether the creation and the use of patterns is part of the used process model.

Five major approaches to overcome the disadvantages of the strict waterfall model have been identified: incremental development, the use of evolutionary prototypes, the use of explorative prototypes, reuse of artefacts, and the use of object-oriented concepts. Frequent application of these approaches among pattern users can indicate that there is a linkage between the two. Because of the nature of the survey this can not be proved since this would require a comparison group of non-pattern users.

AoI5. Characterising Patterns – Pattern Descriptions and Pattern Organisation Schemes

Patterns can be characterised by using pattern descriptions and pattern organisation schemes. The use of specific ways to characterise patterns in practice is investigated in this area of interest. A special focus lies on pattern organisation schemes. The respondents are asked, whether they are satisfied with the organisation scheme in use and which relationships between patterns a newly developed scheme should place its emphasis on.

AoI6. Experience with Patterns

Many single experience reports can be found in the existing literature. To put these single experiences on a broader basis, 22 statements have been identified and respondents are asked to rate these statements according to their own experience. In addition they are asked what they see as problematic with patterns. The success of an often cited statement, that the expectations towards patterns should be kept low, is measures by a special question.

Patterns for reuse, documentation, and deployment have been identified as uncovered. Whether patterns for these areas could be developed and if there is a need for these patterns constitute part of this field of interest.

AoI7. The Value of Patterns for an Organisation

The ratio between patterns developed within the organisation and these from outside it is measured. Similar to that, respondents are asked whether they discuss patterns with people external to their organisation and share them with other organisations.

Because they describe the knowledge of experts, the use of patterns and a pattern itself can be seen as a competitive advantage. This poses two questions. How a potential market for patterns could look is addressed by a separate question.

2.2 Related Work

Besides several experience reports, mostly from single companies, there are not many studies related to the application of patterns.

One of them, of which the results have been considered for the list of advantages and disadvantages of patterns in AoI1 and AoI6, summarises the findings of six different companies [4]. For example statements as

- "Patterns improve communication",
- "Patterns are extracted from working design", and
- "Patterns summarise the essence of a design decision"

can be directly mapped to similar statements within the questionnaire. The goal of this survey is to put these results on a broader basis.

Another study, implemented as an experiment, has been done at the University of Karlsruhe. Based on different tasks performed by students, it proves two hypotheses: the documentation of design patterns in source code makes maintenance faster as well as it improves the quality of the maintenance performed [36].

A similar study at Washington University, St. Louis could only substantiate the first hypothesis. The second one, however, could not been proven right or wrong [37].

Another study strongly supported the hypothesis, that the knowledge of design patterns improves the communication within a team [48]. A similar statement is part of AoI6.

All of the above-mentioned studies are explicitly targeted towards design patterns whereas this survey also explicitly includes other types of patterns.

2.3 Design and Realisation of the Survey

The goal of this survey is to give a descriptive overview of the usage and understanding of patterns in practice. The survey's population consist of individuals who show an interest in patterns in software development. Most of them are software engineers, but managers in software projects or university employees and students are also included. Explicitly excluded are individuals who are not involved with patterns.

Such a definition guaranties that one can make statements about the areas of interest as defined in chapter 2.1. Statements that compare pattern users with non-users can not be made. This includes statements about the number of organisations that use patterns or about different properties of pattern users and non-users, for example about the usage of object-oriented concepts.

The empirical survey has been conducted as web-based. The questionnaire is a self-administered web application. This guarantees world-wide access to potential respondents.

The size of the population is large though unknown. With respect to the size and the costs of reaching each individual of the population a sample had to be defined. Based on its characteristics one can draw conclusions about the population at hand. The sample should consist of randomly chosen individuals of the population. Potential candidates are contacted by e-mail. The names and e-mail addresses have been taken from publicly available lists of participants of conferences, lectures, and seminars dedicated to patterns. In addition authors of articles and books about patterns have been contacted. As a second form, besides direct e-mails, mailing lists have been chosen.

The questionnaire has been tested in two phases. Initially three persons from different companies filled out a paper version of the questionnaire. In a succeeding interview potential problems have been identified. Afterwards the web version of the questionnaire has been developed. This has been tested with three other persons under realistic conditions. While filling out the questionnaire the respondents were under observation. They were asked to state potential problems and misunderstandings aloud. Afterwards these problems were discussed and ambiguities in the questionnaire were removed.

The questionnaire was written in English. Although none of the test persons were native speakers they did not have any problems in understanding the questions. All test persons were experienced software developers who were also familiar with the use of patterns. Based on these pre-tests an average time of 20 minutes for completing the questionnaire was calculated. This time is highly dependent on the motivation of the participants.

With respect to the distance of interviewer and respondent, web-based surveys are similar to self-administered mail surveys, especially compared to telephone surveys. Low costs and the possibility to reach participants in different geographic areas speak in favour of the use of web-based surveys [17]. Directly compared to mail surveys, they are an ideal technology for international studies [20]. These reasons were the most important for choosing the approach described above. On the other hand the approach used here also implies some specific problems which are analysed according to four error categories:

- Coverage Error
- Sampling Error
- Non-response Error
- Measurement Error

Coverage Error

"The Internet user population is clearly not representative of the population at large"[21].

This can lead to a preselection of participants that does not represent the population. This can cause a skewing of the data or even render it completely useless. In this survey this is less acute since one can assume that nearly 100 percent of the population at hand have Internet access. Two more possible sources of errors have been identified by BALLANTYNE: the degree of computer literacy and the acceptance of using a computer to fill out a questionnaire [3]. Again both sources of errors are less significant in the case of the defined population.

Sampling Error

As in the case of mail surveys the selection of the sample is highly important for the significance of the results. It should be a random selection of individuals of the population. This has been achieved by the selection process for e-mail addresses and mailing lists as described above.

"It is noteworthy that newsgroups and discussion lists can assist researchers in generating a screened sample. Internet users that participate in newsgroups and discussion lists do have similar interests …" [21].

Furthermore the quality of a sample is affected by the number of non-responses.

Non-response Error

A high number of non-responses limits the significance of the results. The characteristics of the respondents can completely differ from the characteristics of those who decided not to answer the questionnaire. This is a common problem of self-selection [21], [30].

Measurement Error

The appearance of the questionnaire on the participant's screen depends on the screen size and other selected options of the web browser used. Tips for the design of Internet questionnaires are given by DILLMAN in [18], [19].

Another possibility for measurement errors comes from the fact that English is not the mother tongue of the majority of the respondents. Different cultural backgrounds have an influence on the understanding of the terms used:

"The internet is global; hence differences in language, slang and culture may lead to the misinterpretation of questions or even offend the respondents"[21].

To avoid this, the questionnaire has been tested by software engineers of different backgrounds.

3. Summary of Results

3.1 General Information

An initial contact e-mail was sent on both 5 and 6 September 2001. A reminder was sent on 20 September. The sample consisted of 759 individuals. The number of undeliverable e-mails was 184, mainly caused by invalid e-mail addresses. The number of persons reached calculates to 575. Similar to letters one can not guarantee that an e-mail has been received. Only 76 percent of those we attempted to reach had been contacted.

Parallel to direct e-mails, a link to the questionnaire was posted on several mailing lists. The subject of all mailing lists is related to the use of patterns. An initial e-mail to 15 mailing lists was sent on 18 September 2001 and a reminder was sent on 9 October. The survey was closed on 21 October.

Altogether 200 respondents filled out the questionnaire. For 108 respondents a direct e-mail was the initial contact with this survey. Based on this, one can calculate a response rate of 19 percent. Because many respondents subscribed to several contacted mailing lists, a response rate for this means of contact can not be calculated.

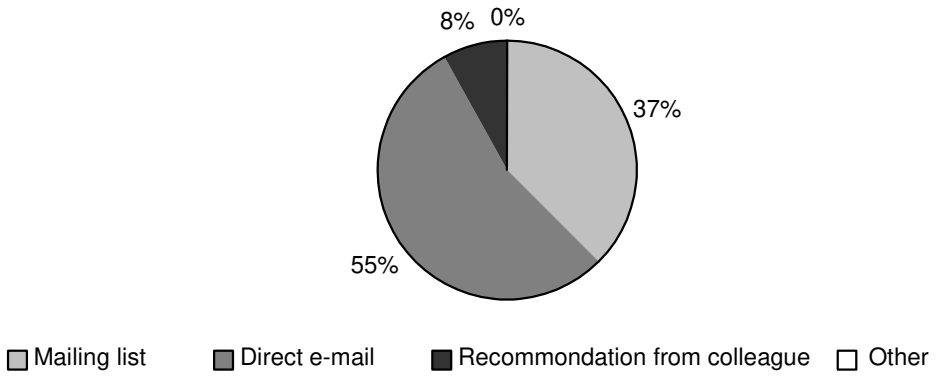

Figure 4: Initial contact with respondents (Question 8D)

Individuals, who did not fill out the questionnaire, gave the following reasons.

- No time (Six entries)
- Only occasional use of patterns (One entry)

- Participant does not use patterns anymore (One entry)
- Lack of knowledge to respond to questions (One entry)
- General resentment towards questionnaires (Two entries)
- Filling out the questionnaire is too boring (One entry)
- Initial creation of a login is too difficult (One entry)
- The link from the e-mail did not work, timeout problems (Five entries)
- Long response times of web pages (One entry)

The last two items can be explained as technical difficulties with the servers and the instability of the Internet connection to the servers.

Some of the persons contacted indicated that they could not fill out the questionnaire, so instead they forwarded the invitation to others. Sixteen respondents stated that they filled it out after they received an invitation from their colleague.

Much more then half the respondents work in a company. About one fourth of the respondents work or study at a university. The prevalent answer in the category “other” was to work at a research centre (five responses). Besides that, this category was chosen by people who could not identify with one of the pre-defined categories, for example students who work for a company.

Figure 5: Type of organisation of respondents (Question 8C)

3.2 Significance of Results

Most of the following analysis is based on the examination of a single trait. Each member of the population can be classed as either having or failing to have that trait. Thus, as an example, a statistical analysis of question 1B is presented. This question only corresponds to one parameter – the percentage of individuals who consider algorithms as patterns. Other questions lead to more than one trait, for example question 1A leads to 15 parameters – one for each pattern quality.

To what degree is it possible to extend statements about the sample to the population at hand? In statistics one uses the concept of confidence intervals. The following statements are based on a 95 percent confidence interval ($\alpha = 0.95$). This means that a given statement about the sample also applies with a probability of 95 percent to the population. The size of the confidence interval can be calculated based on the following formula:

$$d = \frac{c \bullet \sqrt{\hat{p} \bullet (1 - \hat{p})}}{\sqrt{n}}$$

$$[\hat{p} - d; \hat{p} + d] = confidence\ interval\ \ for\ \ p$$
$$c = (1 - a/2) - percentile\ of\ \ the\ standard\ \ normal\ \ distribution$$
$$n = sample\ \ size$$
$$\hat{p} = sample\ \ proportion$$

In the case of question 1B, 24 percent of the respondents consider algorithms as patterns. In three of the questionnaires this question has been left blank. This reduces the sample size to 197. For large samples the percentile used can be taken from the standard normal distribution. For a 95 percent confidence interval one gets a percentile c of 1.96.

$$d = \frac{1{,}96 \bullet \sqrt{0{,}24 \bullet (1 - 0{,}24)}}{\sqrt{197}} = 0{,}060$$

$$\textit{confidence interval for proportion } p : [0{,}24 - 0{,}060 ; 0{,}24 + 0{,}060]$$

Based on the collected information, the following statement can be made: We can be approximately 95 percent confident that the true proportion p of software engineers who think that algorithms are patterns, lies between 18 and 30 percent.

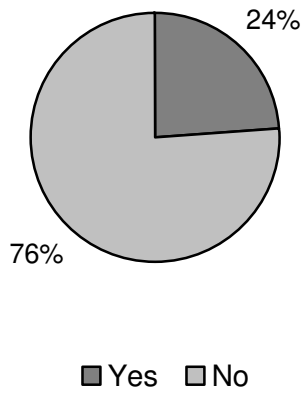

Figure 6: Do you consider algorithms as patterns (Question 1B)?

As seen from the formula, the size of the confidence interval depends on the proportion measured as well as on the size of the sample. In a worst-case scenario, in which the sample proportion is 50 percent, the confidence interval would increase to +/- 0.070.

The sample size is reduced by the number of non-response items. Their actual number among the respondents varies between 1 and 26. Calculatively, this has only a negligible effect on the size of the confidence interval.

As already mentioned in chapter 3.3 a high number of non-response items affects the interpretation of the results in that, they could indicate that the respondents did not understand the question, did not have enough knowledge to answer the question, or simply did not want to respond to a particular question. In the case of an exceptional high number of non-response items, this will be noted in the analysis of the question (greater than five percent or ten non-responses) [17].

3.3 Areas of Interest

3.3.1 Understanding of Patterns (AoI1)

The goal of this area is to reach a common understanding of patterns. In a pragmatic approach, one can say that ten of the proposed fifteen qualities characterise patterns. At least two thirds of the respondents agree on these qualities. Based on this, the following characterisation can be derived:

Characteristics Based on Pattern Origins

A pattern occurs again and again (Q1) and is successfully used more than once (Q4). It captures the knowledge of experts (Q2). Patterns are independent of object-oriented concepts (Q6).

Characteristics Based on Patterns Descriptions

A pattern describes a problem-solution pair (Q10) as well as its context and the forces which limit its solution (Q11). A pattern is a sample, template, and model (Q9).

Characteristics Based on Pattern Application

A pattern can be applied to many actual problems (Q14), it is modified when used, but keeps its general form (Q13). It unifies the communication of experts (Q15).

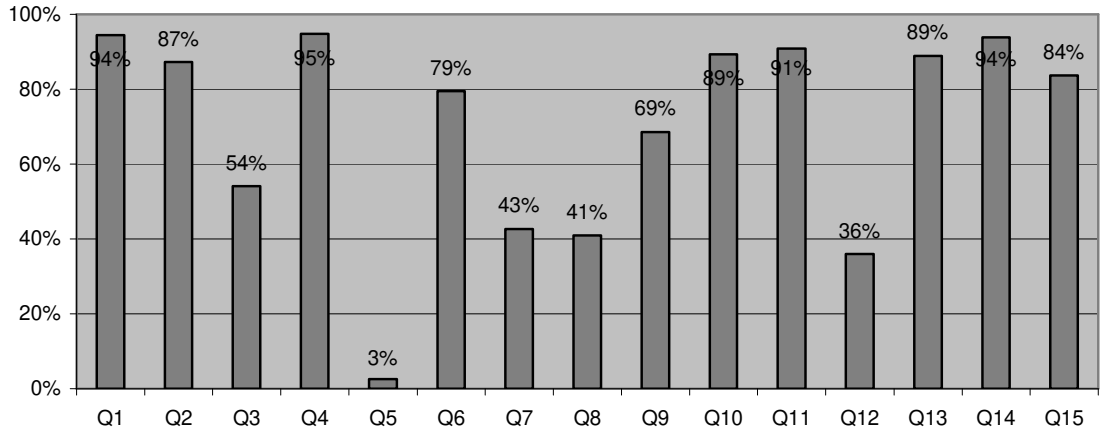

Figure 7: Which qualities characterise the term "pattern" (Question 1A)?[1]

The two qualities Q10 and Q11 have been mapped according to the general pattern definition by ALEXANDER, which is also used in this paper. About 90 percent of the respondents agree with these two qualities and therefore with ALEXANDER'S definition.

As already analysed in the previous chapter, 24 percent of the respondents see algorithms as patterns, while 76 percent do not. Clearly, this has an effect on the notion of the term "pattern" as it is used in the remaining questions of the survey. This different view on patterns should be considered in an interpretation of the results presented here.

3.3.2 Patterns in an Organisation (AoI2)

95 percent of the respondents use patterns. This high rate is based on the definition of the population as all "individuals who show an interest in patterns in software development." Another reason could be the generality of the term pattern.

Contrary to the high rate of pattern users among the respondents, only 78 percent of them use patterns also in an organisational context, for example for communication in software development teams or in documentation. Although organisational aspects, as improved communication are the main advantages of patterns [4], [24], [44] 22 percent of the respondents can not gain from these aspects within their organisation.

[1] A list of all pattern qualities (Q1-Q15) can be found in Appendix I.

55 percent of the respondents write or discover new patterns in their respective organisation. This number seems to be high and is probably influenced by the way the respondents have been chosen.

The questions in AoI2 aim to clarify the use of patterns in the respondents' organisations. This analysis of their answers is based on individual views of each organisation's members and not on direct observations in each organisation.

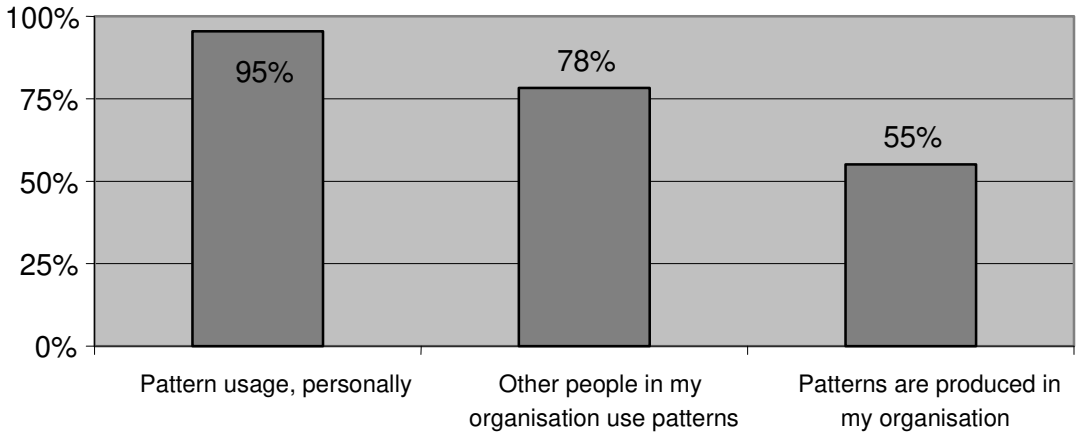

Figure 8: Percentage of pattern consumers and producers (Questions 2A, 2B)

Which methods do software developers use to learn patterns and how do they discover new ones? Nearly all respondents learn patterns. Studying on one's own is the most popular method which is used by 92 percent of them.

The prevalent answer in the "other" category was the informal means of communication (30 entries). Among them were individual mentoring, spontaneous discussions about patterns, and learning-by-doing. The latter can be triggered by the requirement to use patterns within a project team. Sometimes these informal methods were supported by internal mailing lists. Further entries were:

- Attendance of pattern conferences (seven entries)
- Learning of patterns in system reviews (four entries)

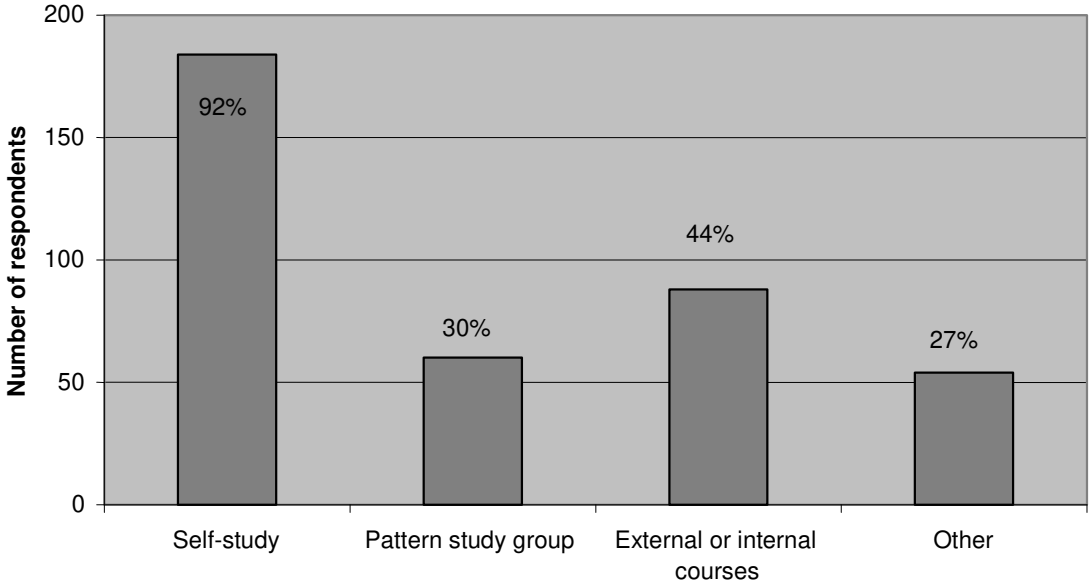

Figure 9: How do you introduce patterns in your organisation (Question 2C, multiple choice)?

The production of patterns, respectively the search for new pattern ideas, their evaluation, and their development, has been analysed by question 2D. 87 percent of the respondents use at least one of the possible means to find new patterns. Compared to the 55 percent of the respondents who use patterns within their organisation, this seems to be high. Even under the assumption that all non-respondents for this question do not discover new patterns, this number only decreases by three percentage points. While question 2B asked whether patterns are written within the organisation, question 2D asked whether new patterns are discovered. One would expect similar figures for both questions. The reason for this difference remains open.

Most of the software developers, who show an interest in patterns, discover them as part of their daily work. While this method indicates an informal procedure, the others are more formal. Expert interviews, external pattern workshops, and system reviews remain less often used. As already found in the analysis of open-ended answers of question 2C, most entries in the "other" category were informal methods (16 entries). This includes discussions with other pattern authors, occasional meetings, and simply the analysis of artefacts. Three entries, all of them from respondents of universities, explicitly mention the search for new patterns as part of their research work.

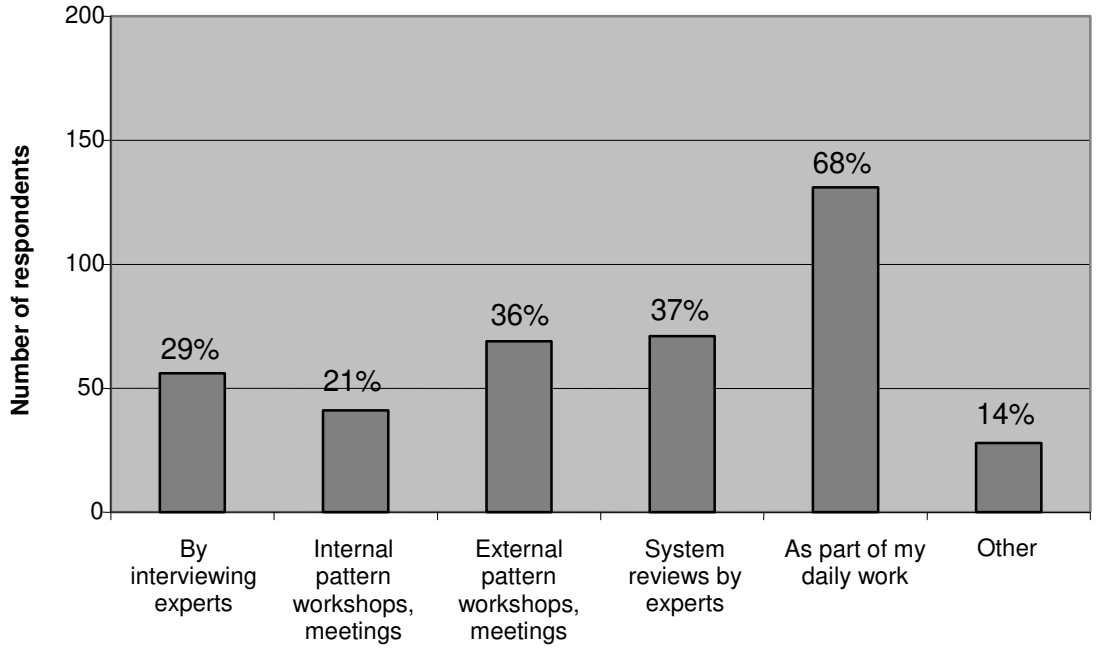

Figure 10: How do you discover new ideas for patterns (Question 2D, multiple choice)?

Questions 2E and 2F aim at different methods to support pattern use and pattern production. 46 percent of the respondents state that the use of patterns is supported by their organisation. If one restricts this analysis to individual pattern users only, this number does not change at all because nearly all respondents use them individually.

Expectedly, this rate increases if this analysis is restricted solely to organisational users (sample size = 158). Only in 55 percent of the cases, the use of patterns is promoted by the organisation. This means that 45 percent of the respondents use them in their organisation although this is not explicitly promoted.

Only in 25 percent of the respondents' organisations, the production of new patterns is promoted in one way or another. Restricting the analysis to respondents who actually write new patterns, increases this number to 39 percent. The search for new patterns in an organisation is less often promoted than the use of them.

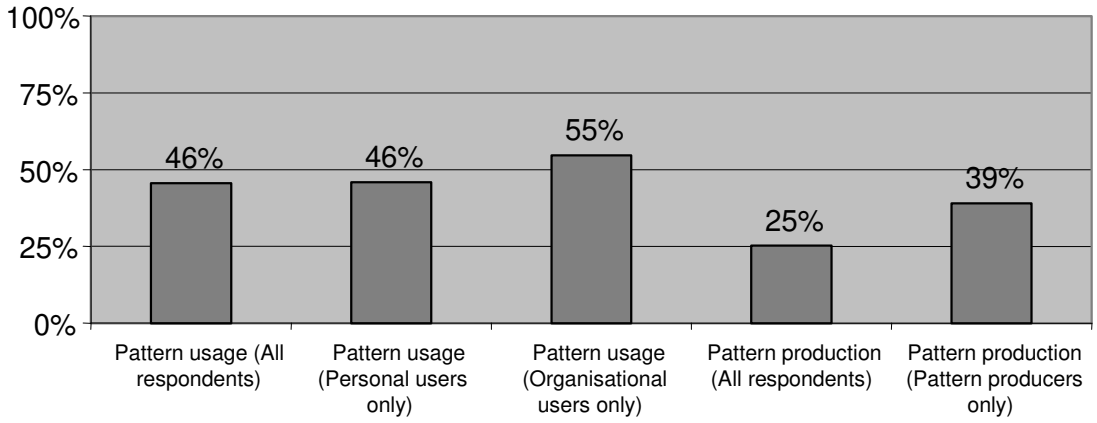

Figure 11: Organisational support for the use and production of patterns (Questions 2E, 2F)

Especially interesting are the open-ended responses since both questions (2E and 2F) were designed without a categorisation of possible answers. The use of patterns as an absolute necessity was mentioned seven times, i.e. there is some unofficial pressure to use patterns in the organisation. This pressure is created by recommendations in system reviews or informal discussions among software engineers. 18 respondents mentioned that the use of patterns was supported by promoting different means of learning new ones (see Figure 9). Both formal support within the software process as well as separate "pattern" teams were each mentioned five times. These teams promote the pattern concept within an organisation. Three respondents mentioned a leader who encouraged the use of patterns: "The big boss loves patterns." Publications about patterns in the organisation's intranet were also focused on (six entries). Bonuses for pattern use were not brought up though one respondent did allude to the possibility of awards: "Awards have recognised early success in patterns usage."

Pattern producers have not received any bonuses at all, nor have they benefited from non-monetary rewards such as advancement opportunities or flexible work schedules. The publication of results from research activities was mentioned by five respondents, three of whom work at a university. One of them is part of an academic research project in cooperation with a company. Furthermore, the publication and discussion in the organisation's intranet (four entries) and the support by special coaches (three entries) was cited. Seven responses were related to the different forms of finding new patterns as was discussed previously (see Figure 10).

3.3.3 Different Pattern Types – Usage and Tool Support (AoI3)

Similar to question 2A, the analysis of the application frequency of various pattern types is also divided in order to treat both individual and organisational usage. For the individual use of process patterns, educational patterns, and pattern writing patterns, the number of non-response items lies between 11 and 13. The amount of non-response items for the use of patterns within an organisation is between 17 and 26. This seems to be high. The assumption that those respondents who indicated in question 2A that they do not use patterns for communication within their organisation must have skipped the respective part of question 3A can not be supported.

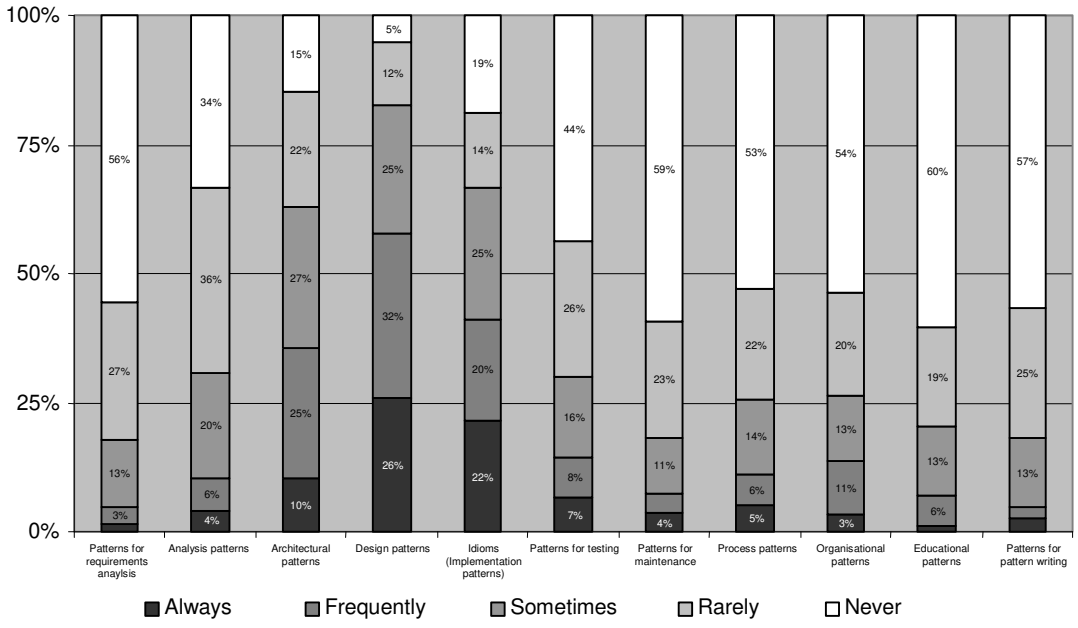

Figure 12: Frequency of individual pattern use (Question 3A, all respondents)

Since the application frequency has been categorised, pattern types can not be ranked according to it. Nonetheless it is possible to see that analysis patterns, architectural patterns, design patterns, idioms, and patterns for testing are more often used than the other types. More than 50 percent of the respondents employ these patterns occasionally at least. Of all pattern types, design patterns are the most often used. This seems to be reasonable since the roots of patterns are based on the development of design patterns [14]. On the other hand this might have been caused by a high number of software developers in the sample who do design activities and therefore use design patterns.

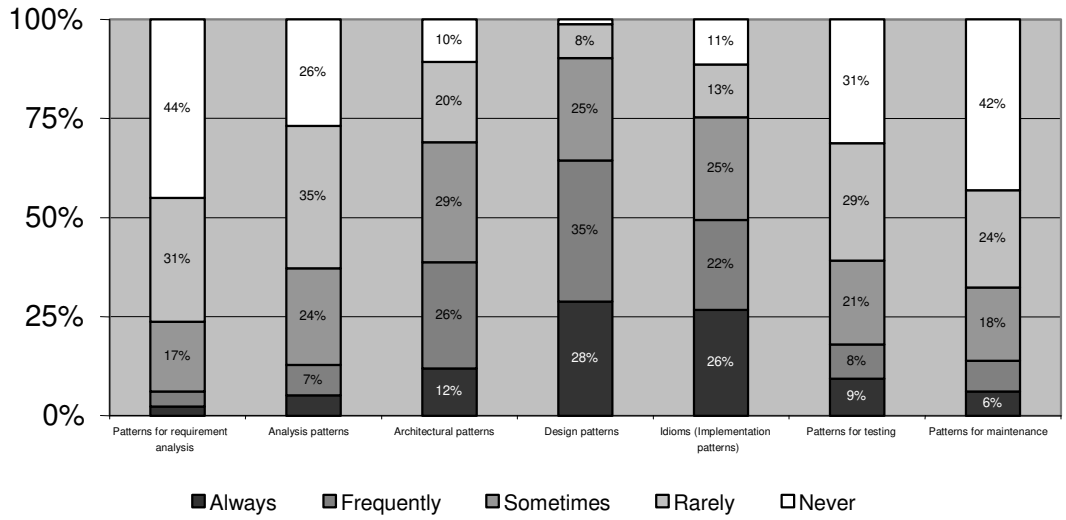

Figure 13: Frequency of individual use of patterns (Questions 3A, 8A, pre-selection of sample)[1]

For activity-based patterns this can be analysed further. Question 8A asks whether the respondent is involved in a certain activity or not. According to the answers to this question, preselecting a sample of those respondents that also do this activity on which a pattern type is based, can prevent any possibility of the resulting data being skewed. For example while according to Figure 12, 5 percent of the respondents do not use design patterns at all, this number is reduced to one percent, if the sample is restricted to software designers. This difference is about the same for analysis patterns, architectural patterns, design patterns and idioms. For requirement patterns, patterns for testing, and patterns for maintenance these differences are more significant – the actual numbers vary between 12 and 17 percent.

How does the frequency of the individual use of patterns compare to the use of different pattern types within the respondents' organisation? In contrast to the other pattern types, patterns for requirements analysis, patterns for maintenance, process patterns, organisational patterns, educational patters and patterns for pattern writing are less commonly used.

[1] The sample on which this chart is based on has been preselected based on question 8A. Only respondents, who actually do the respective underlying activity of a pattern type have been included in the sample. This reduces the sample size. From left to right the sample sizes are (non-response items not included): 134, 161, 173, 167, 155, 130, 66.

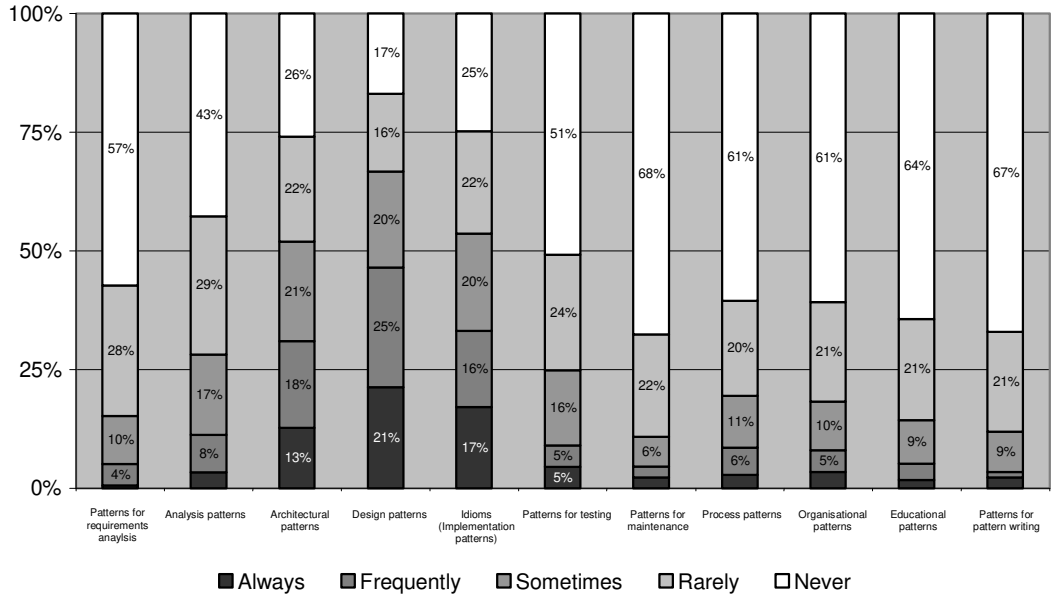

Figure 14: Usage frequency of patterns in an organisation (Question 3A, all respondents)

The usage frequency of different pattern types in an organisation, for example for documentation or for communication, shows about the same distribution as the individual use. In general, the use of pattern types, is less frequent. This corresponds to the responses to question 2A in which a similar observation can be made (see Figure 8).

The familiarity with different pattern types has been measured as the number of familiar patterns in a respective pattern category. This is not necessarily a good indicator since the number of available patterns varies for each pattern type. Moreover, the degree of familiarity can differ among the respondents. Before answering question 3B, the following definition of familiarity was given to the respondents:

"In terms of this study, you are familiar with a pattern if you know the problem it describes and have a general idea of its solution."

As in the analysis of the application frequency, respondents are more familiar with analysis patterns, architectural patterns, design patterns, idioms and test patterns. More than 50 percent of the respondents know at least one pattern of these types. Although less than 50 percent actually use process patterns, organisational patterns and pattern writing patterns, slightly more respondents know at least one pattern of these particular types.

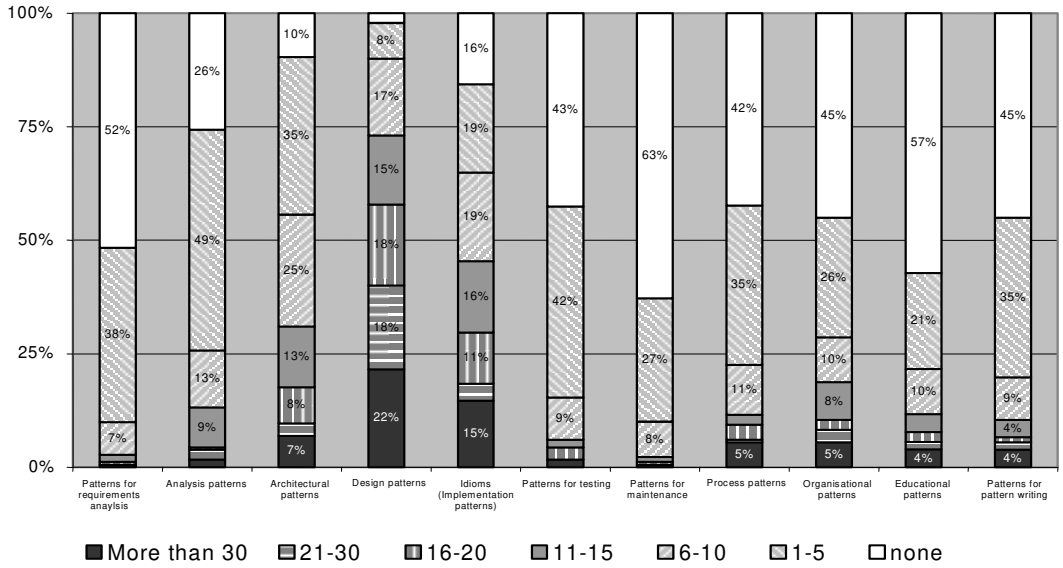

Figure 15: Familiarity with patterns (Question 3B, all respondents)

As already done in the analysis of pattern usage for activity-based patterns, the sample can be restricted to those respondents who actually do the corresponding activity.

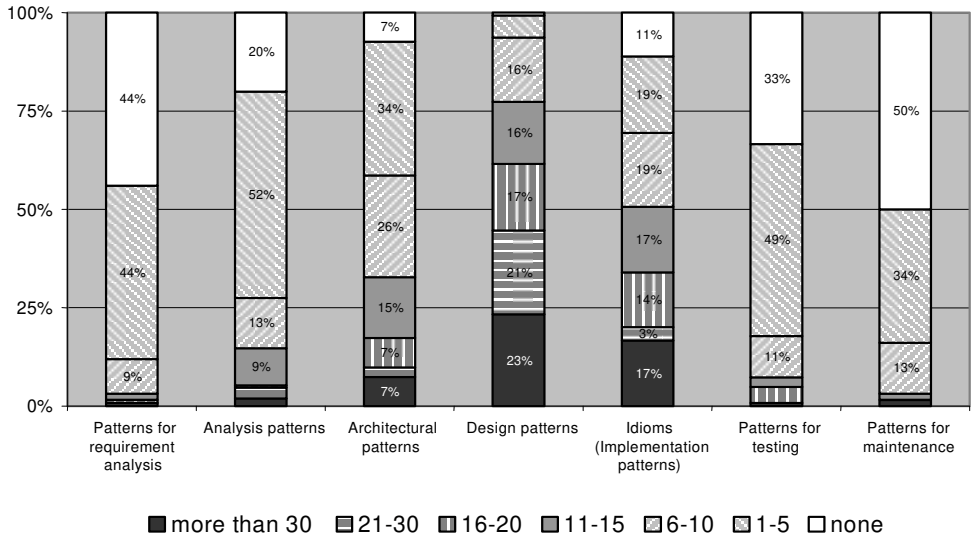

Figure 16: Familiarity with patterns (Questions 3A, 8A, preselection of sample)[1]

[1] For each pattern type a preselection based on the results of question 8A has been made. Only those respondents who actually do the respective activity are included in the sample. This reduces the sample size which from left to right is (number of non-respondents in brackets): 125 (9), 149 (12), 162 (11), 159 (8), 144 (11), 123 (7), 62 (4).

In contrast to a similar analysis done previously for the application frequency of patterns, a restriction of the sample does not cause significant differences. Only with patterns for testing as well as for maintenance, the percentage of those who know at least one pattern increases by 10 and 13 percent respectively.

Chapter 1.3 introduced four different concepts for pattern automation. How are these automation approaches used in practice? First it can be seen that in general they are used for all different pattern types.

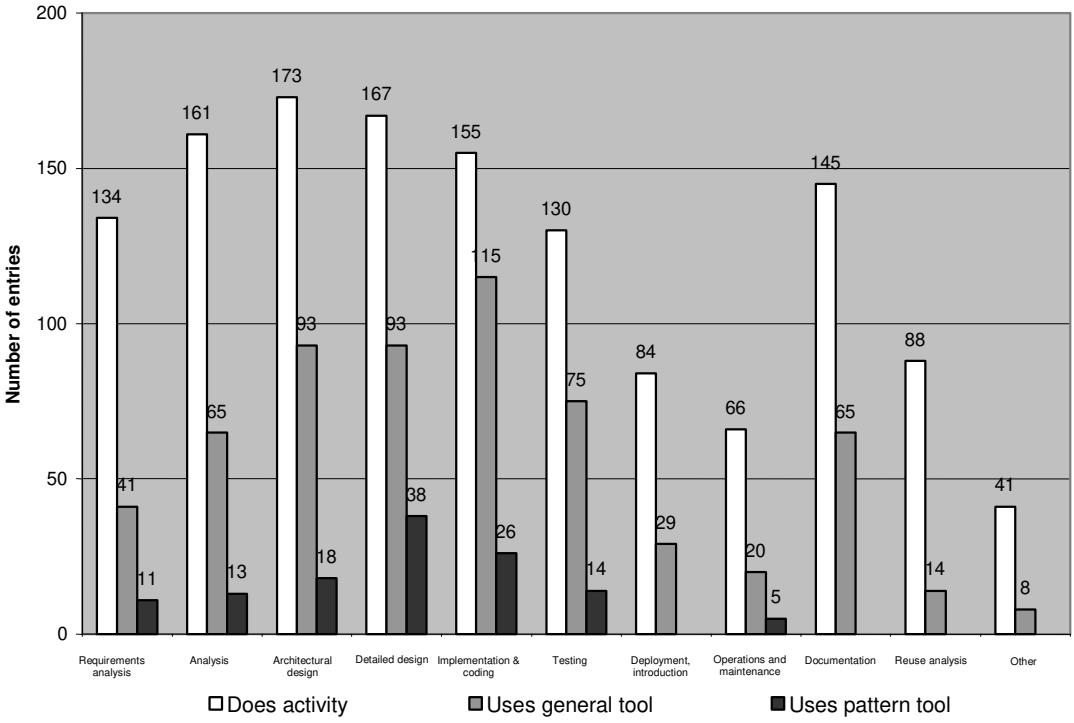

Figure 17: Number of pattern tool users (Question 3C)[1,2,3]

[1] No patterns are known for deployment, documentation, and reuse analysis. Therefore, no pattern automation approaches can be applied. The respective value in the chart is set to zero. This applies also for the category "other" although process patterns and organisational patterns could be connected to management activities which are frequently mentioned in this category.

[2] For an analysis of general tool support only those respondents that actually do the activity have been included. Similarly only those respondents that use a general tool in a category have been included in the analysis of those who use a pattern-specific tool. The sample size for the trait "uses general tool" is the number of respondents who do the respective activity. The sample size for the trait "uses pattern tool" is the number of respondents who do the respective activity and use a general tool. The respective values are shown in the chart.

[3] The numbers of non-response items for the trait "uses general tool" are (from left to right): 8, 6, 11, 11, 6, 7, 5, 4, 16, 2, 8. The number of non-response items for the trait "uses pattern tool" are (from left to right; for the positions marked with X no sample has been taken): 1, 5, 9, 9, 15, 4, X, 1, X, X, X.

Most widespread is the use of pattern automation approaches in design and implementation: For design, 23 percent of the respondents and for implementation, 17 percent of the respondents who do the respective activity also use at least one pattern automation approach. With the exception of design and implementation patterns, automation approaches are only rarely used. 41 percent of those who use a general tool for the design also employ a pattern automation approach. For other activities this percentage is lower and lies between 19 and 27 percent.

Figure 18 divides the use of pattern automation tools into the four subgroups proposed in chapter 1.3. The automated discovery of patterns plays only a minor role in practice (no more than four entries). The automated introduction of patterns concentrates on design patterns and idioms. More commonly used are computerised pattern descriptions and the automated presentation of patterns in models. Computerised pattern descriptions are mostly used for activity-based patterns. 34 respondents claimed to use an automated tool for the presentation of design patterns making it the most frequently applied pattern automation approach. For analysis patterns, architectural patterns, and idioms this approach is also the one most commonly used.

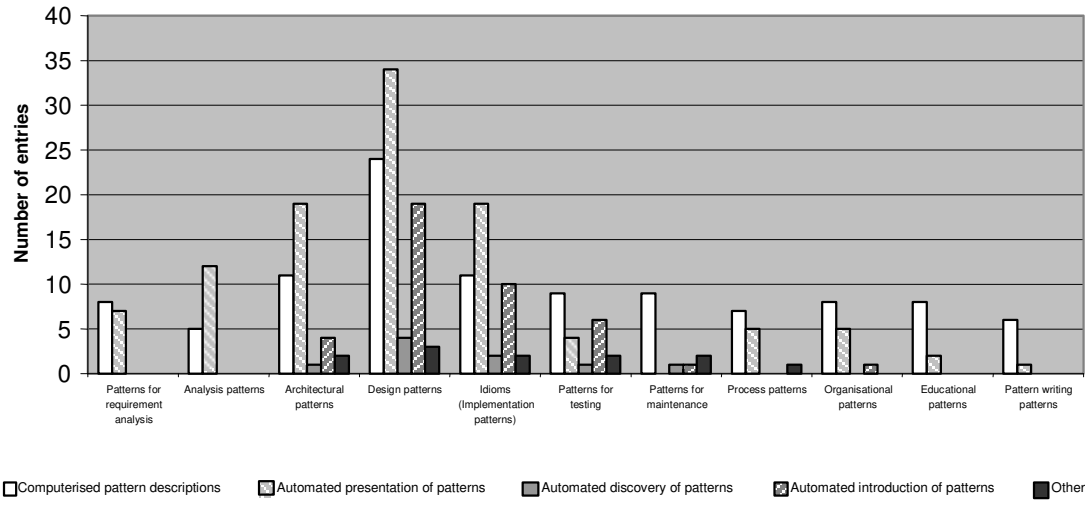

Figure 18: Number of respondents using the different types of pattern automation (Question 3C, see also Figure 17, multiple choice)

The following approaches were cited in the category "other". For architectural patterns, language support and CVOPS were given. How CVOPS, a virtual operating system for the development of protocols, supports the use of patterns has yet to be determined [50]. In the design patterns category language support, CVOPS, and Together Control Center [47] were mentioned. The non-categorised automation approaches for idioms were CVOPS and the support by the programming environment. XUnit, JUnit, and general tool support were cited as automation approaches of patterns for testing. XUnit is a framework for software testing. In the documentation of this framework, a pattern-based approach has been used [5]. For maintenance patterns two approaches were included: language support and special maintenance software. One entry for process patterns even mentioned its own development.

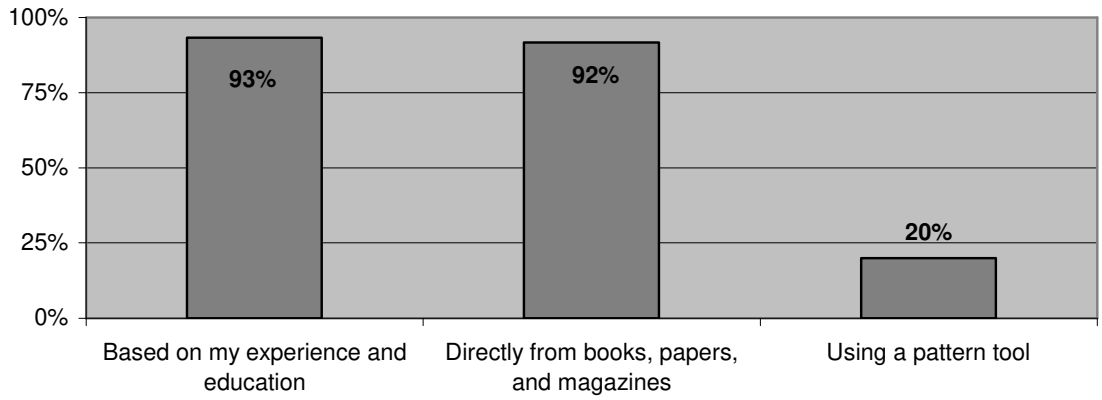

Figure 19: How do you apply patterns (Question 3D)?

The results from question 3D, depicted in Figure 19, verify that pattern automation approaches are not commonly used. Instead patterns are employed in a more informal way based on experience and education or directly from books, papers, and magazines.

3.3.4 Pattern Usage and Pattern Mining in the Process Model (AoI4)

As shown in Figure 20 only 63 percent of the respondents that use patterns in their organisation also work with a process model. By restricting the sample to respondents who work in a company this number increases to 74 percent (sample size: 98). The results from question 4A are given as a preselection for the remaining questions in this area. Only those respondents who use a process model in their organisation are included in the following analysis of questions 4B, 4C and 4D. The respective sample size is 93.

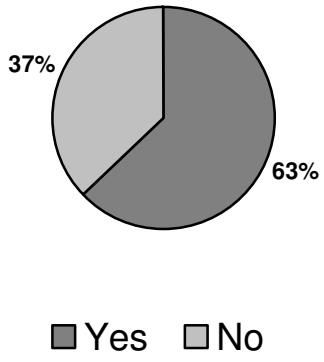

Figure 20: Do you use a process model (Question 4A, 2A)?[1]

Figure 21 analyses five different approaches to avoid the common drawbacks of the original waterfall model. Incremental development as well as object-oriented concepts are commonly part of the process model used in the respondents' organisations. The assumption that this is related to the application of patterns can be made but can not be proven by this survey. Patterns can support incremental development in so far as they simplify the use of artefacts from previous increments. The high number of users of object-oriented concepts could be caused by the dominant position of object-oriented patterns. Explorative prototypes are less often supported. Only in 62 percent of the organisations is this approach included in the process model. As already mentioned in chapter 2.1, this can only be seen as an indication of the actual circumstances.

[1] Based on question 2A, only those respondents who used patterns in their organisation were accepted in the sample. The sample size decreased to 148.

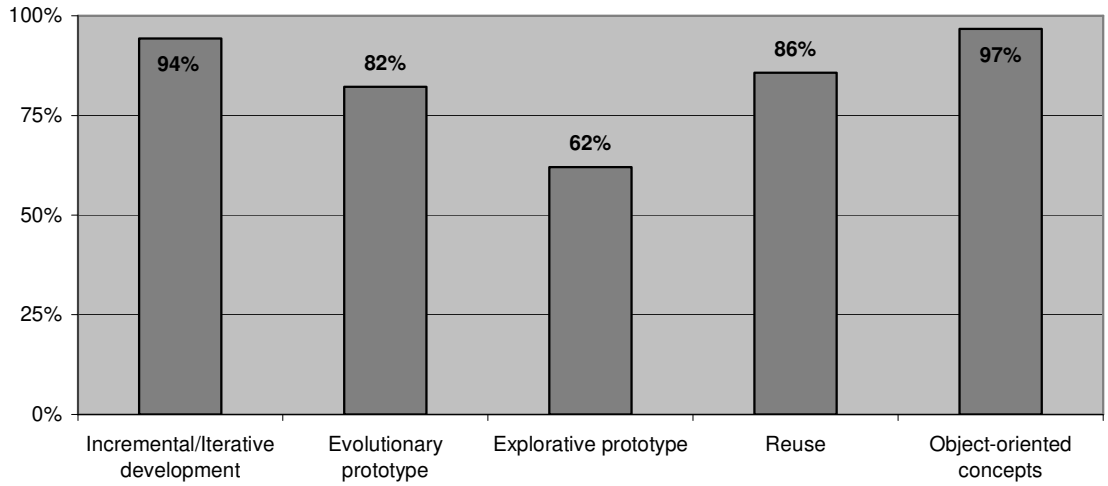

Figure 21: Usage rate of different concepts (Questions 4B, 4A, 2A multiple choice)

Is the use of patterns and the search for new ones part of the process model of the respondents' organisations? For only 38 percent was the use of patterns an explicit part of the process model. Compared to this, the search for new patterns, for example as part of a system review, was part of the process model for only 16 percent of the respondents.

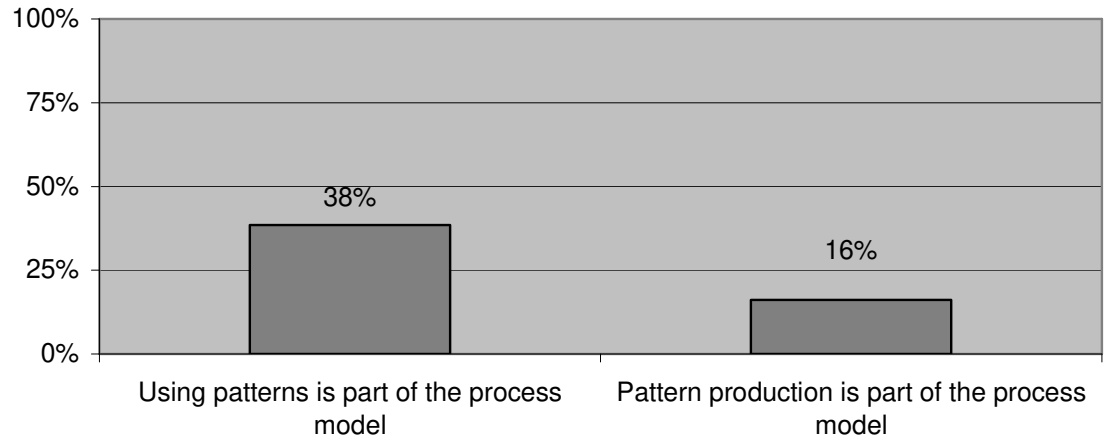

Figure 22: Is pattern usage and pattern production part of the organisation's process model (Questions 4C, 4D, 4A, 2A)?

3.3.5 Characterising Patterns – Descriptions and Organisation Schemes (AoI5)

Figure 23 depicts the usage of different pattern description forms. 89 percent of the respondents do use at least one of the specified pattern description forms. The reason for this high percentage is probably because describing a pattern in a specific and consistent way simplifies its study and usage. Both formal description forms are the most commonly used. Informal description forms such as the Portland form and ALEXANDER'S pattern form are less often used. The most prevalent entry in the category "other" is the POSA form (seven entries). This formal description form was developed by BUSCHMANN et al and is presented in [11]. An adaptation of an existing pattern description form was mentioned six times. Five respondents referred to company-specific or project-specific pattern forms. Three respondents explicitly emphasised, that they always use the most appropriate of various pattern forms. Furthermore, the following pattern forms were mentioned: BINDER'S Test Design Pattern Template [6] and the pattern description form by MOWBRAY and MALVEAUX [32]. Both are formal description forms.

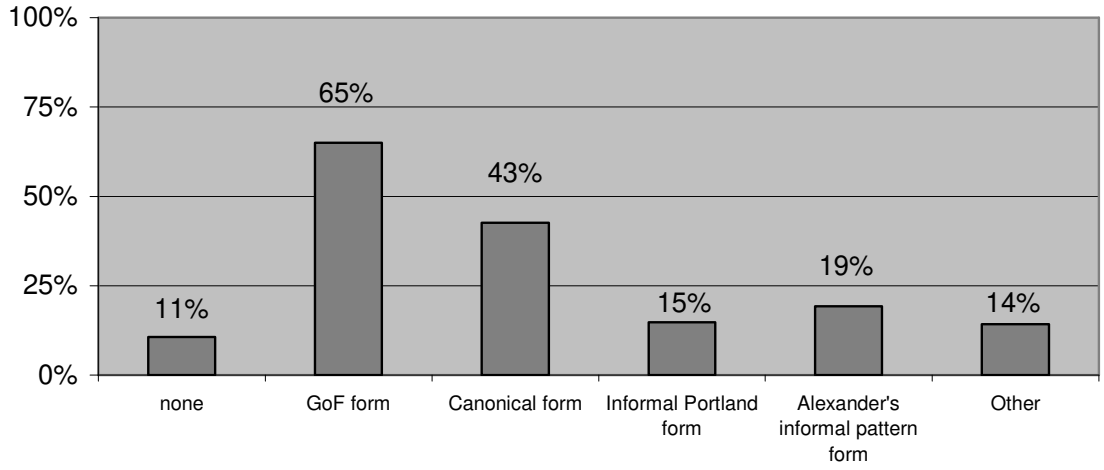

Figure 23: Usage of different pattern description forms (Question 5A, multiple choice)

While question 5A asked for different pattern description forms, question 5B related to pattern organisation schemes. 35 percent of the respondents did not use any pattern organisation scheme (see Figure 24). The schemes proposed by GAMMA et al and BUSCHMANN et al were being used by 45 percent and 31 percent of the respondents respectively. Other schemes did not play an important role. The most prevalent answer in this category was approaches that correlate patterns of a certain application domain (seven

entries). Furthermore, the following approaches were cited: a hierarchy-based classification based on multiple descriptive parameters, MOWBRAY'S and MALVEAUX'S classification [32], RISING'S pattern almanac [39], and an organisation-specific solution.

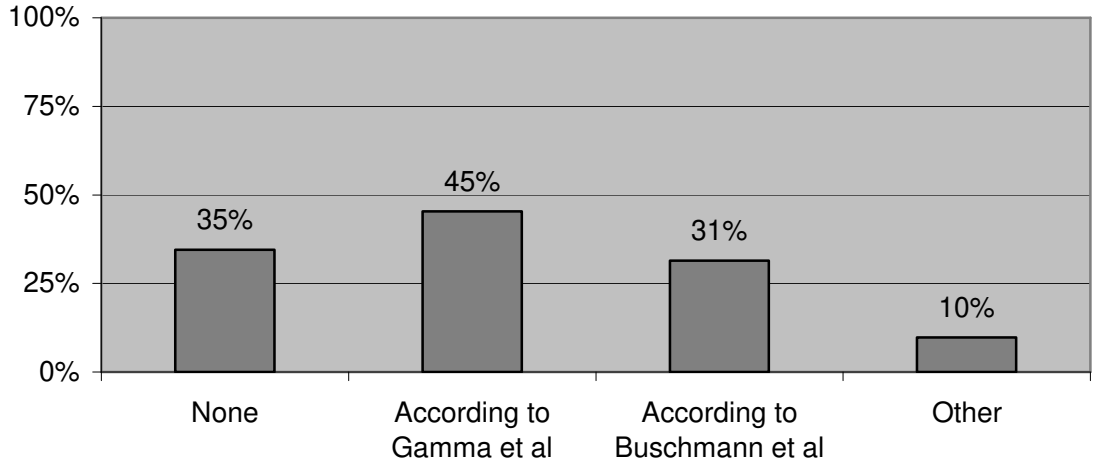

Figure 24: Usage of different pattern organisation schemes (Question 5B, multiple choice)

About two-thirds of the respondents were satisfied with the pattern organisation scheme they used. This also means that one-third was not satisfied, which leaves room for further research on possible organisation schemes.

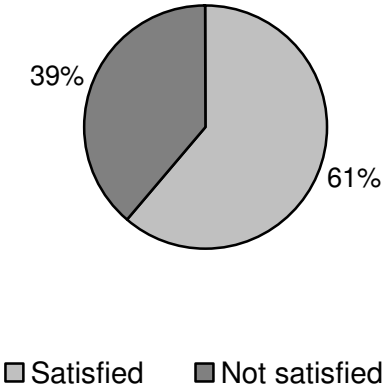

Figure 25: Is the pattern organisation scheme used a suitable means for pattern retrieval (Question 5C)?

Which approach should be followed for the development of a new pattern organisation scheme? Questions 5D and 5E attempt to clarify this. Most respondents see a pattern catalogue and a pattern language as the most appropriate approaches with 43 percent and 32 percent respectively. Other proposals were: a meta-variable based approach, a classification hierarchy, a multidimensional web of patterns, a simple text-based search, a combination of pattern catalogue and pattern language, and a classification scheme. Three responses explicitly mentioned that each approach has its benefits and drawbacks.

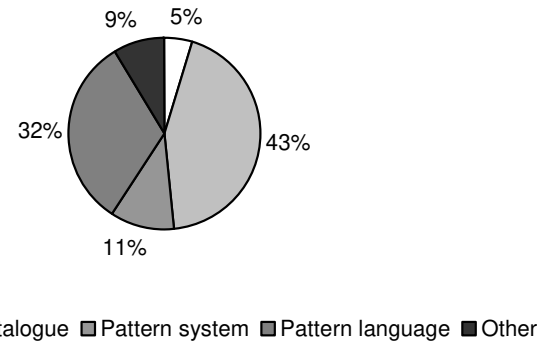

Figure 26: Which is the preferred approach for pattern organisation schemes (Question 5D)?[1]

To get a more detailed picture of relationships that should be mapped by a pattern organisation scheme, question 5E queries the importance of different pattern relationships. Based on the results, which are depicted in Figure 27, it seems difficult do create a ranking for them.

Measured as a fraction of those respondents who see the respective relationship as very important or fairly important, it can be inferred that a new organisation scheme should emphasise relationships that are linked to the chaining of patterns:

- Pattern X uses pattern Y as part of its solution
- Pattern X frequently causes pattern Y

[1] The number of non-response items for this question is 14.

Nonetheless at least 84 percent of the respondents considered all pattern relationships as at least somewhat important.

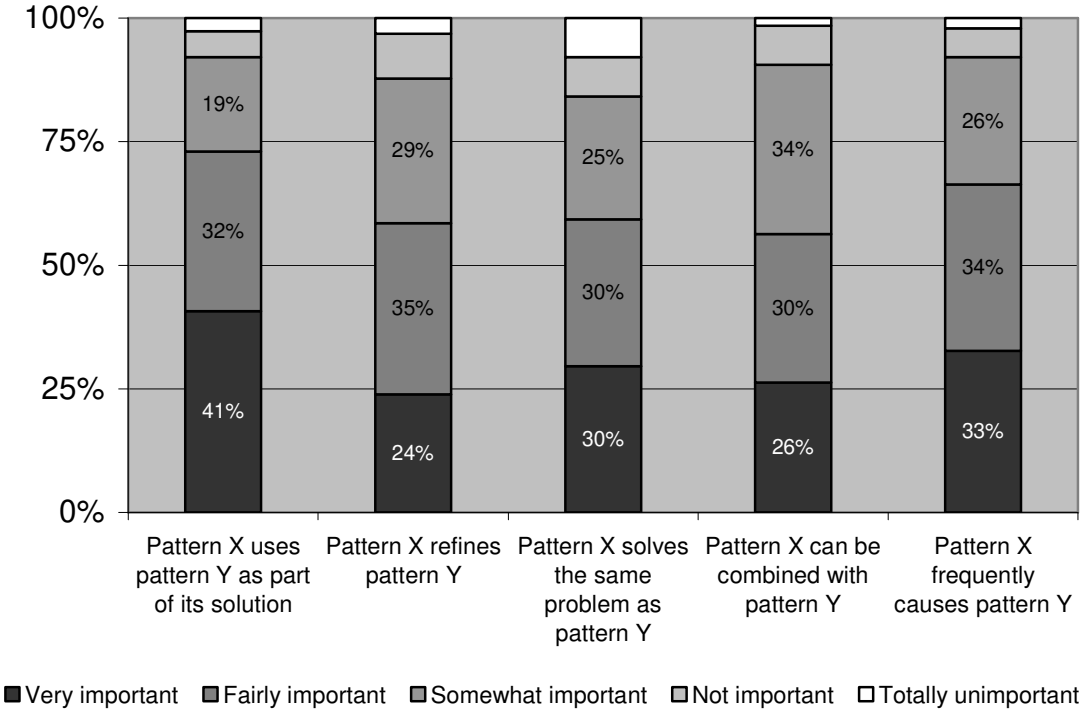

Figure 27: Importance of different pattern relationships for pattern retrieval and pattern application (Question 5E)[1]

3.3.6 Experience with Patterns (AoI6)

Based on available papers and experience reports, the following properties related to the use of patterns were identified.

Software Properties and Pattern Usage

P1. Patterns support the development of software with specific qualities [11].

P2. Patterns help to develop complex and heterogeneous software architecture [11].

[1] The number of non-response items for this question lies between 10 and 12.

P3. Patterns improve the documentation of the software development process [11], [44].

P4. Patterns improve extensibility, portability and reusability of software [12], [24], [43].

P5. The use of patterns results in a clearer design [43].

P6. Patterns guarantee the retention of the original design by restricting changes in maintenance [12].

P7. It is difficult to recognise a pattern in source code although using patterns developed the code [7], [27].

P8. Patterns increase runtime costs by introducing an additional layer of abstraction [43].

Individual Experience with Patterns

P9. Patterns improve communication in the development process between the individuals who are familiar with the respective patterns [4], [24], [44].

P10. Patterns support the integration of new software developers [24], [42].

P11. It is difficult to decide which pattern is the best for a specific problem [46].

P12. Pattern descriptions are insufficient for a direct application [44].

P13. Pattern mining is a human-intensive task [42].

P14. Patterns are difficult to learn [12].

P15. Patterns lead developers to think they know more than they actually do [24], [42].

Patterns in the Software Process

P16. Object-oriented patterns help in the transition to object-oriented technology [42].

P17. Patterns increase the predictability of decisions in analysis, design, and implementation; and increase the standardisation of models and program code.

P18. In contrast to other reuse approaches, patterns have to be implemented again and again [27].

P19. Patterns rely on the experience of previous projects. They do not generate new solutions [35].

Costs and Benefit of Patterns

P20. The use of patterns is not linked to high costs [45].

P21. By using patterns, one can avoid costs related to the development of a new solution [45].

P22. Patterns reduce the risk inherent in software development projects [42].

In the questionnaire question 6A asks for the respondents' perspective on these properties. The results of this are depicted in Figure 28. A more detailed description of the collected data can also be found in Appendix 2.

For an analysis of ordinal data quartiles, for example a median, or the mode can be used. In the following, I will apply the mode although this has some drawbacks. It is influenced by the size of the categories. A different demarcation can result in a different mode. Furthermore, it can not be used to describe the distribution of the data. This does not seem to be a problem since the actual data does not show any bipolar distributions.

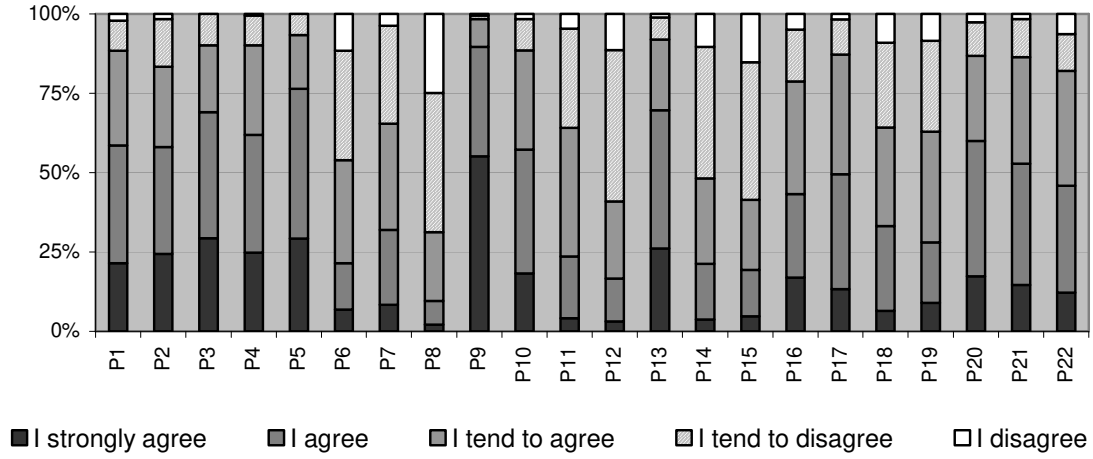

Figure 28: Benefits and drawbacks of the use of patterns (Question 6A)[1]

By far, most respondents (55 percent) agreed with the statement that patterns improve communication among experts and chose the category "I strongly agree". This supports the relevance of pattern description forms. Moreover, a well-chosen name for each pattern can help to convey what its content is. Among the open-ended responses of this question was one remark regarding the negative consequences for the communication process,

"Patterns may even cause more harm than good if some members of the team are unfamiliar with the patterns being used and talked about by others."

For statements P1 to P5, P10, P13, P20, and P21 the majority selected "I agree." This can be interpreted as a broad consensus on these properties of pattern usage. An interpretation of the results for the statements P7, P11, P16, P17, P18, P19 and P22, however, is more difficult since the mode for them was in the category "I tend to agree." The same applies for statements P6, P8, P12, P14 and P15 for which the mode is in the category "I tend to disagree." The most appropriate interpretation could either be one of moderate support or refusal of these statements. The only statement rejected by more than two-thirds of the respondents was that patterns increase runtime costs.

Many respondents took the opportunity to offer their own opinions about the use of patterns. A summary of these can be found in Appendix 3.

[1] The exact values as well as numbers of non-response items can be found in Appendix 2. The figure for non-response items lies between 5 and 17.

About half of the respondents were very satisfied with their use of patterns and so the expectations towards them were exceeded. None of the respondents were dissatisfied with patterns, which not surprisingly reflects the high level of interest in them.

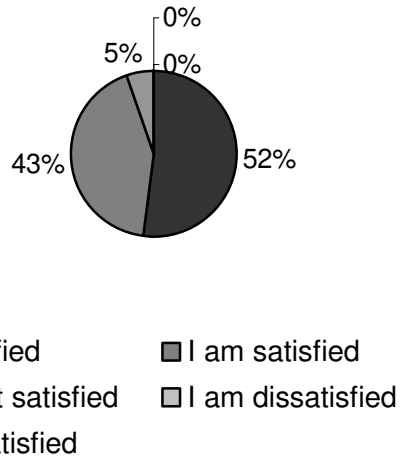

Figure 29: Satisfaction with patterns (Question 6B)

In which ways do pattern users see problems in applying them? Question 6D aims to answer this question. Only the problem to find a matching pattern for a current problem sticks out. Half of the respondents consider this as either very problematic or fairly problematic. This encourages the development of a new pattern organisation scheme that simplifies the search for a matching pattern.

The number of available patterns appears to be relatively unproblematic, since more than 50 percent of the respondents saw it this way. Technologies can be classified as pacemaker technologies, key technologies, or base technologies. A sufficient number of patterns indicates a low potential for further development. This would make patterns a base technology.

An analysis of the remaining four problem categories is difficult. Between 61 percent and 75 percent of the respondents perceived these areas as more or less problematic.

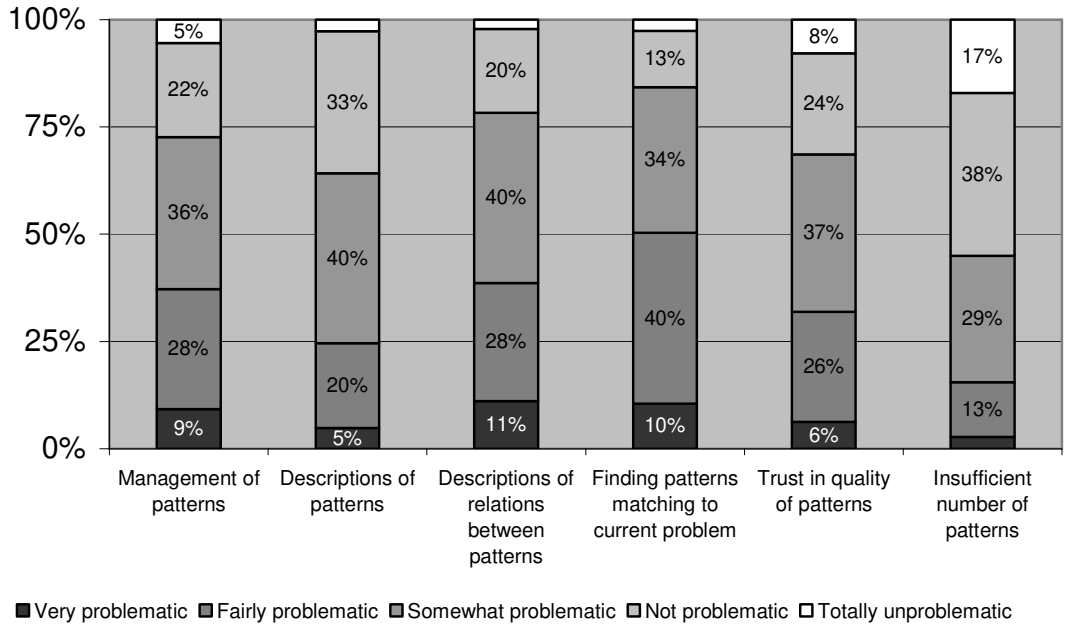

Figure 30: Problematic areas linked to the use of patterns (Question 6D)[1]

In addition, the respondents had a chance to give comments on problems that did not fit in the given problem categories. The following remarks were made:

Organising Patterns

- Rarely are patterns documented at different levels as ALEXANDER did.
- More patterns should be documented in pattern languages.
- There are many non-integrated patterns.
- Explosions of different patterns make their integration useless.

Management of Patterns

- Maintenance of pattern catalogues
- Many duplicated patterns

[1] The number of non-response items for this question (from left to right): 17, 13, 11, 9, 9, 13

- Differentiation of patterns
- Publication of patterns in many sources
- "Communication and retention of patterns"

Pattern Descriptions

- "Pattern descriptions that do not include bad consequences"
- Use of unnecessary domain-specific terminology in pattern descriptions.
- "Some of the published patterns are not really patterns."

Usage of Patterns

- Lack of understanding of pattern languages
- "People use patterns even if they are not needed."

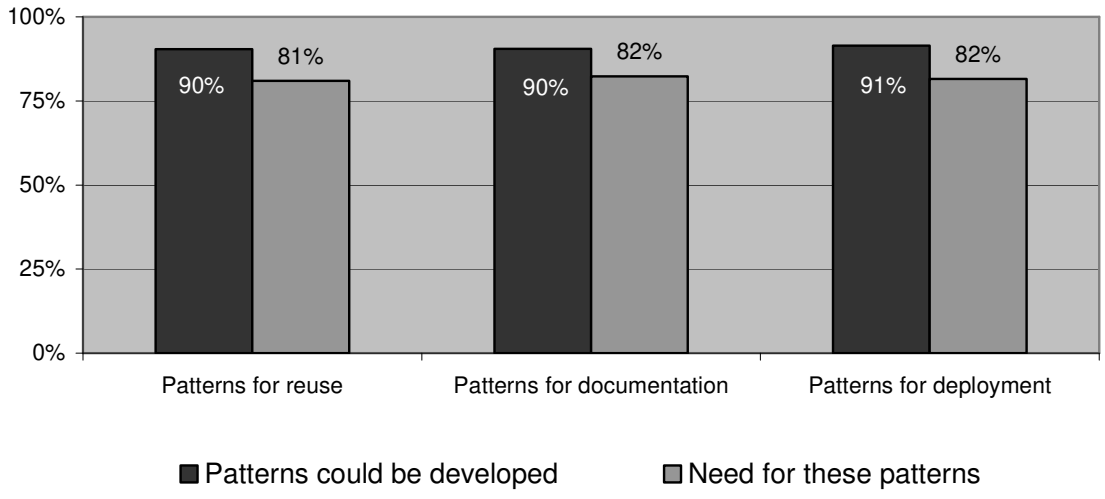

Figure 31:Development of further pattern types (Question 6C)[1]

A very high percentage of the respondents think that patterns for reuse, documentation, and deployment could be developed. About 90 percent of the respondents answered yes to the

[1] The number of non-response items for this question lies between 11 and 21.

respective part of question 6C. A slightly smaller number of respondents (about 80 percent) think that there is a need for such patterns. Based on that, one can conclude that patterns for this software development activities will occur in the future. In fact, one respondent pointed out that there were already papers about the use of patterns in software deployment (e.g. described in [29]).

3.3.7 The Value of Patterns for an Organisation (AoI7)

The majority of the patterns used by the respondents were developed outside of their respective organisations (see Figure 32). While one-third of the respondents exclusively employed patterns from outside their organisation for only eleven percent of the respondents was the main source of the applied patterns from their own organisation. One can conclude that the majority of the respondents did not have any objections against using patterns from the outside.

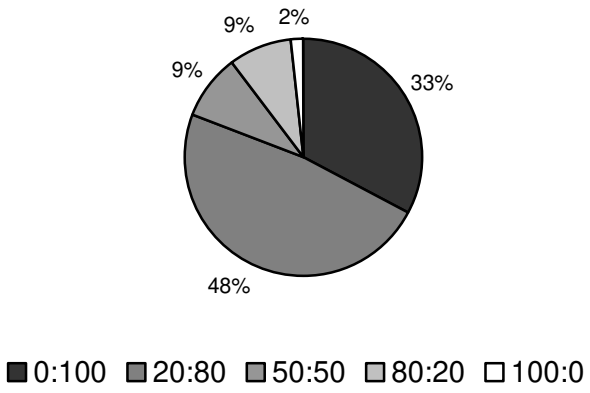

Figure 32: The use of one's own patterns vs. patterns from other sources (Question 7A, own:other)

The responses from question 7A suggest that a market for patterns could evolve. Questions 7B to 7F analyse this further. From Figure 33 we can see that 56 percent of the respondents discussed patterns of their organisations' domain with people from the outside. This contradicts a possible market for patterns in so far as it suggests that the respondents did not consider their patterns as a competitive advantage. In fact, 49 percent of them did not. The high percentage of respondents, who shared their patterns with other organisations supports this finding. On the other hand 69 percent saw the use of patterns as a competitive advantage.

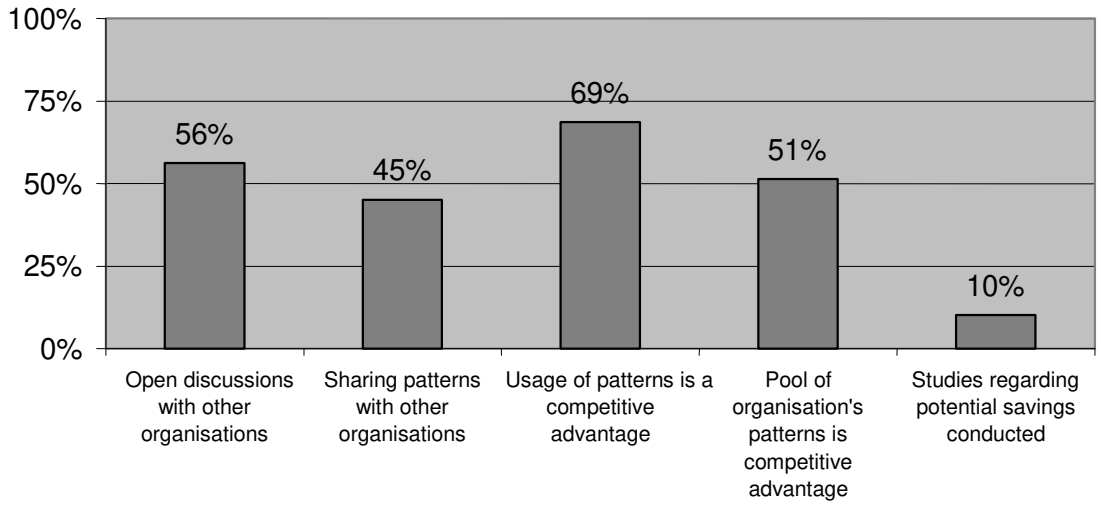

Figure 33: Patterns as a competitive advantage (Questions 7B to 7F)[1]

In only 10 percent of the respondents' organisations were studies conducted about the potential savings of using patterns. One can conclude that the value of patterns for an organisation can not be quantified or at least is difficult to quantify. Any clear reasoning for using patterns, for example based on a cost-benefit-comparison, is hardly feasible.

What could a potential market for patterns look like? Question 7G was designed to answer this. 53 percent of the respondents thought that tools for pattern application would emerge in the pattern-related market. Only 15 percent of them believed that a market for single patterns could evolve while 37 percent considered that a market for groups of related patterns could develop. Eleven respondents saw existing publications about patterns as a confirmation of their marketability. Further open responses in the category "other" were:

- The use of patterns in the documentation of software components
- The possibility to obtain patents on patterns
- The possibility to offer both consulting and educational services related to patterns

[1] The number of non-response item for these questions (from left to right): 8, 9, 12, 15, 15.

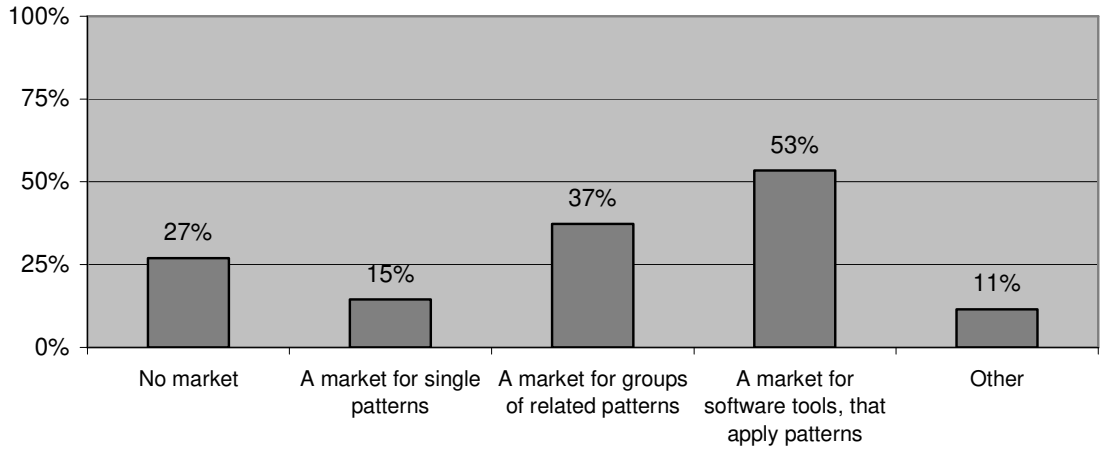

Figure 34: What could a market for patterns look like (Question 7G, multiple choice)?

4. Conclusion

In this paper I briefly discuss the theoretical foundations of pattern usage and then present the results of a survey on it conducted in September and October 2001. In a pragmatic approach, based on an analysis of the understanding of patterns in practice, a characterisation of the term pattern in software development has been given. It includes ten commonly accepted pattern qualities.

A classification in eleven different pattern types, developed in the beginning of this paper, was consistently used in the questionnaire. The categorisation of patterns in activity-based patterns especially contrary to process patterns can cause confusion. An analysis of the application frequency, the familiarity, and the tool support for different pattern types has been made. Among the respondents design patterns were used most frequently and so people were most familiar with them. Design patterns also had the highest degree of tool support of all pattern types. An analysis of the data also revealed, that the use of patterns is dominated by analysis patterns, architectural patterns, design patterns, idioms and patterns for testing. These pattern types are most frequently used and they also show the highest degree of familiarity among the respondents.

Except for design patterns and idioms the use of pattern automation approaches is rare. Nonetheless the presented automation approaches have been put into the context of the developed Microprocess for Patterns. Different forms of pattern characterisations can also be

put into the same context. Pattern description forms and pattern organisation schemes are frequently used in practice. Slightly more than one-third of the respondents were not satisfied with the pattern organisation scheme used. This leaves room for further research. A mapping of pattern hierarchies, similar to ALEXANDER, seems to be a commonly favoured approach.

The idea of a formal integration of patterns into process models was rejected by some respondents. Often patterns were seen as an informal approach that can not be easily formalised.

A list of advantages and disadvantages of patterns, based on separate experience reports, was provided in the questionnaire. The survey put these individual reports on a broader base. The main advantage of patterns is the improvement of communication in the development process. The respective question also offered a chance to give an open answer. A list of these answers can be found in Appendix 3.

Only a brief analysis of patterns as a competitive advantage as well as the likelihood of a pattern-related market has been made. Both tools that apply patterns and further publications on groups of related patterns are expected to emerge. The problem of a possible granting of patents for patterns affects potential marketing strategies. Further research is necessary.

Table of Figures

Figure 1: A classification of patterns 4

Figure 2: The Microprocess for Patterns 6

Figure 3: Pattern automation in the Microprocess for Patterns 9

Figure 4: Initial contact with respondents (Question 8D) 16

Figure 5: Type of organisation of respondents (Question 8C) 18

Figure 6: Do you consider algorithms as patterns (Question 1B)? 19

Figure 7: Which qualities characterise the term "pattern" (Question 1A)? 21

Figure 8: Percentage of pattern consumers and producers (Questions 2A, 2B) 22

Figure 9: How do you introduce patterns in your organisation (Question 2C, multiple choice)? 23

Figure 10: How do you discover new ideas for patterns (Question 2D, multiple choice)? 24

Figure 11: Organisational support for the use and production of patterns (Questions 2E, 2F) 25

Figure 12: Frequency of individual pattern use (Question 3A, all respondents) 26

Figure 13: Frequency of individual use of patterns (Questions 3A, 8A, pre-selection of sample) 27

Figure 14: Usage frequency of patterns in an organisation (Question 3A, all respondents) 28

Figure 15: Familiarity with patterns (Question 3B, all respondents) 29

Figure 16: Familiarity with patterns (Questions 3A, 8A, preselection of sample) 29

Figure 17: Number of pattern tool users (Question 3C) 30

Figure 18: Number of respondent using the different types of pattern automation (Question 3C, see also Figure 17, multiple choice) 31

Figure 19: How do you apply patterns (Question 3D)? 32

Figure 20: Do you use a process model (Question 4A, 2A)? 33

Figure 21: Usage rate of different concepts (Questions 4B, 4A, 2A multiple choice) 34

Figure 22: Is pattern usage and pattern production part of the organisation's process model (Questions 4C, 4D, 4A, 2A)? 34

Figure 23: Usage of different pattern description forms (Question 5A, multiple choice) 35

Figure 24: Usage of different pattern organisation schemes (Question 5B, multiple choice) 36

Figure 25: Is the pattern organisation scheme used a suitable means for pattern retrieval (Question 5C)? *36*

Figure 26: Which is the preferred approach for pattern organisation schemes (Question 5D)? *37*

Figure 27: Importance of different pattern relationships for pattern retrieval and pattern application (Question 5E)..... *38*

Figure 28: Benefits and drawbacks of the use of patterns (Question 6A)..... *41*

Figure 29: Satisfaction with patterns (Question 6B)..... *42*

Figure 30: Problematic areas linked to the use of patterns (Question 6D) *43*

Figure 31:Development of further pattern types (Question 6C) *44*

Figure 32: The use of one's own patterns vs. patterns from other sources (Question 7A, own:other). *45*

Figure 33: Patterns as a competitive advantage (Questions 7B to 7F) *46*

Figure 34: What could a market for patterns look like (Question 7G, multiple choice)?..... *47*

References

[1] Alexander, C., *The Timeless Way of Building*. Oxford University Press, New York, 1979

[2] Ambler, S., *An Introduction to Process Patterns*. Internet: `http://www.ambysoft.com/processPatterns.pdf`, Download: 05.07.2001, 1998

[3] Ballantyne, C., *Why Survey Online? A Practical Look at Issues in the Use of the Internet for surveys in Higher Education*. Internet: `http://cleo.murdoch.edu.au/evaluation/pubs/confs/aea-2000.html`, Download: 05.07.2001, 2000

[4] Beck, K., Coplien, J., Crocker, R., Dominick, L., Meszaros, G., Paulisch, F., Vlissides, J., *Industrial Experience with Design Pattern*. Internet: `http://www1.bell-labs.com/user/cope/Patterns/ICSE96/icse.html`, Download: 05.07.2001, 2001

[5] Beck, K., *Simple Smalltalk Testing: With Patterns*. Internet: `http://www.xProgramming.com/testfram.htm`, Download: 29.09.2001, 2001

[6] Binder, R., *Testing Object-Oriented Systems: Models, Patterns, and Tools*. Addison Wesley Longman, Reading, 2000

[7] Bosch, J: *Design Pattern as Language Constructs*. In Journal of Object-Oriented Programming, May 1998, pp. 18-32

[8] Bosch, J., *Design Patterns & Frameworks: On the issue of Language Support*. Internet: `http://www.ipd.hk-r.se/bosch/lsdforg/bosch.ps`, Download: 05.07.2001, 2001

[9] Brown, K., *Design Reverse-Engineering and Automated Design Pattern Detection in Smalltalk*. Internet: `http://members.aol.com/kgb1001001/Articles/THESIS/thesis.htm`, Download: 05.07.2001, 2001

[10] Bünnig, S: *Entwicklung einer Sprache zur Unterstützung von Design Patterns und Implementierung eines dazugehörigen Compilers, Master Thesis*. Internet: `http://wwwswt.informatik.uni-rostock.de/deutsch/Infothek/uml/pal/DiplomBuennig.pdf`, Download: 05.07.2001, 1999

[11] Buschmann, F., Meunier, R.,Rohnert, H., Sommerlad, P., Stal, M., *Pattern-Oriented Software Architecture*. Wiley, 1996

[12] Cline, M., *The Pros and Cons of Adopting and Applying Design Patterns in the Real World*. In Communications of the ACM, October 1996 Volume 39, No. 10, pp. 47-49

[13] Coad, P., North, D., Mayfield, M., *Object Models: Strategies, patterns, applications*. Prentice-Hall, Englewood Cliffs, 1995

[14] Coplien, J., *A brief history of design patterns*. Internet: http://www.bell-labs.com/user/cope/Patterns/ICSE96/node3.html, Download: 11.09.2001, 1996

[15] Coplien, J., *The Human Side of Patterns*. Internet: http://www1.bell-labs.com/user/cope/Patterns/C++Report/OrgPatterns-1.html, Download: 21.03.2001, 1996

[16] Coad, P., Yourdon, E., *Object-oriented Analysis*. Prentice-Hall, Englewood Cliffs, 1991

[17] Czaja, R., Blair, J., *Designing Surveys A Guide To Decisions and Procedures*. Pine Forge Press, Thousand Oaks, 1995

[18] Dillman, D., Bowker, D., *The Web Questionnaire Challenge to Survey Methodologists*. Internet: http://survey.sesrc.wsu.edu/dillman/zuma_paper_dillman_bowker.pdf, Download: 05.07.2001, 2001

[19] Dillman, D., Tortora, R., Bowker, D., *Principles for Constructing Web Surveys*. Internet: http://survey.sesrc.wsu.edu/dillman/papers/websurveyppr.pdf, Download: 05.07.2001, 1998

[20] Dillman, D., *Mail and Other Self-Adminstered Surveys in the 21st Century: The Beginning of a New Era*. Internet: http://survey.sesrc.wsu.edu/dillman/papers/svys21st.pdf, Download: 05.07.2001, 1998

[21] Forrest, E*., Internet Marketing Research: Resources and Techniques*. Internet: http://gaius.cbpp.uaa.alaska.edu/afef/, Download: 05.07.2001, 1998

[22] Fowler, M., *Analysis Patterns: Reusable Object Models*. Addison Wesley Longman, Menlo Park, 1997

[23] Gamma, E., Helm, R., Johnson, R., Vlissides, J., *Design Patterns: elements of reuseable object-oriented software*. Addison-Wesley Publishing Company, Wokingham, 1994

[24] Helm, R., *Patterns in Practice*. In ACM Proceedings of the Tenth Annual Conference on Object-oriented Programming Systems, Languages, and Applications 1995, pp. 337-341

[25] Keller, R,; Schauer, R., Robitaille, S., Pagé, P., Pattern-Based Reverse Engineering of Design Components. In ACM Proceedings of the International Conference on Software Engineering 1999, pp. 226-235

[26] Kerievsky, J., *Startle & Shout: Describe the Problem*. Internet: http://www.industriallogic.com/pulse/20000130.html, Download: 21.03.2001, 2000

[27] Mannhaupt, D., *Integration of Design Patterns into Object-Oriented Design using Rational Rose, Master Thesis*. Internet: http://wwwswt.informatik.uni-rostock.de/deutsch/Infothek/uml/rational/patterns/DiplomMannhaupt.pdf, Download: 05.07.2001, 2000

[28] Martin, R., Riehle, D., Buschmann, F., Eds., *Pattern Languages of Program Design 3*. Addison Wesley Longman, Wokingham, 1998

[29] Marquardt, K., *Physical Patterns*. Internet http://www.coldewey.com/europlop98/Program/Papers/Marquardt.ps, Download: 05.07.2001, 1998

[30] Medlin, C., Roy, Subroto, Chai, T., *World Wide Web versus Mail Surveys: A Comparison and Report*. Internet: http://www.anzmac99.unsw.edu.au/anzmacfiles/M/Medlin.pdf, Download: 05.07.2001, 1999

[31] Meijers, M., *Tool Support for Object-Oriented Design Patterns, Master Thesis*. Internet: http://www.serc.nl/people/florijn/papers/mm-thesis.ps.gz, Download: 05.07.2001, 1996

[32] Mowbray, T., Malveau, R., *Corba Design Patterns*. John Wiley & Sons, New York, 1997

[33] Nonaka, I., *The Knowledge-Creating Company*. In Harvard Business Review, November, December 1991, pp. 96-104

[34] Ohtsuki, M., Yoshida, N., Makinouchi, A., *A Source Code Generation Support System Using Design Pattern Documents Based on SGML*. In IEEE Proceedings of Sixth Asia Pacific Software Engineering Conference (APSEC 99), 1999, pp. 292-299

[35] Pescio, C., *Principles Versus Patterns*. In IEEE Computer Volume 30 Issue 9, 1997, pp. 130-131

[36] Prechelt, L., Unger, B., Philippsen, M., *Documenting Design Patterns in Code Eases Program Maintenance*. Internet: http://www.ubka.uni-karlsruhe.de/cgi-bin/psgunzip/1997/informatik/34/34.ps, Download: 05.07.2001, 1997

[37] Prechelt, L., Unger, B., Schmidt, D., *Replication of the first controlled experiment on the usefulness of design patterns: Detailed description and evaluation*. Internet: http://wwwipd.ira.uka.de/~prechelt/Biblio/wustl_pattern34-1997.ps.gz, Download: 05.07.2001, 1997

[38] Pressman, R., *Software Engineering: A Practitioner's Approach European Adaptation*. 5th Edition, McGraw-Hill Publishing Company, London, 2000

[39] Rising, L., *The Pattern Almanac 2000*. Addison Wesley Longman, Wokingham, 2000

[40] Rising, L., *Patterns Mining*. Internet: http://www.agcs.com/supportv2/techpapers/patterns/papers/mining.htm, Download: 03.04.2001, 2001

[41] Royce, W., *Managing the development of large software systems: Concepts and techniques*. In: Proceedings of IEEE WESTCON, Los Angeles, 1970, pp. 1-9.

[42] Schmidt, D., *Experience Using Design Patterns to Develop Reusable Object-Oriented Communication Software*. Internet: http://www.cs.wustl.edu/~schmidt/CACM-95.ps.gz, Download: 05.07.2001, 2001

[43] Schmidt, D., Cleeland, C., *Applying a Pattern Language to Develop Extensible ORB Middleware*. Internet: http://www.cs.wustl.edu/~schmidt/PDF/ORB-patterns.pdf, Download: 05.07.2001, 1999

[44] Schmidt, D., Stephenson, P., *Experience Using Design Patterns to Evolve Communication Software Across Diverse OS Platforms*. Internet: http://www.cs.wustl.edu/~schmidt/PDF/ECOOP-95.pdf, Download: 05.07.2001, 1995

[45] Seen, M., Taylor, P., Dick, M., *Applying a Crystal Ball to Design Pattern Adoption*. In IEEE Proceedings of International Conference on Technology of Object-Oriented Languages (TOOLS 33) , 2000, S. 443-454

[46] Shaw, M., *Patterns for Software Architectures*. Internet: http://www.cs.cmu.edu/afs/cs/project/vit/ftp/pdf/PLoP94.pdf, Download: 05.07.2001, 1994

[47] Software: *Together Control Center 5.0*. Internet: http:://www.together.com/, Download: 01.05.2001, 2001

[48] Unger, B, Tichy, W., *Do Design Patterns Improve Communication? An Experiment with Pair Design*. Internet: http://hometown.aol.com/geshome/wess200 0/unger-tichy.pdf, Download: 22.03.2001, 2000

[49] Vlissides, J., *Reverse Architecture*. Internet: ftp://st.cs.uiuc.edu/pub/patterns/papers/revarch.ps, Download: 02.07.2001, 2001

[50] n/a: *CVOPS – A Tool for Portable Communication Software*. Internet: http://www.vtt.fi/tte/tte22/cvops/, Download: 29.09.2001, 2001

[51] Yacoub, S., Xue, H., Ammar, H., *Automating the Development of Pattern-Oriented Designs for Application Specific Software Systems*. In IEEE Proceedings of 3rd Symposium on Application-Specific Systems and Software Engineering Technology 2000, S. 163-170

Appendix 1: List of Pattern Qualities

Q1. A pattern occurs again and again

Q2. A pattern captures the knowledge of experts

Q3. A pattern is a solid practical procedure and no innovative solution

Q4. A pattern was successfully used more than once (Rule of Three)

Q5. A pattern is perfect

Q6. Patterns are independent of object-oriented concepts

Q7. A pattern is a (shining) example

Q8. A pattern is a combination of some components

Q9. A pattern is a sample/template/model

Q10. A pattern describes a problem-solution pair

Q11. A pattern describes its context and the forces which limit its solution

Q12. A pattern description is isolated and short

Q13. A pattern is modified when applied, but keeps its general form

Q14. A particular pattern can be applied to many actual problems

Q15. A pattern unifies communication of experts

Appendix 2: Table of Data for Question 6A

No.	Description	I dis-agree	I tend to disagree	I tend to agree	I agree	I strongly agree	Number of non-responses
P1	Patterns support the development of software with specific qualities.	2%	9%	30%	**37%**	21%	9
P2	Patterns help to develop complex and heterogeneous software architecture.	2%	15%	25%	**34%**	24%	7
P3	Patterns improve the documentation of the software development process.	0%	10%	21%	**40%**	29%	6
P4	Patterns improve extensibility, portability and reusability of software.	1%	9%	28%	**37%**	25%	6
P5	The use of patterns results in a clearer design.	0%	7%	17%	**47%**	29%	5
P6	Patterns guarantee the retention of the original design by restricting changes in maintenance.	12%	**35%**	32%	15%	7%	9
P7	It is difficult to recognise a pattern in source code although using patterns developed the code.	4%	31%	**34%**	24%	8%	9
P8	Patterns increase runtime costs by introducing an additional layer of abstraction.	25%	**44%**	22%	7%	2%	11
P9	Patterns improve communication in the development process between the individuals who are familiar with the respective patterns.	1%	1%	9%	35%	**55%**	6
P10	Patterns support the integration of new software developers.	2%	10%	31%	**39%**	18%	8
P11	It is difficult to decide which pattern is the best for a specific problem.	5%	31%	**41%**	19%	4%	5

No.	Description	I dis-agree	I tend to disagree	I tend to agree	I agree	I strongly agree	Number of non-responses
P12	Pattern descriptions are insufficient for a direct application.	11%	**48%**	24%	13%	3%	7
P13	Pattern mining is a human-intensive task.	1%	7%	22%	**44%**	26%	12
P14	Patterns are difficult to learn.	10%	**41%**	27%	18%	4%	7
P15	Patterns lead developers to think they know more than they actually do.	15%	**43%**	22%	15%	5%	9
P16	Object-oriented patterns help in the transition to object-oriented technology.	5%	16%	**36%**	26%	17%	17
P17	Patterns increase the predictability of decisions in analysis, design and implementation; and increase the standardisation of models and program code.	2%	11%	**38%**	36%	13%	12
P18	In contrast to other reuse approaches, patterns have to be implemented again and again.	9%	27%	**31%**	27%	6%	13
P19	Patterns rely on the experience of previous projects. They do not generate new solutions.	8%	29%	**35%**	19%	9%	11
P20	The use of patterns is not linked to high costs.	3%	11%	27%	**43%**	17%	10
P21	By using patterns, one can avoid costs related to the development of a new solution.	2%	12%	34%	**38%**	15%	9
P22	Patterns reduce the risk inherent in software development projects.	6%	12%	**36%**	34%	12%	10

Appendix 3: Open Answers to Question 6A

„You can regard patterns as a compression method. For example, describing a design with patterns takes less space than doing it without. Patterns compress w/o trading precision for space.“

„Using design patterns without language specific good implementation examples may cause more harm than good. For instance, many of the C++ examples in the GoF book are simply bad. Since patterns are often used by newbies it is of utmost importance that the examples lead the reader in the right and not the wrong direction when it comes to the actual implementation.“

„They are too informal. They should be validated (argument to show that solution solves the problem).“

„I've used patterns in the design and implementation of large software systems. They are a good mechanism for communicating abstractions and discussing alternative approaches. I'm not that dogmatic about their use, however.“

„Patterns are only part of the total solution. They are an excellent communications media between individuals on a project.“

“Patterns can make developers happy. If they are understood, new concepts (patterns) are implemented successfully. Sometimes too many patterns result in over-engineered solutions. It is easier to introduce flexibility than to get rid of unneeded flexibility.”

“Some background: during my time as a research assistant at the university, I dealt a lot with all kinds of pattern related question. In my current affiliation, patterns are not really supported. I am using them more on a personal basis. There's a good chance I'll use patterns more frequently in near future. Personally, I really like the idea behind patterns. I believe, however, that the term pattern is somewhat excessively stressed by some individuals. I believe patterns are valuable for many steps in software development, but I do not believe they can solve all issues raised by the inherent complexity of today’s software systems. For me, it is a good concept 'above single objects'. Not more - and not less!”

“Risks: are hardly related to software design issues, so patterns do not play a major role. Maintenance: patterns help understanding an existing system when there is a short hint given which role a specific class plays to other classes I would not overestimate the use of pattern tools. Most patterns are basically ideas, and contain little code to implement. A very global base class could make your design worse rather than save some lines of code. Templates can sometimes help here.”

“Object-oriented patterns greatly aid understanding how to use object-oriented concepts properly.”

“I think that hype (especially the hype surrounding MVC) and lazy dissemination of pattern understanding (bad seminars about Singleton) are damaging to the cause of good development and of pattern usage.”

“Unfortunately most developer don't know patterns. If they are using a oo-language like Java most of them use idioms like Event Generator, but they don't know that this is just a application of GoF's Observer/Subscriber pattern. If you need to understand the design behind some Java APIs you need to understand patterns. But most developers hack. Patterns avoid long, ugly methods. I use refactoring to reduce long methods and I use patterns to avoid this refactoring. We need more people that teach and proclaim the benefits of using patterns.”

“For some reason, it took a lot of time for me to understand the pattern language by various schools of thought. For this reason, I have had bitter experiences trying to understand the various terminologies that have proliferated even among the leading pattern advocates like GOF and Siemens. I would wish it were possible that there was less confusion regarding the terminology and a common standard for expressing a pattern is developed. So that once and for all patterns are standardized. My experience with patterns has indicated that it takes a lot of time and its easy for you to misinterpret them. But once understood, well, you will see the dividends. I like patterns if it was possible for someone to guide me and the books are not the best of places. I must admit that I still am struggling to create a pattern language of my own.”

“Once one is familiar with patterns, they probably flow naturally, and the ‘pattern mining’ process probably becomes more efficient. When first introduced to the concept, though,

current pattern terminology tends to be confusing and abstract; only with experience does the use of patterns begin to come naturally."

"Patterns are more useful when one has already some experience in the context. They cannot help understand the context."

"Patterns reduce complexity and thus help to comprehend an unknown design for new developers, though they have to be familiar with the most known patterns. The ability to handle abstraction is a prerequisite."

"Ad Patterns in the SW Process #3: There are generic implementations of a couple of design patterns."

"My experience with patterns is this: I'm a student about to begin my masters thesis in computer science and have chosen patterns and their implications in the development process as my headline (i.e. do they in reality improve communication, and between whom). Unfortunately most firms I have contacted (now counting five major companies) state that it is a very relevant question and that they consider patterns a very applicable tool (in whatever field they have chosen to use it in), but unfortunately there is no time in their schedules to get involved in such a project. It is my feeling that they (till now) see this project as a pressure to really use patterns 'after the book' and this will take too much time. I hope to get some data soon."

"Patterns are good - without them I would not be a respected coder/designer!"

"For many of these questions my answer would be 'It depends' or 'It can be'. There are trade-offs to be made when using patterns. Patterns can be very useful but they don't necessarily make a hard problem easy - they just help in solving the problem."

"The primary value of patterns, in my experience, is that they facilitate communication of design concepts, between software designers, and as software designers try to understand software written by others. When the members of a team are all familiar with a set of patterns, they provide a concise vocabulary for discussing design concepts. This value is greatly diminished, and patterns may even cause more harm than good if some members of the team are unfamiliar with the patterns being used by and talked about by others."

"Patterns help elucidate the differences between surface structure and deep structure of types of problems. Experienced developers and designers recognize the deep structure for the sake of less experienced developers and designers (like AI systems do), so less experienced designers do better - if they understand the pattern. There's no harm in that!"

"It was difficult for me to decide whether to agree or disagree with the statement above, 'Patterns rely on the experience of previous projects. They do not generate new solutions."

"I have observed that object-oriented design involving competent use of patterns can lead to quicker applications with smaller memory foot prints than object-oriented applications designed without patterns. In some cases this has been due to patterns such as fly-weight that directly addresses the memory footprint issue (and has improved speed as a side effect), in other cases the use of patterns gave a deeper level of understanding of what was being designed, which lead to a design that was fast because it was simple, and that could be optimised because it was simple."

"As an academic, I have mostly been involved with creation rather than application."

"Usability patterns are the ones I am most familiar with. These have not been highlighted in this survey. Responses are partly from the usability patterns perspective, but not entirely."

"Hard to answer the questions here as our case is unusual - we are a research group … (i.e. we do not use patterns in our (University) organisation, but our main partner organisation does. I have answered from the perspective of the partner organisation in so far as I am aware of how they use them.)"

"The pattern form can capture design experience and design decisions. Documenting a design with the pattern form, even if the design decisions are not proved patterns, is a good idea."

The following three responses are taken from question 8E because they also describe experiences with patterns:

"One big problem with patterns is that individual ones aren't all that useful by themselves. It's having a structured and hierarchical pattern language that's useful, letting people go from abstract to concrete, at different scales. Lots of books out there with lots of isolated patterns

are not very useful. Makes it hard to find things. GoF book has lots of links between their patterns, but they're all at the same level of architecture (i.e. a few objects at most). We need macro-design, as well as micro-design patterns."

"This survey make impression of a fairly formalistic approach to patterns. Patterns are more literature than technique, they have a different value system than formal approaches."

"I have spent several years participating in the patterns community, writing patterns, attending workshops, and on the whole, I have been disappointed. What the industry needs is fewer, better patterns - some general-purpose, some domain-specific."