University of Technology, Dresden

Department of Systems Engineering

Pattern-based Software Development
An Empirical Study

- Questionnaire -

Thoralf Czichy (thoralf@czichy.org)

Helsinki, 18. November 2001

Abstract

This is the paper version of the questionnaire used in the survey.

Table of Contents

1. Cover Letter/Email 1

2. How to Use This Online Questionnaire 3

3. Understanding of Patterns (AoI1) 4

4. Patterns in Your Organisation (AoI2) 6

5. Different Pattern Kinds - Usage and Tool Support (AoI3) 8

6. Pattern Usage and Pattern Mining in the Process Model (AoI4) 12

7. Characterising Patterns – Descriptions and Organisation Schemes (AoI5) 14

8. Experience with Patterns (AoI6) 17

9. The Value of Patterns for an Organisation (AoI7) 21

10. General Information, Comments, Receiving a Summary 23

11. Final Page 26

12. Descriptions of the Pattern Types used in the Survey 27

1. Cover Letter/Email

Dear Colleague,

This email is about a survey on the use of Patterns in Software Development conducted by the University of Technology, Dresden, Germany. It is part of a research project for the Systems Engineering department. You are kindly requested to participate in this survey by following this link:

`http://wiseweb.wiwi.tu-dresden.de/survey/`

You have been chosen to take part, since you showed an interest in this subject in the past (e.g. by publishing papers on Software Patterns). This survey is largely dependent on your assistance. If you know other people who might be interested in this survey please feel free to forward this email. Everyone's help is deeply appreciated.

The survey aims to clarify practical issues in seven major areas of interest:

1. Reaching a common understanding of the term Pattern as it is used in software engineering.
2. Do organisations produce/write and consume/introduce Patterns? How?
3. Usage of different pattern kinds (e.g. Design Patterns, Organisational Patterns, Architectural Patterns) in general and usage of tools for pattern automation.
4. Possible inclusion of pattern usage and pattern mining in the process model.
5. Usage of different pattern organisation schemes (e.g. a pattern catalogue).
6. Summarise gained experience with pattern usage.
7. Recognition of Patterns as a potential competitive advantage.

The questionnaire will take about 20 minutes to complete. All information will be handled anonymously and will only be published in aggregated form. If you provide your email address, a summary of the results will be sent to you as soon as it is available.

To start the survey please follow this link for a brief introduction:

http://wiseweb.wiwi.tu-dresden.de/survey/

Many thanks for your time and effort,

Sincerely,

Thoralf Czichy, Student

Andreas Dietzsch, Department of Systems Engineering, University of Technology, Dresden

Prof. Dr. Werner Esswein, Department of Systems Engineering, University of Technology, Dresden

2. How to Use This Online Questionnaire

Try to Answer all Questions

Please try to answer all questions. If any questions do not apply to you, or you do not feel you are able to answer them, just leave them blank or unchecked.

Navigation

To navigate from one section to another, use the 'Previous' and 'Next' Buttons located at the top and bottom of each page. No additional action is required to save the data you entered, this happens automatically when you change sections.

Use the 'Cancel' button to suspend the survey. This button is located at the right top and bottom of each section. This saves all data you have entered so far and suspends the survey.

When you come back at a later time using your previously created login, all data will be reloaded, so you can continue where you left off.

Javascript

This survey can be used without JavaScript enabled. However, some actions might not work as expected:

You will not be able to close windows by clicking on a hyperlink.

In some sections, JavaScript helps you by answering some questions based on your previous answers.

It is best to enable JavaScript while using the survey.

You can now click the button to proceed to the first section.

3. Understanding of Patterns (AoI1)

(1a)

In your opinion, which of the following qualities characterise a pattern in the software engineering domain (e.g. Design Patterns, Analysis Patterns, Process Patterns)?

Click (Yes) to agree, (No) to disagree.

(Yes) (No)

Characteristics Based on Pattern Origins

- A pattern occurs again and again
- A pattern captures the knowledge of experts
- A pattern is a solid practical procedure and no innovative solution
- A pattern was successfully used more than once (Rule of Three)
- A pattern is perfect
- Patterns are independent of object-oriented concepts

Characteristics Based on Pattern Descriptions

- A pattern is a (shining) example
- A pattern is a combination of some components
- A pattern is a sample/template/model
- A pattern describes a problem-solution pair
- A pattern describes its context and the forces which limit its solution
- A pattern description is isolated and short

Characteristics Based on Pattern Application

- A pattern is modified when applied, but keeps its general form
- A particular pattern can be applied to many actual problems
- A pattern unifies communication of experts

(1b)

Do you consider algorithms (e.g. Quicksort, Dijkstra's shortest path algorithm) as patterns?

(Yes) (No)

4. Patterns in Your Organisation (AoI2)

(2a)

Do you use patterns in your organisation?

(Yes) (No)

- Personally
- Other people in my organisation

(2b)

Do you write new patterns in your organisation (e.g. for later internal or external use)?

(Yes) (No)

(2c)

How do you learn/introduce new patterns in your organisation?

(Yes) (No)

- No introduction
- Self study (e.g. books, internet)
- Pattern study group (a group that meets on a regular basis and discusses patterns)
- External or internal courses
- Other - If you selected Other, please describe:

(2d)

How do you discover new ideas for patterns in your organisation?

Click (Yes) to agree, (No) to disagree.

- No discovering
- By interviewing experts
- Internal pattern workshops, meetings (e.g. a monthly meeting)
- External pattern workshops, meetings (e.g. a conference workshop)
- System reviews by experts (e.g. code reviews)
- As part of my daily work (e.g. while creating a design model)
- Other - If you selected Other, please describe:

(2e)

Is the use of patterns in your organisation promoted?

(Yes) (No)

If yes, by which means:

(2f)

Is the discovery of new patterns in your organisation promoted?

(Yes) (No)

If yes, by which means:

5. Different Pattern Kinds - Usage and Tool Support (AoI3)

(3a)

Which of the following pattern types do you use strictly personally? Which ones do you use in your organisation (e.g. for communication amongst software engineers or in documentation)? Please indicate the frequency of use of patterns of these types.

For a more detailed description of each pattern kind you can click here.

Please select a rating from 1 to 5 with the following meanings: 1 = Never | 2 = Rarely (less than once a month) | 3 = Sometimes (1-3 times a month) | 4 = Frequently (once or twice a week) | 5 = Always (more than twice a week)

Personally

- Patterns for Requirement Analysis
- Analysis Patterns
- Architectural Patterns
- Design Patterns
- Idioms (Implementation Patterns)
- Patterns for Testing
- Patterns for Maintenance
- Process Patterns
- Organisational Patterns
- Educational Patterns
- Patterns on Pattern Writing

In your Organisation (e.g. for communication, in documentation):

- Patterns for Requirement Analysis
- Analysis Patterns
- Architectural Patterns
- Design Patterns
- Idioms (Implementation Patterns)
- Patterns for Testing
- Patterns for Maintenance
- Process Patterns
- Organisational Patterns
- Educational Patterns
- Patterns on Pattern Writing

(3b)

Please give an estimate of the number of patterns that you are familiar with in each category. 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. For a more detailed description of each pattern kind you can click here.

(Number of Familiar Patterns = none, 1-5, 6-10, 11-15, 16-20, 21-30, more than 30)

- Patterns for Requirement Analysis
- Analysis Patterns
- Architectural Patterns

- Design Patterns
- Idioms (Implementation Patterns)
- Patterns for Testing
- Patterns for Maintenance
- Process Patterns
- Organisational Patterns
- Educational Patterns
- Patterns on Pattern Writing

(3c)

Please indicate whether you use tools in relation with each pattern type. If you use a tool please select its type(s) from the five options. For a more detailed description of each pattern kind you can click here.

A. Computerised pattern descriptions (e.g. using a document management system as Lotus Notes or a database-like repository using HTML, SGML or XML)

B. Automated presentation of patterns in models/source code (e.g. comment tags or an ellipse with role associations, UML 1.3)

C. Automated discovery of patterns (e.g. automated pattern mining in existing code)

D. Automated introduction/implementation of patterns into models/source code (e.g. code refactoring or a 'pattern wizard' in a design tool)

E. Other (please describe)

(Yes) (No) (A) (B) (C) (D) (E) If you selected (E), please describe:

- Patterns for Requirement Analysis

- Analysis Patterns
- Architectural Patterns
- Design Patterns
- Idioms (Implementation Patterns)
- Patterns for Testing
- Patterns for Maintenance
- Process Patterns
- Organisational Patterns
- Educational Patterns
- Patterns on Pattern Writing

(3d)

How do you apply patterns?

(Yes) (No)

- I am using patterns as a part of my previous experience and education.
- I am using patterns directly from books, papers, magazines etc.
- I am using a software tool, which supports pattern usage.

6. Pattern Usage and Pattern Mining in the Process Model (AoI4)

(4a)

Do you use a process/method/model in your company? A process/method/model is a formal and disciplined approach to assigning task and responsibilities within a software development organisation (e.g. Rational Unified Process, Waterfall Model).

(Yes) (No)

If you choose no, you may proceed directly to the first question of the next section (section 5). Otherwise, please anwer the following questions too.

(4b)

Do you think the process/method/model, as you are using it in your company, supports the following concepts:

(Yes) (No)

- Incremental/Iterative Development
- Evolutionary Prototype (A prototype which evolves into a final product)
- Explorative Prototype (A Throwaway Prototype)
- Reuse (e.g. reuse of components, code, design models, …)
- Object-oriented Concepts (e.g. object-oriented programming languages, object-oriented design)

(4c)

Is the utilisation of patterns an explicit part of your development process/method/model (something that is formally described as part of your development process/method/model)?

(Yes) (No)

(4d)

Is the discovery of new patterns and pattern ideas (pattern mining) an explicit part of your development process/method/model?

(Yes) (No)

7. Characterising Patterns – Descriptions and Organisation Schemes (AoI5)

(5a)

Which of the following pattern forms do you use? If you use a pattern form, click (Yes), otherwise click (No)

(Yes) (No)

- None
- GoF (GAMMA et al) pattern form (Pattern Name and Classification, Intent, Also Known As, Motivation, Applicability, Structure, Participants, Collaborations, Consequences, Implementation, Sample Code, Known Uses, Related Patterns)
- Canonical form (Name, Alias, Problem, Context, Forces, Solution, Example, Resulting Context, Rationale, Known Uses, Related Patterns)
- Informal Portland form (Opening statement, therefore)
- Alexander's informal pattern form (IF Context, Examples, Problem and Forces THEN Reasons, Design Form, Rule, Solution, Context and Other Patterns)
- Other - If you selected Other, please describe:

(5b)

Which of the following pattern organisation schemes (e.g. Pattern Systems) do you use?

(Yes) (No)

- None (please proceed to question 5d)
- According to the GoF (GAMMA et al) scheme for Design Patterns
    - Purpose - Creational, Structural, Behavioural

    - Scope - Class, Object

- According to the BUSCHMANN (BUSCHMANN et al) scheme

    - Pattern Type - Architectural Pattern, Design Pattern, Idiom

    - Problem Area - e.g. Concurrency Issues

- Other - If you selected Other, please describe:

(5c)

Do you consider the used pattern organisation scheme as a suitable means for pattern retrieval?

(Yes) (No)

(5d)

Considering pattern retrieval, which of the following terms/approaches do you think is most appropriate as a pattern organisation scheme?

Choose one.

- None

- Pattern Catalogue

- Pattern System

- Pattern Language

- Other - If you selected Other, please describe:

(5e)

Considering the development of a new pattern organisation scheme for pattern retrieval and pattern application, how important do you rate the following pattern relationships for your work with patterns?

Please select a rating from 1 to 5 with the following meanings: 1 = Totally unimportant - 2 = Not important - 3 = Somewhat important - 4 = Fairly important - 5 = very important

- Pattern X uses Pattern Y as part of its solution
- Pattern X refines Pattern Y
- Pattern X solves the same problem as pattern Y
- Pattern X can be combined with pattern Y
- Pattern X frequently causes pattern Y (e.g. Model-View-Controller causes Observer Pattern)

8. Experience with Patterns (AoI6)

(6a)

Based solely on your experience only, which of the following statements would you support? Please select a rating from 1 to 5 with the following meanings: 1 = I disagree - 2 = I tend to disagree - 3 = I tend to agree - 4 = I agree - 5 = I strongly agree

Software Properties and Pattern Usage

- Patterns support the development of software with specific qualities.
- Patterns help to develop complex and heterogeneous software architectures.
- Patterns improve the documentation of the software development process.
- Patterns improve extensibility, portability and reusability of software.
- Use of patterns results in a clearer design.
- Patterns guarantee the retention of the original design by restricting changes in maintenance.
- It is difficult to recognise a pattern in source code although using patterns developed the code.
- Patterns increase runtime costs, by introducing an additional layer of abstraction.

Individual Experience with Patterns

- Patterns improve communication in the development process between the individuals who are familiar with the respective patterns.
- Patterns support the integration of new software developers.
- It is difficult to decide which pattern is the best for a specific problem.
- Pattern descriptions are insufficient for a direct application.

- Pattern mining is a human-intensive task.
- Patterns are difficult to learn.
- Patterns lead developers to think they know more, than they actually do.

Patterns in the Software Process

- Object-oriented patterns help in the transition to object-oriented technology.
- Patterns increase the predictability of decisions in analysis, design and implementation and increase standardisation of models and program code.
- In contrast to other reuse approaches patterns have to be implemented again and again.
- Patterns rely on the experience of previous projects. They do not generate new solutions.

Costs and Benefit of Patterns

- Use of patterns is not linked to high costs.
- By using patterns, one can avoid costs related to the development of a new solution.
- Patterns reduce the risk inherent in software development projects.

Other

If you want to tell us something else about your experience with patterns, you can do it here:

(6b)

Did patterns meet your expectations?

Choose one

- I am very satisfied (patterns exceeded my expectations)

- I am satisfied (all of my expectations have been met)
- I am somewhat satisfied (most of my expectations have been met)
- I am dissatisfied (some of my expectations have not been met)
- I am very dissatisfied (none of my expectations have been met)

(6c)

Do you think patterns exist or could be developed for the following areas? Do you think there is a need for these patterns?

(Yes) (No)

Patterns Could be Developed?

- Patterns for Reuse
- Patterns for Documentation
- Patterns for Installation/Deployment

Need for Patterns?

- Patterns for Reuse
- Patterns for Documentation
- Patterns for Installation/Deployment

(6d)

Which areas do you consider problematic with patterns? Please select a rating from 1 to 5 with the following meanings: 1 = Totally unproblematic - 2 = Not problematic - 3 = Somewhat problematic - 4 = Fairly problematic - 5 = Very problematic

- Management of patterns within the pattern community (e.g. the "approval" of new patterns)

- Descriptions of patterns
- Descriptions of relations between patterns
- Finding patterns matching to your current problem
- Trust in quality of patterns
- Insufficient number of patterns

Other problematic areas:

9. The Value of Patterns for an Organisation (AoI7)

(7a)

Could you give an estimate for your usage of patterns developed within your organisation against patterns from somewhere else (e.g. from books, conference papers)?

Choose one.

- Own 0% : other 100%
- Own 20% : other 80%
- Own 50% : other 50%
- Own 80% : other 20%
- Own 100% : other 0%

(7b)

Do you openly discuss patterns of your organisation's domain with people external to your organisation?

(Yes) (No)

(7c)

Do you share patterns with other organisations?

(Yes) (No)

(7d)

A competitive advantage makes it possible to produce at lower costs or to differentiate yourself from your competitors by providing unique or superior value to the buyer. Do you consider the use of patterns as a competitive advantage?

(Yes) (No)

(7e)

Do you consider the pool of patterns specific to your organisation as a competitive advantage?

(Yes) (No)

(7f)

Did you conduct studies regarding potential savings by using patterns in your organisation?

(Yes) (No)

(7g)

Do you expect a pattern-related market to emerge?

(Yes) (No)

- No, not at all
- Yes, a market for single patterns
- Yes, a market for groups of related patterns
- Yes, a market for software tools, that apply patterns
- Yes, other - If you selected Other, please describe:

10. General Information, Comments, Receiving a Summary

(8a)

Which of the following activities are part of your work?

(Yes) (No)

- Requirements Analysis
- Analysis
- Architectural Design
- Detailed Design
- Implementation/Coding
- Testing
- Deployment/Introduction
- Operations and Maintenance
- Documentation
- Reuse Analysis
- Other - If you selected Other, please describe:

(8b)

Do you use general tools for one of the following activities?

Examples:

Rational Suite AnalystStudio for Requirements Analysis

Together ControlCenter for Detailed Design

Microsoft Visual Studio for Implementation

(Yes) (No)

- Requirements Analysis
- Analysis
- Architectural Design
- Detailed Design
- Implementation
- Testing
- Deployment/Introduction
- Operations and Maintenance
- Documentation
- Reuse Analysis
- Other - If you selected Other, please describe:

(8c)

I'm working in a ... Choose one

- Company.
- University.
- Self-employed.
- Other - If you selected Other, please describe:

(8d)

How did you learn about this questionnaire?

Choose one

- I am subscribed to a mailing list where a link to this questionnaire was published.
- I have got an email directly from the survey's conductors.
- A friend/colleague sent me a link to this questionnaire.
- Other - If you selected Other, please describe:

(8e)

Please include any additional comments. If you have any further inquiries about this survey you can also send an email to pattern-survey@wiseweb.wiwi.tu-dresden.de.

(8f)

If you are interested in receiving a summary of results when available, please enter your email address in the following textfield:

If you have completed all sections, you are now done. Click to go to the final page.

11. Final Page

Many thanks for your time and effort.

If you have entered your email address in the previous section, a summary of this survey will be sent to you when it becomes available.

If you know other people who might be interested, click here to send them a link to this page (http://wiseweb.wiwi.tu-dresden.de/survey). Everyone's help is deeply appreciated.

If you have any further questions, feel free to send an email to pattern-survey@wiseweb.wiwi.tu-dresden.de.

Sincerely,

Thoralf Czichy, Student, University of Technology, Dresden

Andreas Dietzsch, Department of Systems Engineering, University of Technology, Dresden

Prof. Dr. Werner Esswein, Department of Systems Engineering, University of Technology, Dresden

12. Descriptions of the Pattern Types used in the Survey

Patterns for Requirement Analysis

Description: A Pattern for Requirement Analysis describes techniques and processes for capturing and validating the behavioural and non-behavioural requirements of a software system.

Example: Customer Rapport. How do you build and establish a good relationship with a customer.

References:

Creel, C. "Requirements by Patterns", Internet: `http://www.sdmagazine.com/articles/1999/9912/9912d/9912d.htm`, 01-03-21.

Whitenack, B. "RAPPeL: A Requirements-Analysis-Process Pattern Language for Object-Oriented Development", in Coplien, J., and Schmidt, D., eds. "Pattern Languages of Program Design", pp. 259-291, Wokingham: Addison-Wesley Publishing, 1995.

Analysis Patterns

Description: Analysis Patterns are concepts that represent a common construction in analysis or business modelling.

Example: Organisation Hierarchies. This pattern describes the modelling of (multiple) organisational hierarchies in an object-oriented class diagram.

Reference:

Fowler, M. "Analysis Patterns: Reuseable Object Models", Menlo Park: Addison Wesley Longman, 1997.

Architectural Patterns

Description: Architectural Patterns help you to specify the fundamental structure of a software system.

Example: Layers. The Layers pattern helps to structure applications that can be decomposed into groups of subtasks in which each group of subtasks is at a particular level of abstraction.

References:

Buschmann, F., et al. "Pattern-Oriented Software Architecture, Volume 1: A System of Patterns", New York: John Wiley & Sons, 1996.

Martin, R., et al., eds. "Pattern Languages of Program Design 3", Wokingham: Addison Wesley Longman, 1998.

Design Patterns

Description: A Design Pattern is an effective solution to a recurring problem in the design stage. Object-oriented Design Patterns consist of a specification of class and object interactions and their underlying intent.

Example: Observer. Defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.

References:

Gamma, E., et al. "Design Patterns: elements of reuseable object-oriented software", Wokingham: Addison-Wesley Publishing Company, 1994.

Mowbray, T., Malveau, R. "Corba Design Patterns", New York: John Wiley & Sons, 1997.

Idioms (Implementation Patterns)

Description: An Idiom is a pattern defined for a specific programming language on a low level of abstraction. It describes how to implement particular aspects of components and their relationships in a programming language.

Example: Virtual Constructor. How to build an object of known abstract type, but of unknown concrete type, without violating the encapsulation of the inheritance hierarchy.

References:

Buschmann, F., et al. "Pattern-Oriented Software Architecture, Volume 1: A System of Patterns", New York: John Wiley & Sons, 1996.

Coplien, J. "C++ Idioms", Internet: http://www.bell-labs.com/user/cope/Patterns/C++Idioms/EuroPLoP98.html, 01-07-02.

Patterns for Testing

Description: Patterns for Testing are related to Design Patterns. A Pattern for Testing accompanying a Design Pattern defines a configuration of objects needed to test the interactions between classes that are integrated according to the design described by the Design Pattern.

Example: ObserverTestPattern. This pattern allows for testing the correct implementation of the Design Pattern Observer. It pays special attention to the completeness of notifications, the correctness of queries sent by the Observers and the consistency of the data in the Subject and the Observers.

Reference:

McGregor, J. "Test Patterns: Please Stand By", in: Journal of Object-Oriented Programming, pp. 14-19, June 1999.

Patterns for Maintenance

Description: Patterns for Maintenance describe techniques for typical maintenance activities as optimisation, stabilisation or dealing with depreciation.

Example: Consolidate the Program to Support Evolution and Reuse. Insight gained during the prototype and consolidation phases can be employed to refactor the software system.

References:

Dewar, R., et al. "Identifying and Communication Expertise in Systems Engineering: A Patterns Approach", in: IEEE Software, pp. 145 - 152, Volume 146 Issue 3, 1999.

Foote, B., and Opdyke, W. "Lifecycle and Refactoring Patterns That Support Evolution and Reuse", in Coplien, J., and Schmidt, D., eds. "Pattern Languages of Program Design", pp. 239-257, Wokingham: Addison-Wesley Publishing, 1995.

Process Patterns

Description: A process pattern is a collection of general techniques, actions, and/or tasks (activities) for developing software systems within a software development organisation.

Example: The Selfish Class. Describes how to overcome psychological problems related to the reuse of existing artefacts (e.g. source code).

References:

Ambler, S. "An Introduction to Process Patterns", Internet: `http://www.ambysoft.com/processPatterns.pdf`, 01-07-05

Foote, B., and Yoder, J. "The Selfish Class", in Martin, R., et al., eds. "Pattern Languages of Program Design 3", pp. 451-470, Wokingham: Addison Wesley Longman, 1998.

Organisational Patterns

Description: Organisational Patterns describe the structure and practices of human software development organisations.

Example: Design by Team. A general pattern that describes how people can work together to design and develop software.

References:

Harrison, N. "Organizational Patterns for Teams", in Coplien, J., and Schmidt, D., eds. "Pattern Languages of Program Design 2", pp. 345-352, Wokingham: Addison-Wesley Publishing, 1996.

"Organizational Patterns", Internet: http://www.bell-labs.com/cgi-user/OrgPatterns/OrgPatterns?OrganizationalPatterns, 01-03-21

Educational Patterns

Description: Educational Patterns describe proven techniques used for teaching technical knowledge.

Example: Iterative Course Development. Describes the use of iterative improvement to develop a course that takes the needs of all kinds of students into account.

Reference:

Anthony, D. "Patterns for Classroom Education", in: Coplien, J., and Schmidt, D., eds. "Pattern Languages of Program Design 2", pp. 391-406, Wokingham: Addison-Wesley Publishing, 1996.

Patterns on Pattern Writing

Description: Patterns on Pattern Writing describe and demonstrate writing practices which have been observed to be particularly effective.

Example: Code Samples. Describes the use of code samples to improve the quality of a pattern description.

Reference:

Meszaros, G., and Doble, J. "A Pattern Language for Pattern Writing", Internet: http://www.hillside.net/patterns/Writing/pattern_index.html, 01-07-05