

Where are my patterns when I need them?

ANTONIO MAÑA, ITIS Software, University of Málaga

This paper summarizes the outcome of the Imagination Run Wild session “Where are the patterns when I need them?” at PLoP’24. The paper describes the session proposal and key aspects, its evolution and the conclusions and agreements reached. Patterns are essential tools in different disciplines, and especially in software engineering, offering proven solutions to recurring problems and promoting reusable, maintainable, and scalable designs. However, identifying and applying the right patterns in a specific project can be challenging, particularly when faced with complex requirements or unfamiliar contexts. In this session we argued that a significant challenge in this process is the absence of a widely accepted repository of patterns. While numerous resources and tools exist, they often vary in terminology, organization, and scope. This fragmentation makes it harder for developers to locate, understand, and adopt the most appropriate patterns, ultimately hindering the potential benefits of pattern-driven development. Moreover, they normally focus on the pattern consumption side (e.g. search and classification) but lack support for the pattern development and trust sides.

Categories and Subject Descriptors: **[Software and its engineering]**: Software creation and management—*Design patterns*.

General Terms: Security Architecture, Pattern-driven Development.

Additional Key Words and Phrases: Pattern-driven Development, Pattern tools

ACM Reference Format:

Maña, A. 2024. Where are my patterns when I need them? . HILLSIDE Proc. of Conf. on Pattern Lang. of Prog. 31 (October 2024), 9 pages.

1. INTRODUCTION

PLoP’24 featured workshops and presentations under the umbrella of “let your imagination run wild” sessions, which as the title suggests, are limited only by the proposers’ imagination. One of these sessions dealt with the issue of the lack of a widely accepted place to look for patterns. This paper describes the session proposal and key aspects, its evolution and the conclusions and agreements reached. The main call for participation for attendees was:

Have you ever spent time on a problem and wondered whether there is a pattern that could save your day? Have you decided that it was not worth going through different searches just to find such a pattern, in case it existed, because that’s what your past experience indicates? Do your students believe that all “patterns” are in the GoF book? Have you found badly written, hard-to-adopt patterns? Have you wished the pattern was not just a pdf document and that you could use it in your software engineering tools? And after these questions, are you wondering why these problems have not yet been solved?

Join us in a collaborative effort for taking patterns to a new practical level, not by adding more patterns, but by defining the tools that we need to support the whole pattern lifecycle and to facilitate the seamless use of patterns in your daily work, no matter if you are a software engineer or are applying patterns to other domains.

The goal of the session’s was to answer three main questions:

1. **Why** in 2024 we have not managed to have a common, usable, comprehensive, collaborative and open tool (repository/website/application/...) to collect, curate and share patterns?
2. **What** has prevented this? What are the **problems** that have made **previous initiatives** fail?
3. **How** can we **make it succeed**? What **features** should it have?

This work is supported by the Junta de Andalucía, Spain, under contract QUAL21 010UMA.

Author’s address: Antonio Maña. ITIS Software, Universidad de Málaga. Calle Arquitecto Francisco Peñalosa, 18, 29010 Málaga, Spain; email: amana@uma.es.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission. A preliminary version of this paper was presented in an Imagination Run Wild session at the 31st Conference on Pattern Languages of Programs (PLoP). PLoP’24, OCTOBER 1316, Columbia River Gorge, Washington, USA. Copyright 2025 is held by the author(s). HILLSIDE 978-1-941652-03-9

Then, the final goal was to start a community of committed colleagues to work on a plan and to carry it out. The session was focused on learning from past initiatives, to find out whether a tool beyond a simple repository of pdf documents is possible and, if deemed possible, collaboratively define the basic elements for it to be implemented.

If we look at the many years of working on patterns, and the fact that there is no widely used tool to really support the collection, evolution and sharing of patterns (and other related tools like principles), maybe it is quite wild to imagine such a tool becoming reality in the end.

Before the conference, and to help starting the discussion, we contacted some relevant members of the community to gather information on past initiatives, with their focus, characteristics, basic functionalities, lifetime, etc. This information was provided to the attendees during the session.

The session was conducted following the 3 main questions. The ideas are described in the following sections. In each of them, we also provide the results of the preliminary interaction with relevant researchers and practitioners in the field.

Preliminary Note: To make the discussion less verbose and more precise, in the remainder of the paper we will use the following terms:

- **Artifact**: will be used to refer to all knowledge representation artifacts, like patterns, principles, etc.
- **Forge**: will be used to refer to all activities in a pattern lifecycle, including identification, collection, curation, evolution and sharing of artefacts. Hence, the complete process would be called **forging**.
- **PatternForge**: will be used to refer to our target tool.

1.1 Question 1. Do we need it? What for?

The first question wanted to validate the perceived usefulness of the tool for the audience. The expected outcome was to identify reasons for doing/not doing it and a model of a generalized pattern lifecycle graph and to reach a decision on the usefulness of a tool to support it.

The main reasons for not developing such a tool were that people are already used to search for patterns in papers, and the lack of confidence on the tool/repository gaining the necessary popularity.

The main ideas for doing it were that it could represent a step forward in the adoption of patterns as a useful problem-solving tool, that it could potentially promote the creation of new patterns and that it would also advance knowledge on the ways people develop and use patterns. The issue of trust in the content of a general community-driven repository was also mentioned. Wikipedia was suggested as an example, since the wiki was the first attempt to create a tool to develop and present patterns.

Finally, the session proposer described additional ideas coming from the preliminary work done that were well accepted by the attendees. Among these, we highlight the following reasons and goals for developing PatternForge:

Centralized Knowledge Repository. Just like Wikipedia or other authoritative centralized sources of information the tool would provide access to a vast amount of knowledge currently spread across books, articles, and various online resources. A dedicated web application can serve as a centralized repository, making it easier for people to access and share valuable information. This would have advantages like:

- **Efficiency**: It would save time to be able to find well-organized, curated content in one place.
- **Consistency**: Ensures that artifacts are presented in a standardized manner, reducing ambiguity and errors in interpretation.
- **Up-to-Date Information**: Patterns in papers are frozen. Patterns in PatternForge would be continuously updated to adapt to the evolution of their contexts.
- **Integration with other tools**. For instance, in the area of software engineering, integration with various Integrated Development Environments (IDEs) Continuous integration/continuous deployment (CI/CD) tools and modelling environments would foster the adoption of the artifacts in

practice, enhancing productivity and ease of use. Moreover, in other environments the API would enable the creation of custom applications, extensions, and tools tailored to specific needs.

Standardization and Best Practices. Adhering to well-defined pattern crafting processes that guarantee the peer-review and collaboration in improving the quality of the artefacts and guarantees the quality of the artifacts we publish.

Learning and Professional Development. Continuous learning is essential for us to keep up with the evolving life and technology landscapes. A tool like PatternForge would be beneficial as an educational resource for students, junior, and even experienced professionals looking to deepen their knowledge. Moreover, the tool would also foster community engagement by promoting discussions, feedback, and collaboration among developers. The tool would also support the exchange of ideas, could be used to host collaborative projects in specific areas with peer reviews, leading to the development of higher-quality artifacts, and could act as a hub for open-source initiatives, where consumers can contribute to and benefit from a collective pool of knowledge.

A basic pattern (artifact) lifecycle was also discussed. Figure 1 depicts the states identified.

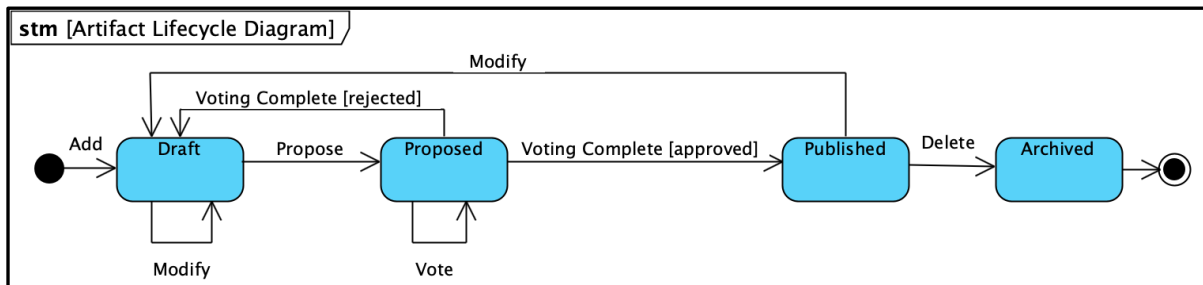


Figure 1. Artifact Lifecycle

The depicted artifact lifecycle assumes a collaborative approach supported by the tool. Proposers create *Drafts* of the artifacts, they evolve with the contributions from the community and when they are considered good enough they are *Proposed*. A voting process is used to decide whether the artifact is ready to be published or needs additional improvements. Once it is approved, the artifact is Published. All patterns are Archived after deletion. Both the voting and deletion processes can be tailored to fit different contexts for different types of artifacts and different domains.

1.2 Question 2. Learn from the past.

Second topic was to analyze previous initiatives, the way they were designed, implemented and used. In this case, the audience was expected to prepare lists of to-do and not-to-do for a tool to support the defined pattern lifecycle.

Among the reasons proposed by the audience, we can highlight the following ones:

- Wrong or limited niche “market” targeting. Most initiatives were developed to serve a specific community.
- Lack of support for evolution. Some of the fields that have adopted patterns and other artifacts like software engineering are rapidly evolving and require an effort to keep the pace in maintaining, evolving and adding new the knowledge representation artifacts.
- Lack of community orientation and contributor engagement. A majority of the existing initiatives have been a one-person heroic effort. Making a successful PatterForge requires active participation and contribution from the community.
- Quality Control and Standardization. As an extension of the previous point, ensuring the quality and consistency of the content is critical but challenging for initiatives that do not have a large community behind.

- Adapting existing successful platforms. In some fields, there are already several platforms and communities, such as Stack Overflow or GitHub, that can partially fulfill the role of PatternForge.

1.3 Question 3. Shape the future. Be part of it.

The last part of the session was short due to the limited duration of the session. However, we were able to agree on the following:

- The reasons to build PatternForge are already clear, and the advantages seem worth the effort.
- A small group will further explore different alternatives. The Pattern Repository developed by Chris Barney (<https://patternlanguageforgamedesign.com/PatternLibraryApp/PatternLibrary/>) was considered an excellent starting point and will be explored as a candidate for developing new types of patterns such as Security Patterns, in the line defined by the late Eduardo B. Fernandez.
- The tool should support different roles, different communities and implement a well-defined and community-oriented workflow and lifecycle.
- The tool should have version control features to manage changes, branching and evolution of artifacts over time.
- The tool should provide advanced search capabilities.
- The tool should provide APIs to allow other tools to interact with it.
- The tool should provide features for users to comment on and discuss patterns, providing feedback and sharing experiences.
- The tool should provide mechanisms for peer review, allowing reviewers to provide structured feedback.

By incorporating these features and exploring several existing tools as candidates, a tool designed to support the artifact lifecycle can significantly enhance their practical use and management. This is especially useful for software engineering patterns but most of the features are also useful for other types of artifacts and other domains.

2. SESSION REPORT

The fact that despite decades of experience with design patterns, many software engineers and practitioners still struggle to find, adopt, and share them effectively is at the heart of the proposal for the session. In fact, this session was born from a common frustration: the feeling that there should be a better way to access and work with patterns, but somehow, no widely accepted solution has emerged.

The discussion began with a fundamental question: why is it so difficult to locate the right pattern at the right time?

Participants reflected on their experiences, noting that while design patterns are essential tools, they often remain buried in books, scattered across diverse repositories, or locked within academic papers. Some admitted to abandoning the search entirely, either because they didn't expect to find a relevant pattern or because prior experiences suggested it wouldn't be worth the effort. Others pointed out that the perception of patterns is often limited to those found in the famous Gang of Four book, ignoring the vast number of useful artifacts developed over the years. One participant mentioned that he believed that patterns would remain a teaching tool and would never become core elements in any development process. Finally, several participants felt that the effort required to understand and ensure the correct application of a new pattern was an important barrier. The lack of a common format, and the trust in the pattern were also mentioned as barriers.

One of the central goals of this session was to explore why, in 2024, there still isn't a common, usable, and widely adopted platform for collecting, curating, and sharing design patterns. Consequently, participants were asked for feedback on reasons that they believe have prevented previous initiatives from succeeding and to envision what an effective solution might look like. We also discussed the possibility of creating a new tool –tentatively named PatternForge– that could finally provide a

structured, collaborative, and open environment for managing artifacts like patterns, anti-patterns, principles, and best practices.

The conversation quickly focused to the reasons why previous attempts at pattern repositories had failed. One major challenge was the sheer diversity of problems that patterns try to solve. With such a wide range of potential use cases, different communities often struggled to agree on a common format or organizational structure. Some participants argued that pattern repositories suffered from “pattern zealots” who insisted on rigid definitions, making it difficult for more flexible, practical approaches to gain traction. Others pointed out that many previous efforts had no sustainable business model, making long-term maintenance impossible. A few also mentioned that security concerns had plagued earlier repositories, either due to misuse or because of credibility and trust issues.

Beyond these challenges, there was also a sense that many developers simply don’t think about patterns in their daily work. Patterns remain largely absent from best practice workflows in many fields, particularly in security and software architecture. Even when repositories exist, they often lack the depth, freshness and adoption levels necessary to become truly valuable. However, despite these obstacles, participants remained optimistic about the potential for a better approach.

The session then moved toward imagining what a successful PatternForge could look like. It was agreed that such a platform should go beyond static documentation and become an interactive, metadata-rich tool that integrates with modern development environments. Instead of being just another website filled with PDFs, it should support dynamic curation, community-driven validation, and even real-time integration into software engineering workflows. Some envisioned a system that could evolve patterns based on community feedback, while others emphasized the need for sustainable governance to ensure long-term success.

By the end of the discussion, several key takeaways emerged. There was broad agreement that a generalized pattern lifecycle model was needed—one that clearly defined how patterns are created, refined, and consumed. Participants also compiled a list of common pitfalls from past initiatives, hoping to avoid repeating the same mistakes. Perhaps most importantly, a concrete next step was identified: forming a group of committed individuals to refine the vision, build a prototype, and present it to the broader community.

While the road ahead remains uncertain, this session laid the foundation for what could become a transformative effort. The desire for a better way to manage and use patterns is evident, and with the right approach, PatternForge could be the tool that finally makes it possible. Anyone interested in contributing is welcome to contact the session proposer in order to become involved.

APPENDIX

This appendix summarizes the results of the preliminary study done before the session. It follows the structure of the questions asked to the participants.

Q1¹

How do you use patterns and other similar tools (principles, etc.)? NOTE: we will refer to them as "artifacts" in this questionnaire

11 responses

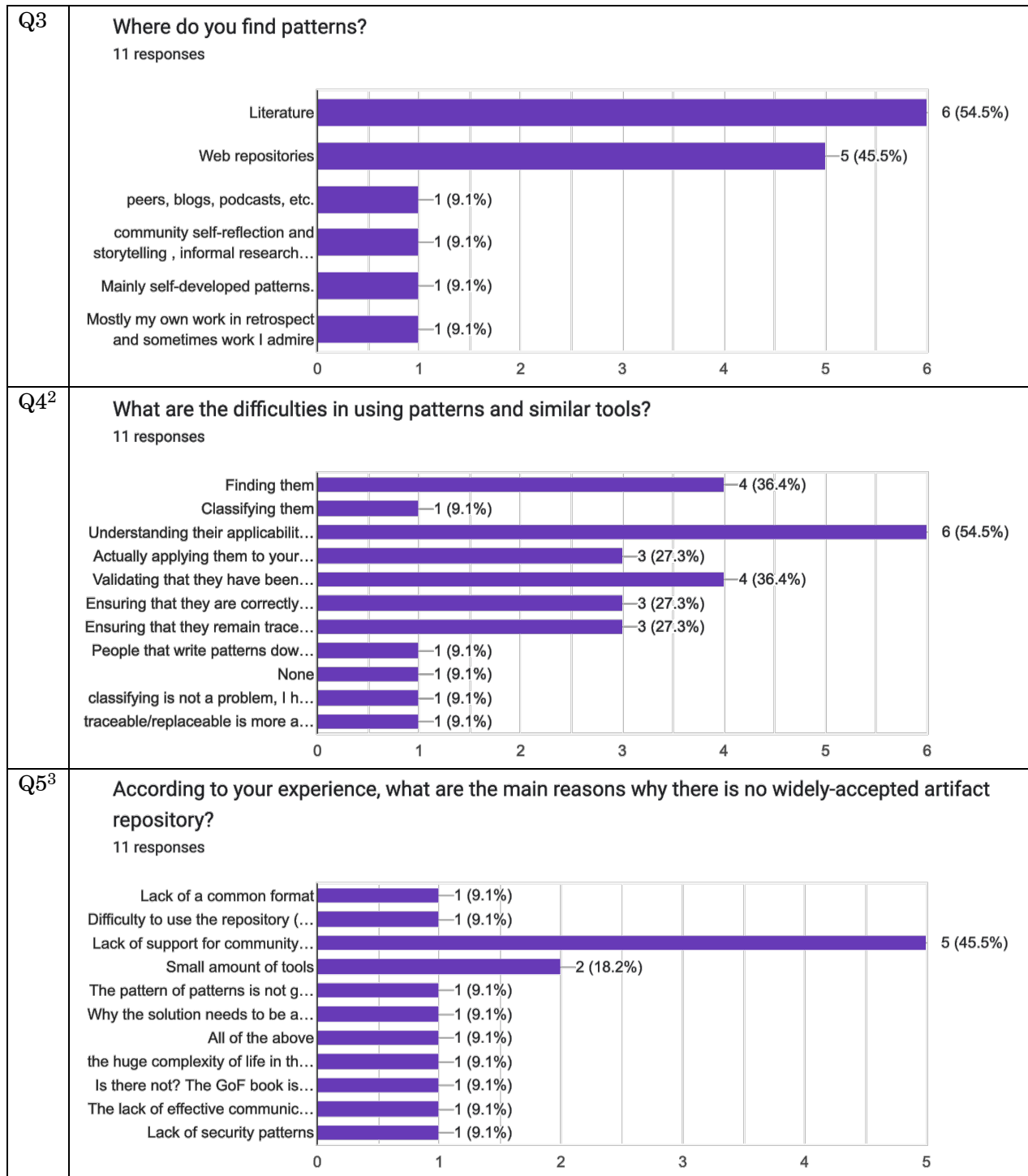
Usage Category	Count	Percentage
I don't normally use them	1	9.1%
I use the ones I know by heart	4	36.4%
I search for patterns and other t...	5	45.5%
I just create patterns for others...	3	27.3%
I generally view patterns broadl...	1	9.1%
I aim to seed a practice of patte...	1	9.1%
I distinguish things I learn as p...	1	9.1%

Q2

How do you envisage the main role of artifacts in the future?

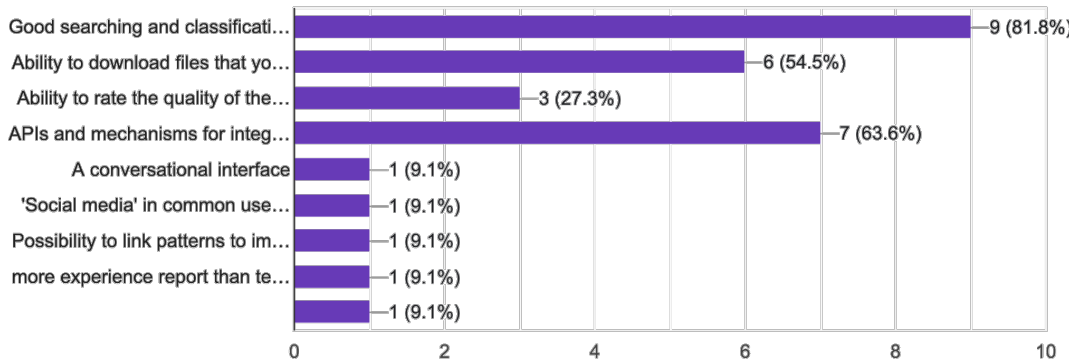
- I see them as a key solution, due to they are the way to provide consolidated solutions.
- Annotated with meta data so that they can be used more effectively, e.g. dependent type systems.
- continuation of design and communication tool
- (assuming you mean patterns) The same role it has today: document practical knowledge.
- Time-proven tools
- In a close partnership with community self-reflection and heart searching, as global crisis deepens and stable order comes to pieces
- They will evolve to the point of becoming a 100% must to solve problems for everybody without reinventing the wheel. Optimizing solutions for most problems, and having to think about solution to specific problems that don't really fit into the categories because the system is custom made and fitted to the client's requirements.
- In the view of security, patterns should be available as design and system architecture patterns, in order to advance the way systems (in the widest possible sense) are designed today. The current best practice results in largely insecure and brittle architectures that do not make adequate use of the existing technologies.
- principles => patterns => frameworks => libraries => code
- They will help to provide good solutions together with quality code.

¹ Possible responses: I don't normally use them / I use the ones I know by heart / I search for patterns and other tools when I need to solve a problem / I just create patterns for others to use / Other



² Possible responses: Finding them / Classifying them / Understanding their applicability to your specific problem / Actually applying them to your system / Validating that they have been correctly applied (i.e. it is really the pattern and not a wrong interpretation of it) / Ensuring that they are correctly implemented (i.e. they are translated to the implementation) / Ensuring that they remain traceable and can be later replaced or removed / Other

³ Possible responses: Lack of a common format / Difficulty to use the repository (e.g. lack of search, indexing, etc.) / Lack of support for community contributions / Small amount of tools / Other

Q6	<p>Continuing with the previous questions, and according to your experience, what other reasons would explain why there is no widely-accepted pattern repository?</p> <ul style="list-style-type: none">- The wide variety of problems to be solved and consequently of possible patterns- Too many pattern zealots with differing ideas of what a proper pattern and it's application is.- This is a very old problem. There is no business model with incentives for competent people to build and maintain such a repository- There is not an initiative that includes a reasonable amount of patterns. Today with the LLMs, maybe they can be a better alternative.- None needs it- Basically, 'pattern' itself (in Alexander's sense) is an orientation not stated by very many people. Most folks are into local (and fairly uncritical) narrative. Not many are into systemic ontology and plurality.- I guess a good format would be a curated wiki or encyclopedia but this is not a very popular format at the moment.- *answered in the previous "other:" section*- In Information technology and in particular in security, patterns are not part of best practice processes. Thus, there probably first needs to be a set of patterns that software and systems engineers can start using.- The c2.com pattern repository was subject to abuse; the replacement, federated wiki, was not embraced by the community.- Currently, many developers do not really use patterns, they don't even think about them.																											
Q7 ⁴	<p>Which of the following features would be important for you to use such a repository</p> <p>11 responses</p>  <table><thead><tr><th>Feature</th><th>Count</th><th>Percentage</th></tr></thead><tbody><tr><td>Good searching and classification</td><td>9</td><td>81.8%</td></tr><tr><td>Ability to download files that you could use to adopt the pattern (or other tool)</td><td>6</td><td>54.5%</td></tr><tr><td>Ability to rate the quality of the artefacts</td><td>3</td><td>27.3%</td></tr><tr><td>APIs and mechanisms for integration with other tools</td><td>7</td><td>63.6%</td></tr><tr><td>A conversational interface</td><td>1</td><td>9.1%</td></tr><tr><td>'Social media' in common use</td><td>1</td><td>9.1%</td></tr><tr><td>Possibility to link patterns to images</td><td>1</td><td>9.1%</td></tr><tr><td>more experience report than text</td><td>1</td><td>9.1%</td></tr></tbody></table>	Feature	Count	Percentage	Good searching and classification	9	81.8%	Ability to download files that you could use to adopt the pattern (or other tool)	6	54.5%	Ability to rate the quality of the artefacts	3	27.3%	APIs and mechanisms for integration with other tools	7	63.6%	A conversational interface	1	9.1%	'Social media' in common use	1	9.1%	Possibility to link patterns to images	1	9.1%	more experience report than text	1	9.1%
Feature	Count	Percentage																										
Good searching and classification	9	81.8%																										
Ability to download files that you could use to adopt the pattern (or other tool)	6	54.5%																										
Ability to rate the quality of the artefacts	3	27.3%																										
APIs and mechanisms for integration with other tools	7	63.6%																										
A conversational interface	1	9.1%																										
'Social media' in common use	1	9.1%																										
Possibility to link patterns to images	1	9.1%																										
more experience report than text	1	9.1%																										
Q8	<p>Please add other comments you would like to add to further illustrate why we still don't have an artifact repository and how it could be materialized.</p> <ul style="list-style-type: none">- Patterns are best shared through mentor/mentee relationships and practiced over a lifetime, much like martial arts. A static repository is not going to be very useful by itself.- I think we need to step back and ask if a repository is the best solution in the first place.- I cannot possibly imagine a single repository (especially in the highly 'untechnical' intersection of fields that I'm in). The challenge would be to handle the necessary plurality, across regions, languages, communities. Maybe you're thinking of more technical fields, like engineering or software production. But even in Alexander's field, I wonder whether there could ever be just one language (aka repo).- I strongly believe that communication is key, but not only that, initiative and the need to improve things; to move forward. For this to happen in the future, we first have to																											

⁴ Possible responses: Good searching and classification tools / Ability to download files that you could use to adopt the pattern (or other tool) / Ability to rate the quality of the artefacts / APIs and mechanisms for integration with other tools / Other

	<p>understand that there are issues, these issues will result in a "need" (the need to solve the problem/puzzle), which in turn will makes us try and find a most optimal solution. This solution is obviously the key point of this form, an artifact repository.</p> <ul style="list-style-type: none"> - Cryptographers focus on education, publishing research and developing software, including quite advanced compilers that generate bespoke protocols for individual needs. The problem that developers and designers first need to understand how their design has to be in order to benefit from these primitives has been largely ignored. This leaves a gap that could be filled by a cybersecurity design artifact repository. The first step can be to establish a project that develops and (more importantly) populates such a repository with advanced cryptographic primitives such as the once available in open-source libraries, for example in Bouncy Castle. - I offered to host a modern distributed repository for free. I know what I am doing in this regard, but the response was "give us an executive summary". I moved on. - I think we need a common position from community experts regarding this topic.
--	--