

The Value of Pattern Collections

Rebecca Wirfs-Brock¹[0009-0006-7215-3788] and Lise Hvatum²[0009-0007-3753-0063]

¹ Wirfs-Brock Associates, Sherwood OR 97140, USA
rebecca@wirfs-brock.com

² Independent, Ellsworth ME 04605, USA
lisehvatum@gmail.com

Abstract. Over the past 10 years there has been a growing awareness among software business and technology leaders that a product focus (which takes a longer-term view) instead of a project-based focus (which is mainly concerned with short-term delivery of results on time and within budget) can positively impact organizations, their software development processes, and business outcomes. In this essay we explore whether considering collections of patterns to be products might lead to radically different approaches to how they are initially conceived, and then developed and maintained over their lifetime. We ponder several questions: What exactly is the value proposition for pattern collections? What value do different people place on them? How might we increase their value?

Keywords: pattern collections, patterns, product lifecycle, value propositions

1 Introduction

This essay introduces a new perspective on patterns, considering them as products. Products that provide value to other designers. We think that this shift in mindset—from thinking in terms of shorter-term projects (that end with publication of the pattern papers) to longer lived products (that continue to provide ongoing value to their users)—will lead to a sharper focus during pattern discovery, documentation, and evolution. We also hope for improved outcomes for patterns and pattern collections developed with this framing.

Why look at value? Our more recent experience with software product development is influenced by product management principles drawn from a flow-based value creation approach as described in *Project to Product* [1].

Companies who take a product-oriented view of software are not simply concerned with short-term delivery of results on time and within budget. Instead, with a focus on products, they take a longer-term view. They do market research and listen to user feedback. They consider outcomes, not simply timely output. With such a focus, the experience of users becomes a key factor in how products are designed and implemented. User feedback is crucial for guiding product development. The goal is to create as much value for users while wasting as little effort as possible on unnecessary features. No fluff, no extra stuff. Just what the user needs and values.

What *would* be the outcome if we applied a user-centric product management mindset to patterns and their documentation?

To follow a lifecycle from inception to obsolescence, maximizing their value, patterns might need to be created and managed differently. Furthermore, when you consider patterns to be products which continue to provide value, it becomes evident that longer-lived pattern collections should be repaired and improved over their lifetime. It is our view that pattern documentation should rarely (never?) be conceived of as a one-and-done activity.

2 A Focus on Product Value

As explained in *Project to Product* [1], a product management approach runs counter to the industry focus on streamlining projects which has dominated the software industry since the early 1990s. The mindset promoted in *Project to Product* centers around principles and practices for strong ownership of products and their development. It has its roots in management practices that dominated software companies in the 1970s and 80s. A great read that reintroduces these earlier ideas is *The Phoenix Project* [2]. Written as a novel, it is a companion to the *DevOps Handbook* [3]. In *The Phoenix Project*, the protagonist is an IT manager who gradually comes to understand the benefits of paying attention to ongoing value creation—that is the outcomes—and not simply worrying about the efficiency of the development process—the outputs.

The main takeaway from these works is that to be successful, organizations need to concentrate more on the long-term ownership of their products and pay more attention to how the products are valued by their users. Successful product creation involves clearly articulating the assumptions about the product’s use and value, designing early versions to test these assumptions with a minimal product, and receiving feedback from users used to subsequently steer ongoing product decisions.

So how might this product management approach apply to pattern collections?

We recognize that the process of discovering and documenting patterns is quite different from the development of commercial software products. However, experienced pattern authors know that high quality patterns take time and energy and the involvement of experts to yield positive results. So, unless your pattern writing is driven by a need for personal fulfillment (as it may well be), your goal of creating a pattern collection is to create something useful to others—e.g., something of value to other designers, enhancing their ability to ultimately create products of high value for *their* end users.¹

However, if your goal is to produce papers, attend conferences like EuroPLoP, and get them published in the proceedings, then thinking of your patterns as products isn’t likely to provide you any new insights. The main benefit of taking on a project of

¹ The value proposition may be easier to see with UI design patterns where the outcome for the ultimate end user is more explicitly described. However, the quality of any software product constructed using patterns is also important to both its designers and end users—either directly through better ways users interact with it, or indirectly through the improved ability of designers and developers to maintain and extend their products’ capabilities over time.

writing a patterns paper for EuroPLOP is that writing forces you to organize your thoughts. You get to reflect on your experience. Incidentally, you receive feedback on your patterns from a shepherd and fellow writers' workshop participants. Although later on you might be pleasantly surprised to hear that someone else found your patterns to be useful, sharing your knowledge was never a primary goal. Your main focus was on short term output—a published patterns paper. And the main beneficiary of your efforts was you—you got to reflect on your experiences, learned how to write pattern descriptions, and received feedback from other authors on how to communicate more clearly.

However, if you have more ambitions for your patterns—perhaps to share your solutions to a larger audience and provide them useful information—then product thinking might help you achieve those goals.

3 Patterns as Products

A software designer's toolbox contains patterns, collections, pattern languages, design heuristics, and more. Patterns are just one among many tools we use. Some patterns are basic and fundamental; others have a specialized purpose. Some require significant experience and knowledge to use. Niche patterns are critical for special situations. Generally applicable patterns can be used more broadly. A good pattern can be adapted to fit into the context we are working in. A good pattern collection enables us to craft custom-made patterns (either by adapting existing patterns or creating new ones), which increases its value. And the ability to replace older patterns with newer, improved ones is important for keeping that collection relevant.

As we discussed the notion of patterns as products, we came to realize that what is most valuable to designers are pattern collections and not individual patterns. Furthermore, the overall impact of any particular pattern can be hard to measure. The value of a pattern collection comes in its wholeness and the interconnectivity of a group of related patterns. Those patterns fit within a context (domain) and can be used in combination to create larger design solutions. For example, to design and implement a monitoring system for industrial pumps, in addition to an understanding of the domain of pump management, you also need fundamental architecture solutions for data management, user interface design, etc. (found in pattern collections and other engineering resources), and a process for developing and delivering the system (perhaps found in pattern collections or other industry practices).

Given this insight, we decided to move forward in our “patterns as product” thought experiment with pattern collections as our primary focus. Individual patterns are useful in their own right, but what really brings value to a designer is the fact that patterns in a collection are used together, along with design practices and other patterns to create solutions at a larger scale. Although we will take a passing look at a value proposition for individual patterns, we think it is more important to consider what goes into the design of a pattern collection that provides value.

We view pattern languages as a special kind of pattern collection. While a pattern collection includes related patterns, it makes no claim to cover an entire problem

domain or offer strategies for combining those patterns. A pattern language, on the other hand, promises more—a cohesive and comprehensive set of patterns expressly intended to be used in combination to form larger design elements. Where we need to distinguish differences between pattern collections and pattern languages, we will make this clear; otherwise, when we refer to pattern collections, we implicitly include pattern languages.

Let us now take a deeper look at what it means for pattern collections to be considered as products. For the purpose of this essay, we will adopt the following definition of product from Nick Brown [4]:

A product is something that closely meets the demands of a particular market and yields enough value to justify its continued existence. It is characterized by its benefits, features, functions and uses that satisfy the needs of the consumer.

Let's break this down this definition into the separate qualities and explore how they relate to pattern collections:

It closely meets the demands of a particular market. The potential market—or *target customer*, if you will—for your pattern collection could be yourself, other individuals, colleagues, organizations, educators... And, depending on your understanding of your target audience, you will create and communicate your collection according to their needs.² During writers' workshops, we always remind pattern authors to add if they haven't clearly stated it: "Who is your target audience?" Pattern collections should be aimed at a specific set of users—whether it be UI designers, software security professionals or software architects.

It has Benefits. A good pattern offers a solution that has been proven to be useful, and that can be adapted by the designer. A good pattern collection has the added benefit of offering a broader set of related patterns. Based on feedback from users, patterns in the collection are revised, updated, and improved to better fit their needs. An extensible pattern collection is open, enabling users to enrich the collection with their personal and local (team) knowledge, to create alternative patterns and expand the collection, and to offer additional context-specific design guidance, all without "breaking" the collection.

It has Features. Key features of any individual pattern includes a tested solution that solves a recognized problem and advice on when to consider that solution to be viable

² Furthermore, we don't equate "meeting the demands of a particular market" with any guarantees of success. Many well-designed products (and patterns collection) do not reach their target audiences or have a significant impact due to forces outside of the control of their creators. However, the chances that a pattern collection reaches and positively impacts its intended audience can be improved if pattern authors put concerted effort into understanding their intended audience.

(e.g. describing the design forces and context). The features of a pattern collection are a set of named solutions that can be used in combination to solve larger scale problems.

It has Functions. The named patterns in a collection create a design concept vocabulary (language) that compresses design ideas and enables higher level design discussions.

It has uses that satisfy the needs of the consumer. This reflects the purpose of the pattern collection in providing useful solutions to other designers, allowing them to successfully create value for *their* end users.

We conclude that it makes sense to consider patterns and pattern collection as products, but are they *valuable* ones? What makes them valuable and to whom are they valuable? How should we discuss that value? How can we decide if the effort to document a patterns collection is worth the investment by authors, reviewers, and others involved?

4 A Value Proposition for Patterns

Let's define how we consider the value proposition of a pattern collection. A value proposition is multi-dimensional. A value proposition is what makes a product competitive or distinctive. Different values matter to different "customers." We decided to break down our understanding of value as it is perceived by these three roles³— the author, the designer, and the end user. The value proposition differs depending on who is considering it.

Pattern authors are often, but not always, designers in the domain of the patterns they write. Sometimes pattern authors identify patterns by observing and/or discussing what domain experts do to solve a problem.

Author:

- By documenting patterns, I gain a better understanding of how they work together to solve real problems (e.g., there is value in the act of writing).
- Interacting with my shepherd reveals deeper insights into how I can clarify my thoughts and enables me to communicate my knowledge in a written form.
- Getting feedback from others in a writers' workshop opens my eyes to issues that I was not aware of. This may be variations and other uses, other versions of these patterns, or ways they relate to patterns in other collections.
- I gain personal fulfillment by publishing my patterns.

³ There are other roles we could have considered—quality assurance or educators. But three roles are enough to illustrate the differences in what they value.

- The ultimate expansion of my understanding comes from not only my experiences with these patterns, but from others actively using the patterns and providing feedback that I can use to improve the pattern collection.

Designer:

- I use patterns in this collection frequently; they are part of my core toolset because they solve problems that I have to address in my designs.
- I understand these patterns, when to use them and when not to, how to tweak them to meet my needs, and the effects and side effects of using them.
- Using these patterns improves my design (it makes it easier to maintain/understand, more resilient/flexible/simpler...).
- I can talk about my design and design alternatives at a higher level (using pattern names).
- My colleagues are also using them, so these patterns offer familiar solutions for my team.
- Or, these are unfamiliar patterns that I discovered when researching design approaches and they hold a lot of promise for improving my design.
- And/or I use this pattern collection to teach less experienced developers about important concepts, effective practices, and design solutions.

End User⁴:

- The product that was created using these patterns is reliable and of high quality.
- Using this product satisfies my real needs, improving my life.
- The product does common things in a way I'm familiar with.
- The product is adapted to meet my evolving needs. Improvements are timely and based on my (and others') feedback.

At first glance, the most frequently used patterns might be assumed to be the most valuable. Why else would they be so prevalent? But just because a pattern has been used many times, doesn't mean it offers much value. A pattern may be used everywhere, it's necessary, but adds little to a design. Rather, a better measure of a pattern's value is how much it positively impacts the design or resolves larger or more fundamental problems. A highly valuable pattern goes beyond satisfying everyday (mundane) design requirements.

In the patterns community we focus on the quality of the pattern's solution, of the writing of the pattern's description, and of the analysis that goes into defining the pattern. However, we should not conflate pattern quality with pattern value. Pattern descriptions may be of high quality—they are well formulated, crisp in their explanation of the various elements (problem, forces, solution, etc.), have good

⁴ End users don't care about patterns per se; they care about the outcome of their application.

examples, all in all they fill the quality criteria we set for good pattern documentation. But unless they are useful in solving fundamental problems that designers actually struggle with, and used to generate outcomes that end users are happy about, that quality is not generating much value.

We should point out that as prospective pattern users, we can only assess the potential value of patterns by reading their descriptions or seeing how they have been put to good use. While great documentation is not sufficient to make a pattern collection valuable (and quite possibly superficially can make it look good), poorly documented collections can hide or obfuscate their true value. High quality, approachable pattern descriptions matter.

5 Creating a Pattern Collection using a Product Management Approach

Software products typically go through a lifecycle from inception through implementation, delivery, maintenance, and finally obsolescence. We don't find that the development of a pattern collection follows such a strictly linear process. However, we still find it useful to align concerns around developing a pattern collection with product lifecycle phases. A product manager preparing to start a new product, or more specifically an author starting to develop a new patterns collection, should think through a number of topics related to these phases.

We'll take a look at some questions to ask and challenges to consider in this context.

5.1 Inception

Just as for any software development process, it is better to consider main concerns up front and not treat them as an afterthought. The number one question is: Should you even develop this pattern collection?

Pattern authors are excited to share their experiences and knowledge through patterns. Pattern collections differ from other types of software products: patterns latently exist. Our product—that is our documented pattern collection—organizes and explains solutions to problems we've encountered.

If others don't share our enthusiasm, they simply may not find our patterns that interesting or relevant. Maybe they don't have the same needs. This is where a product mindset with a focus on outcome and long-term value can take you in new directions. Maybe you'll need to tweak your initial plan, or make new ones. Or find different customers with a better fit to your product.

Before investing a lot of effort, spend time with colleagues and professional communities to figure out what interest there is (e.g., user research). During this discovery process you will come to know whether your pattern collection is valuable enough to spend more time on.

If you find an audience of enthusiastic future users, ask them to share their insights and offer feedback. They might even become collaborators in your effort or the start of

an active user community. At some point, you need to decide whether there is enough interest to make the effort of creating your pattern collection worthwhile.

Early feedback also helps clarify what your product should be and help determine what it is not. You will eventually have to decide what content to deliberately exclude, thus creating a product boundary. Early feedback⁵ can also help prioritize your work. Start on the patterns that seem to have the most value. You will get more meaningful feedback if you state which patterns are core to your collection and show how your patterns relate to each other and to other patterns outside of your collection.

Software products are built over time (piecemeal growth), starting with the most basic functionality and adding new features (increments) gradually through iterative development. The term *minimum viable product* denotes an early version of a product—the simplest version that brings enough value—created specifically to get feedback from early users. Based on that feedback, we continue to evolve our product. This approach for creating a pattern collection—creating a *minimum viable pattern language*—also makes sense. Start with the core patterns. Test out their value and quality of their solutions. Then, continue to work with your user community. Improve and expand your collection based on their feedback and ongoing needs.

5.2 Design

How can you structure your pattern collection to make it available and inviting to potential users? You want to provide knowledge tailored to the different types of designers that you have identified during inception. They have different skills and backgrounds—their perceptions have been shaped by *their* past experiences. Experiences that can be quite different from yours. While we may share common design interests, our skills and backgrounds and what we perceive as important likely differs: “How, then, can pattern authors effectively share their patterned experiences with others? The challenge is to convey both their patterns’ significance along with the essence of the solutions they purportedly solve. Pattern descriptions could and should be a bridge between these different umwelts⁶.” [5]

We cannot emphasize enough the importance of building an active community of interest around your patterns from the onset. The strength of that community takes no glory away from you as an author. Patterns are about sharing experiences and providing solutions to known problems. Community involvement brings additional viewpoints and broader reflections on your patterns’ usage and value. It widens your perspective as an author. Maybe you can gather stories from the community and get help with the documentation effort. A community helps ensure that what you write reflects real needs, actual problems with viable solutions (validating product assumptions).

⁵ Writing a paper for a PLoP conference which outlines your collection, and perhaps presents one or two central patterns is one way to get early feedback.

⁶The word “umwelt” in German means “environment” or “surroundings.” We humans, like all living things, experience life in terms of our unique perceptions of our environment. Our individual umwelts are distinct.

Your goal should be to make your pattern collection materials as accessible and user friendly as possible—catering to the needs of your target audience. Even if your audience is quite experienced with the topic, they may not be familiar with your patterns. Your ideas are likely new to them. They’ll need some background. To grab their interest, include supporting information—such as a realistic example that illustrates a sequence of design decisions that use the patterns. State your design values. Show how your patterns can add value to their designs.

To avoid mind-numbing repetition, you may want to write an introductory section that discusses shared content common to many of the patterns—such as design values that underlie the patterns and common forces that affect these patterns. Or, you may determine that every pattern description should stand alone and be of value without having to reference other material. It all depends on your audience.

Consider the level of detail and how much domain knowledge that you expect other designers to have—should you include very basic knowledge or are you primarily documenting for experts? You may want to do a bit of both and write different pattern descriptions explicitly tailored for each audience.

A major strength of a pattern collection is its ability to be adapted and amended based on specific design needs. So how can you design the collection to enable this as well as share ongoing experiences, adaptations, and improvements with others?

There are many decisions to make about the design of your collection—ideally that you can test out on your community, and building on their feedback make necessary changes before committing to a final design.

5.3 Implementation

Implementation is where the rubber meets the road. Hopefully you have tested your ideas before spending a lot of time writing (implementing the product). If you started out by documenting your pattern collection as a set of papers submitted to patterns conferences, eventually you’ll have to revise your patterns and make the collection cohesive. You might create an online presence that is a meeting place for your community of designers. Or you could do both. Or something entirely different. Maybe you plan to publish a book. When you have collaborators, editors, and others involved, it can be productive to have a living (evolving) version of the patterns available online, or an online version of your book that is updated as your work progresses.

Creating this kind of documentation and support for a pattern collection takes significant time and effort. Your goals need to be realistic. How willing are you to commit to developing your product in the near term? What longer term plans do you have for your work? It is good to define a timeline and set some realistic end goal—whether it be a book, the handover of the pattern collection to a community, or a decision to stop the work based on elapsed time or life events.

Another important issue to address is intellectual property (IP) rights. If you’ve developed your patterns while working for a company, you need to respect company restrictions on IP (or obtain company approval). In addition, if you share the experiences of people in stories and examples, you need their consent.

5.4 Launch (Initial Delivery)

How do you get the pattern collection to other designers?

One characteristic of product success is that, “it yields enough value to justify its continued existence.” To provide value your collection must be used. To be used it must reach other designers. Alister Cockburn stated that, “patterns are written once and read never” [6]. We believe that getting the attention and reaching other designers is one of the biggest challenges that pattern authors struggle with. After your paper has been workshopped at a PLoP, it still isn’t a finished product. Papers at xPLop conferences should be viewed as a good opportunity to get quality feedback from pattern experts and fellow authors.

So, what should you do next? Software developers rarely read PLoP conference proceedings. Collection authors may spend several years writing several papers before the collection is “complete enough” for broader distribution.

One option for delivering this multi-year development effort to a wider audience is to write a book. Your earlier writing will require significant revision. Most likely you’ll find that your thinking has evolved as you wrote papers about your collection. At a minimum, you’ll need to revise and rewrite your patterns to remove inconsistencies. Conference paper lengths and formatting constraints have likely cramped your style. Worked examples, necessary background material, even the way you decide to organize, describe, and illustrate your patterns will need to be thought through.

There’s a lot of work involved in writing a book! And although software related pattern books continue to be published, these days software developers don’t buy or read many books. So, we believe that successful delivery of a pattern collection needs an online presence and support for a user community, in *addition* to a coherently published work—most likely a book only being part of that effort.

Rather than wait until your collection is fully featured, another viable option is to incrementally deliver your work, using multiple channels to get your patterns out there and noticed. As you write patterns, share them. Share them at local meetups. Create a website where work-in-progress on your pattern collection is visible. Invite feedback. Publish your work in small digestible installments—short videos or blog posts or worked examples. When you have enough, present your collection at software conferences. Find opportunities to reach more potential designers and get their feedback. In any case, some marketing will be needed.

5.5 Maintenance and Continued Delivery

The work of supporting a product doesn’t end with its initial delivery. However, we know of relatively few pattern authors who spend much time maintaining their collections, once they are published. They may not have thought through the implications of creating a collection that will need curation and updating to remain relevant. They may become too busy with other concerns (or life and work gets in the way).

Do you think that your work won't need (much) updating ? Or, is it closely tied to the current technical hype-cycle? (If so, it may have a short product life, and that's OK). Do you have concrete ideas about how to keep your patterns alive and useful? Are you expecting a publisher to help?

If you aspire to create a pattern collection with long-term value, consider how you might update it. How can inform other designers (your user community) about those updates? How might you engage them in your collection's evolution?

As for any product, if you are not collecting and analyzing usage data, you are operating blind. Stuart Brand, in *Maintenance: Of Everything Part One* [7], points out that while maintenance is what keeps everything going, paradoxically it is all too easy to put off. Rather than shirking necessary maintenance (because it may be tedious or boring or difficult), Brand challenges us to reframe our thinking by "expanding the term 'maintenance' beyond referring only to preventative maintenance...Let 'maintenance' mean the whole grand process of keeping a thing going. From that perspective, occasional repair is part of the process. Close monitoring is part of the process...Eventually replacing the thing is part of the process." Maintenance is what can keep pattern collections alive and useful.

Over time and through practice, users of your patterns form their own connections between your patterns. As they adapt them and fold them in with their own design patterns and practices, they may gain new insights. If you create a forum that enables them to freely share these experiences, then you and your community have the opportunity to collectively grow and evolve the collection.⁷ Innovations from designers using your patterns could feed into your collection's ongoing maintenance.

5.6 Obsolescence

How do you remove a collection when it is no longer needed, when it has been superseded by a better product, or when you realize that its quality is not what it should be?

Quite likely, this is only an issue for those few pattern collections that have had the greatest successes. Most software pattern collections slip into oblivion without much notice. And yet, planned obsolescence is an often overlooked but important phase of any product's lifecycle. When no longer relevant, pattern collections should be deprecated.

Whose responsibility is it to deprecate patterns or pattern collections? It should be clear who owns the collection and who can take decisive actions. The original author(s) may no longer be involved or may have moved on. Without clear ownership, taking decisive actions is difficult. We'd like to point out that pattern descriptions may be copyrighted (by authors or publishers), but patterns themselves are not and cannot be. There will be changes made to pattern collections. Some patterns will be revised (or become obsolete). Others will be added. Unless there is a community surrounding a

⁷ It is rare that pattern collection authors have the time or energy or means to create and grow a community around their work without the help from others.

collection, and a centralized place to collect this evolving knowledge, these updates will become scattered.

6 Experiences from Existing Pattern Collections

We start this section with a disclaimer—this is not meant to be a deep dive analysis of pattern collections. To do so, would require substantial research on our part. This is beyond our capacity and out of the scope for this paper.

That said, we did reach out to Ward Cunningham to ask about the history of the Portland Pattern Repository (personal email communication, April 27, 2026). The Portland Pattern Repository [8] was the first effort within our community explicitly designed as an open, interactive, collaborative space for developing and communicating software patterns. Over a period of twenty years, the wiki-based platform created by Ward Cunningham, became a vibrant collaborative space for patterns development and discussion.

Its origin story is interesting: As program chair of the first PLoP conference, Ward was encouraged to host a pattern repository. At the conference he also learned about a new form of online publishing for World Wide Web pages called HTML. Returning home, he built the first Portland Pattern Repository from scratch using hand-crafted HTML. Quickly recognizing that it would be too cumbersome to continue adding pattern descriptions by hand, Ward wrote a perl script to automate pattern submission. According to Ward, “authors would have to learn the rules, and they could, if they wanted their work to look good.” After inviting a dozen pattern insiders to try out his system (only half took up his offer), a week later he announced the site, “called ‘wiki’, on the 500-member software patterns mailing list. From there it grew very quickly.”

Behind the scenes, Ward spent two to three hours a day curating Portland Pattern Repository content. His diligent (but largely invisible) work had a big payoff: “I wrote fancy scripts that would analyze site traffic and could devote my attention to pages that were proving to be important. Most of my work was unsigned, this gave the illusion that wiki somehow made writers more careful and consistent.” Site traffic grew and grew and grew, providing a new generation of pattern and software design enthusiasts an inviting platform for sharing and developing their ideas. Eventually the maintenance effort for the repository became so much that Ward recruited “stewards” to help. They curated the Portland Pattern Repository contents using increasingly advanced analysis tools to aid their efforts. Sadly, after 20 years, the site was closed to any further content after a hacker threatened to take down the whole site.

After thirty-two years in print, the twenty-three patterns in *Design Patterns: Elements of Reusable Object-Oriented Software* [9] (affectionately known as the GOF patterns) are the most widely known software design patterns. They were the first software patterns book published. The timing was right. In a pre-release review Rebecca wrote for the publisher, she expressed a strong desire that *Design Patterns* be produced and delivered in a way that allowed for yearly updates and extensions. But this was not to be. Book sales wildly exceeded everyone’s expectations. Consequently,

these patterns reached a massive audience, and have been broadly applied in industry and frequently taught in university level software engineering courses.

After the lead author, John Vlissides, died the other authors did not update the original collection. Over time the patterns' significance has diminished. The value of several patterns have been questioned (e.g., No one should ever use the Singleton pattern! Why create an Iterator when you can use the built-in programming language features for iteration. Well, even though there is built-in language support, it still is a pattern.). The patterns are still widely used. Shorter, simpler descriptions have been written (for example, *Head First Design Patterns: A Brain Friendly Guide* [10], and the updated *Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software 2nd Edition* [11]). Books have been written showing programming language specific (for example, *Design Patterns in C#* [12] or *Design Patterns and Best Practices in Rust* [13]).

The *Fearless Change* pattern collection [14] was developed by Mary Lynn Manns and Linda Rising. They engaged with many people as they developed and evolved their patterns over a number of years. They brought work in progress to PLoPs and shared their patterns at other conferences. Eventually they formed a community specifically to empower grassroots organizational change agents. After the first book was published, there still was a lot of ongoing interest. People who were exposed to their early material were eager to contribute to the collection and share their stories. Subsequently, a second book was published [15], which updated several of the original patterns and added new ones. The authors maintain a website [16] and hold an annual virtual campfire event, where Fearless Change patterns users are invited to tell inspiring stories about how they've used these patterns. The patterns are stable, and it is unclear whether further contributions or refinements will be made.

The microservices pattern collection developed by Chris Richardson was initially developed within a closed ecosystem and promoted on his website [17] <https://microservices.io>. Chris came to a PLoP before his book [18] was published to share and discuss with interested attendees. In 2018, when the pattern collection book was first published, microservice technology was relatively new. The collection was prominently featured on the website and enthusiasts eagerly commented on them and asked questions. A second edition of the patterns collection is in the works [19] adding new patterns for faster flow and delivery. As the pattern collection has evolved and the use of AI technology has become increasingly important, the focus of microservices website has broadened to include how to effectively use AI technology when designing and implementing microservice architectures. It is unclear how much of an impact AI technology will have on the evolution of this pattern collection.

Not only can technology advancements impact the relevance of software design collections, they also change our development practices and processes. At VikingPLoP 2026, three of the Scrum Pattern collection authors led a focus group [20], "Are Scrum Patterns ready for Human-AI Collaboration?" Participants were asked to examine whether classic Scrum Patterns might break down under Human-AI collaboration, identify candidate patterns that might need revising, as well as those that seemed stable and did not need revision. Stay tuned.

Each collection involves a significant effort on the part of the authors to create.

Original authors retain ownership, but move on to new endeavors. What may have been an avid user community moves on too. With no clear feedback mechanisms to guide revisions to the original published works, that work can grow stale. Others may write books or blog posts, updating a collection, but this knowledge becomes scattered and fragmented. Most software related pattern collections that we know of eventually end up being frozen in time (or plateau) for a variety of reasons. And most collections have limited impact (outcome) because they are not living products that are adopted by a user community that enriches them and brings them into the future.

However, we aren't discouraged.

We see glimpses of product thinking in each of the above collection stories. This gives us some hope. How *Fearless Change* patterns were developed comes closest to user research. The Portland Pattern repository enabled quick and ongoing feedback of nascent patterns. *Design Patterns* has been rewritten to cater to different audiences' needs, just not by the original authors. And microservice patterns seem to be evolving to stay in alignment with technology advances.

We think any pattern collection author should address several questions should be addressed before embarking on any significant effort: Are you willing and able to take on the responsibility of owning your work and its evolution, treating it as a long-lived product? Are there enough benefits for you and your community in doing so (even if your main reason for creating your collection is the love of your domain and a desire to share your knowledge)? What help and support might you need and want from others to curate your patterns and manage future revisions in support of your community?

7 Closing Remarks

This essay was a thought experiment. We hope it stimulates further discussion and experimentation by our patterns community friends. The desire for pattern collections to be useful and alive (and long-lived) is a recurring topic of conversation when we hang out to discuss "world issues" at xPLoP conferences. We believe that there is value in considering pattern collections as products. Perhaps collectively, as a patterns community, we can figure out some ways to better support pattern collections (and their authors) throughout their product lifecycle.

Acknowledgments. The authors would like to acknowledge our shepherd, Klaus Marquardt. Klaus generously gave us his time and sound advice and encouragement. We also acknowledge the participants in our writers' workshop who graciously contributed their diverse opinions and reactions to our thought experiment and thoughtfully critiqued our work.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this paper.

References

1. Kersten, M. Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework. IT Revolution. (2018)
2. Kim, G., Behr, K., Spafford, G. The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win. IT Revolution. (2024)
3. Kim, G., Humble, J., Debois, P., Willis, J., and Forsgren, N. The DevOps Handbook, 2nd Edition: How to Create World-Class Agility, Reliability, & Security in Technology Organizations. IT Revolution. (2021)
4. Brown, N. "Project to Product nudges for large organisations." March 4, 2022. Retrieved April 29, 2026 from: <https://medium.com/nationwide-technology/project-to-product-nudges-for-large-organisations-e2867d4a3fd2> (2022)
5. Wirfs-Brock, R. and Kohls, C. "Patterns and their Umwelt – How to make patterns come to life" in Proceedings of 32nd Conference on Pattern Languages of Programs, People, and Practices (PLoP). (2025)
6. Cockburn, A. Keynote: "The Take-Home Value of a Pattern is Its Name & Other Observations from 30 Years of Patterns Use" at the 30th Conference on Pattern Languages of Programs (PLoP). (2023)
7. Brand, S. Maintenance: Of Everything: Part One. Stripe Press. (2026)
8. Portland Pattern Repository. Available at: <https://c2.com/ppr/>. (Accessed: 26, August, 2026)
9. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. (1995)
10. Freeman, E. and Robson, E. Head First Design Patterns: A Brain Friendly Guide. O'Reilly Media. (2004)
11. Freeman, E. and Robson, E. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software 2nd Edition. O'Reilly Media. (2021)
12. Metsker, S. Design Patterns in C#. Addison-Wesley Professional. (2004)
13. Williams E. Design Patterns and Best Practices in Rust: Enhance your Rust skills by applying idiomatic approaches to real-world software design. Packt Media. (2026)
14. Rising, L. and Manns, M. Fearless Change: Patterns for Introducing New Ideas. Addison-Wesley. (2004)
15. Manns, M. and Rising, L. More Fearless Change: Strategies for Making Your Ideas Happen. Addison-Wesley. (2015)
16. Fearless Change Website. Available at: <https://fearlesschangepatterns.com/> (Accessed: 26, August, 2026)
17. Microservices.io Website. Available at: Fearless Change Website. Available at: <https://microservices.io> (Accessed: 26, August, 2026)
18. Richardson, C. Microservices Patterns: With examples in Java First Edition. Manning. (2018)
19. Richardson, C. Microservices Patterns: Second Edition. Early access version available: <https://www.manning.com/books/microservices-patterns-second-edition>. Manning. (2027 est.)
20. Aguiar, A. and Eloranta, V-P. Focus Group: Are Scrum Patterns ready for Human-AI Collaboration? VikingPLoP 2026. (2026)