Chapter 9

Introduction to the PBE Patterns and Guidelines

This chapter opens Part II of the book, which discusses the PBE Patterns and Guidelines.[1] The PBE Patterns and Guidelines are to be used in conjunction with the PBE Core Values introduced in Chapter 1 and the PBE Practice discussed in Chapter 8 and Appendix F.

Why patterns and guidelines? PBE includes a set of patterns that have surfaced over the years. It makes sense to discuss patterns that can help us follow PBE; think of them as metapatterns. Also, it is important to see that new patterns are discovered and created; we are not restricted to using patterns that have been discovered by others. Patterns are for everyone—we all need to be on the lookout for pattern opportunities.

Guidelines are also needed to provide advice on how to successfully apply PBE. Not everything needs to be or should be a pattern. We need to evaluate and review patterns to ensure that they are worthy of the name and add value.

Relationship of Patterns and Guidelines to Other Elements within PBE

As a quick refresher, let's review a short definition of PBE and a summary of the PBE Core Values. First, recall that we define PBE as

> a systematic, disciplined, and quantifiable approach to software development that leverages the use of pattern implementations and specifications throughout the software development and delivery process.

[1] A set of graphics depicting the collection of PBE Patterns and Guidelines is provided in Appendix D.

And, when following a PBE approach, we embrace the following core values:

1. Patterns are best used in combination, rather than in isolation.
2. Always identify and build new patterns.
3. Patterns can be built and used within the same project.
4. Make your patterns live.
5. Focus on making patterns consumable.
6. PBE can fit into many different development processes.

The patterns and guidelines, as shown in Figure 9.1, build upon and leverage the PBE Core Values, while providing details on how we realize the tasks and deliverables associated with the PBE Practice. In this chapter and those that follow in Part II we will discuss the PBE Patterns and Guidelines.

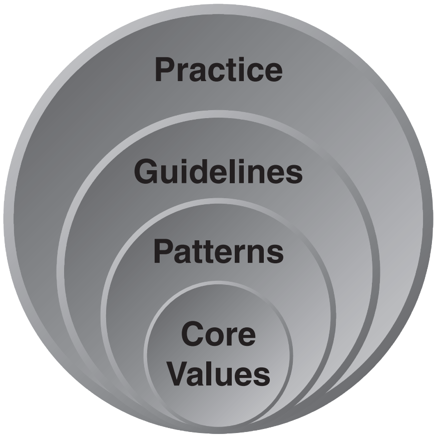

Figure 9.1 *A model bringing together the key elements that support PBE efforts*

Quick Guide to PBE Patterns and Guidelines

In this section we've assembled a quick guide to the PBE Patterns and Guidelines. As discussed earlier, a pattern is a proven best-practice solution to a known recurring problem within a given context. A guideline is less formal; essentially it is a piece of advice or a general rule.

The PBE Patterns and Guidelines are organized into a set of categories, as follows:

- **Foundational.** This is the base set of patterns and guidelines. They form the foundation of our PBE efforts.
- **Discovering patterns.** This set of patterns and guidelines focuses on the discovery of new patterns. To find patterns we need to have the proper skills and focus.
- **Designing patterns.** This set focuses on patterns and guidelines related to the design of patterns.
- **Creating patterns.** With a design in place, these patterns and guidelines focus on successful pattern creation.
- **Packaging patterns.** Once created, patterns need to be packaged and made available for reuse.
- **Using DSLs and patterns.** There is a strong connection between the use of patterns and DSLs. These patterns and guidelines help us in using these two mechanisms together.
- **Consuming patterns.** With thousands of patterns already available and more being identified, we also need to focus on how we can successfully consume patterns. Although just one of the seven categories, it is one of the most important. Our patterns will have little value if we fail to consume them successfully.

The following sections provide an overview of each category along with its patterns and guidelines. The goal is to provide both a summary view and a quick reference.

Chapter 10: Foundational

This chapter focuses on the base, foundational PBE Patterns and Guidelines. These patterns and guidelines will help you incorporate PBE into your projects, guiding your focus, scope, and approach.

Patterns

End-to-End Pattern Use
Problem: How do we accelerate the speed of the project and the quality of the resulting artifacts? And most important, how do we do so across all of the phases and disciplines within the SDLC?

Solution: Use patterns with an end-to-end focus across the entire SDLC and the associated set of tasks and tooling.

Piecemeal Pattern Creation
Problem: How and when do we determine what patterns to use, which ones need to be built, and when and where they should be used?

Solution: Take a piecemeal approach to creating patterns. Analyze and prioritize the development of the pattern features, then map that prioritized effort across a set of iterations and incrementally deliver the needed pattern functionality.

Simple Solution Space
Problem: We use a combination of generic and extremely rich languages to define a solution. In addition, we often work at multiple levels of abstraction. How do we boost productivity, improve quality, and support governance while working in a solution space that is complex?

Solution: Simplify. Reduce the number of moving parts. Review and reduce models, DSLs, diagrams, and elements within diagrams. To assist and accelerate the effort, ensure that you are finding, creating, and applying the correct set of patterns.

Single Pattern–Varied Use Cases
Problem: How do we ensure that we are fully leveraging the investment made in a pattern and that we magnify the impact of the pattern?

Solution: Use a pattern to support multiple use cases in the development effort. Think in terms of the various ways in which you can use the pattern.

Guidelines

Domain-Driven Patterns
Summary: Create, acquire, and use patterns focused on meeting the needs of a domain as well as providing a bridge between domains. Typically, we would look to patterns in the business domain, technology domain, and crossing between these domains.

Pattern Search
Summary: We know that we should be using patterns as we build our solutions. However, it is hard to find patterns to address a problem. Where and how do we find the patterns that we should be using? The short and simple answer is that we need to search for them.

Chapter 11: Discovering Patterns

This chapter focuses on patterns and guidelines that will help in discovering patterns within an organization.

Patterns

Antipatterns

Problem: How do we capture and then share common worst practices with the team? How do we ensure that the team is aware of and then avoids duplicating these worst practices?

Solution: Ensure that you are identifying, detailing, and then communicating the details associated with antipatterns, proven worst-practice solutions applied within a given context.

Pattern Opportunity

Problem: How do we identify the opportunities for using patterns on a project?

Solution: Opportunity identification is all about looking for and finding places where patterns can be used. Tune the creativity and sleuthing skills of the team to be on the lookout for pattern opportunities.

Recurring Solution

Problem: Although we want to use patterns that are specific to a situation and are looking for them, we are not sure when and where the best opportunities to create patterns exist. Where do we find patterns?

Solution: Look for recurring implementations of solutions as a key mechanism for creating patterns. You can use a number of approaches and guidelines such as the Rule of Three, déjà vu, experience, and so on.

Guidelines

Determine Business Impact

Summary: Creating patterns for reuse requires an investment; we cannot build everything, so how do we pick which patterns justify the expense? Develop criteria and mechanisms that can be used to help decide and direct investments.

Pattern Description

Summary: How do we capture a sufficient level of detail about a pattern to support evaluation and feed into design activities without overinvesting in these efforts? Create a pattern description to capture initial details of a pattern that can be used to evaluate the pattern and then as a starting point for designing the pattern.

Pattern Harvest

Summary: How do we capture and elicit best practices, in a more consumable form, from completed and proven projects? Use harvesting to capture the details of the best practice and put them into a pattern.

Update Existing Patterns

Summary: Best practices are evolving as the technology and the practices get spread out. How do we ensure that the collection of patterns within our organization continues to represent best practices? Update existing patterns as necessary to adapt to new technology or to evolving practices.

Chapter 12: Designing Patterns

This chapter focuses on the patterns and guidelines that assist in designing patterns.

Patterns

Compound Pattern

Problem: In situations where we have a number of patterns that work together to solve a related set of problems, how do we simplify the user experience and increase the value and power of the patterns? How do we ensure that related patterns are used together consistently and correctly?

Solution: Create a pattern that is a composite of other patterns. Combine patterns into a larger-grained unit that can easily be consumed by the Pattern User. As a result, the Pattern User needs to know only how to apply the compound pattern rather than the set of patterns encapsulated within.

Exemplar Analysis

Problem: How do we go from an exemplar to a pattern?

Solution: Perform an analysis of the exemplar. Identify the key entities involved in the pattern, the points of variability, a definition of the information that needs to be provided by a user of the pattern, the rules to be used to calculate any additional information needed by the pattern, and the aspects of the pattern that stay the same across applications of the pattern.

Meet-in-the-Middle Pattern Design

Problem: How do we successfully transition from the point where we have identified a pattern to where we have produced a usable pattern?

Solution: Follow a meet-in-the-middle approach to produce the pattern. Such an approach is pragmatic and practical and encourages focusing on the details of the solution as well as the higher-level issues, such as how the pattern should be written up, how it is consumed, and so forth.

Pattern Implementation

Problem: How do we speed up the use of a pattern in a solution, while ensuring that it is applied correctly and consistently?

Solution: Automate the pattern via the creation of a pattern implementation. In contrast to a pattern specification, which captures a pattern as formal written documentation, an implementation codifies a pattern in tooling, automating the use of the pattern.

Guidelines

Limited Points of Variability
Summary: When creating a pattern, how do we ensure that the user of the pattern is not overwhelmed and confused by the configuration and customization options for the pattern? Limit the number and types of points of variability. Ensure that the pattern provides the most important points of variability and then uses calculations and assumptions to leverage the user-provided information.

Pattern Creation Lifecycle
Summary: There is a flow in which we use existing patterns to help us build a solution that will serve as an exemplar. With an exemplar in hand, we can use abstraction and analysis to create a set of one or more patterns. We then use those patterns in the creation of future models that will serve as the basis for future patterns.

Pattern Implementation Extensibility
Summary: How can we facilitate the use of a pattern implementation in situations for which it was not originally intended? When designing a pattern implementation, ensure that the tooling used provides support for extensibility. Then, as part of the design, consider how and where to enable others to extend the solution.

Team Pattern Implementation Use
Summary: Software development is a team sport, requiring the efforts of many to succeed. How do we design pattern implementations to support team development scenarios? Consider the reapplication strategy, management of user-provided content, model structuring and merging support, and the languages that will be used.

Chapter 13: Creating Patterns

This chapter presents a set of patterns and guidelines that focus on providing guidance for creating patterns.

Patterns

Model-to-Model Pattern Implementation
Problem: When using models at multiple levels of abstraction, how do we support the transition between the different models while following best practices?

Solution: Create model-to-model pattern implementations that automate the transformation of a model at one level of abstraction to a model at another level of abstraction.

Model-to-Text Pattern Implementation

Problem: How can we transition from representations in models to text-based artifacts that directly represent the solution while following best practices?

Solution: Create model-to-text pattern implementations that take a model as input and then generate a set of text-based artifacts. These text-based artifacts can include code, documentation, scripts, and other elements.

UML Pattern Implementation

Problem: When using patterns within a UML model, we run into a number of challenges. How do we accelerate the application of patterns and ensure that they are applied correctly and consistently?

Solution: Create a UML pattern, that is, a pattern implementation that is specialized for use within UML models.

Guidelines

Automate Creation of Pattern Implementations

Summary: How can we reduce the amount of time it takes to build high-quality pattern implementations? Create and use pattern implementations to automate the creation of pattern implementations.

Pattern Specification

Summary: How can we quickly and efficiently document and communicate a pattern so that it can be reused? Write a pattern specification detailing the pattern name, context, problem, forces, solution, examples, related patterns, and other information.

Pattern Testing

Summary: How do we ensure that a pattern is consumable and of high quality? The simple answer is to test the pattern to ensure that it is consumable, meets quality expectations, and satisfies requirements.

Chapter 14: Packaging Patterns

This chapter presents a set of patterns and guidelines that provide guidance on how to package patterns to ensure that patterns can be found and consumed.

Patterns

Embedded Pattern Implementation Guidance
Problem: How can we eliminate hurdles and friction (and excuses) related to not following documented guidance and best practices?

Solution: Embed guidance on how to use the pattern into the tooling that hosts the pattern.

Provisionable Pattern Implementation
Problem: How do we support the deployment and provisioning of patterns to the Pattern User's work environment?

Solution: Leverage the capabilities of the tooling to create manageable units for the patterns and their associated artifacts.

Guidelines

Document Pattern
Summary: How do we ensure that the Pattern User is able to understand the details associated with how to install, configure, use, and uninstall a pattern? Document the pattern, providing the Pattern User with details on how to work with the pattern.

Document Pattern Relationships
Summary: How do we communicate which patterns should or should not be used with one another? Document the relationships between patterns, including dependencies, hierarchy, and cases where patterns are mutually exclusive.

Make Pattern Documentation Easily Accessible
Summary: How do we make a pattern easy for others to understand and work with? Make the pattern documentation that supports the pattern easy to find and access.

Package Related Patterns Together
Summary: How do we help Pattern Users to succeed in consuming multiple related patterns? Simplify how people find, consume, and work with related patterns by packaging them together.

Pattern Packaging
Summary: How can we share patterns in such a way that they are easy for others to consume? Package patterns so that they bring together the appropriate set of artifacts needed for consuming the pattern.

Pattern Version
Summary: How do we succeed in an environment where multiple versions of patterns exist? Use version numbers and a versioning strategy to organize and support the various instances.

Use Models in Documenting Patterns
Summary: How can we improve and simplify the documentation regarding the details about the design and use of a pattern? Provide visual models that show the details of a pattern and how to use it.

Chapter 15: Using Domain-Specific Languages with Patterns

This chapter focuses on patterns and guidelines to help in creating a simplified modeling experience for those using the patterns.

Patterns

DSL Model Template
Problem: How should models using a DSL be structured?

Solution: Provide the users of the modeling language with a model template to supply a starting structure and guidance.

Integrated Patterns and DSLs
Problem: How can we raise the level of abstraction and have the team focus on the problems to be solved while preventing them from getting lost in the details of the development languages?

Solution: Use DSLs within a PBE approach to enable working at higher levels of abstraction, a level closer to the solution space, while also leveraging best practices in the form of patterns.

Guidelines

Create a DSL
Summary: How do we take a design as input and create a DSL? Use a framework to help build the implementation, using either a specialized language creation framework or a general-purpose language that has extension capabilities.

Design a DSL
Summary: How do we design a consumable and high-quality DSL? Use a meet-in-the-middle approach to building the DSL that accounts for language scope, granularity, semantic completeness, user friendliness, and tool friendliness.

Meaningful Icons in a DSL

Summary: How do we make the DSL more intuitive, user-friendly, and easy to work with? Use meaningful and easy-to-understand icons to provide visual cues to those using the DSL.

Chapter 16: Consuming Patterns

This chapter focuses on a set of guidelines to help get the most out of using patterns. We change our focus to look at how the Pattern User role can succeed in using patterns.

Guidelines

Communicate Design with Patterns

Summary: How do we describe and communicate key aspects of a design to others? Use patterns to describe and communicate key design aspects of a solution.

Design Solutions with Patterns

Summary: How can we improve the delivery and the quality of applications while leveraging best practices? A key and traditional approach is to use patterns in designing the solution.

Pattern Density

Summary: How many patterns should be used in creating a solution? Manage the density of patterns that are used within a solution as a means to using the correct number of patterns.

Pattern Selection Driven by Requirements

Summary: How do we determine what patterns should be selected and used on a project? Use a combination of functional and nonfunctional requirements as a guide in selecting the patterns to use.

Refactor with Patterns

Summary: How do we improve an existing solution to align with best practices? Refactor the existing solution using patterns as the means and endpoint for the updated solution.

Select Large-Scope Patterns First

Summary: When it appears that it is possible that many patterns could/should be used in a solution, where should pattern selection start? Start with the patterns that have the largest impact or scope and then use smaller-scope patterns to embellish the details.

Use an Asset Repository
Summary: How do we quickly and easily find the patterns that are available for reuse within the organization? Use an asset repository to manage the patterns like any other reusable asset.

Use Pattern Definitions to Understand Existing Solutions
Summary: How do we quickly understand an existing solution? Is it well built? Has it stayed true to the design? Using a set of known pattern definitions, search the solution for occurrences of the patterns.

Use Patterns to Find Patterns
Summary: How do we find new patterns in existing solutions? Using the collection of known pattern definitions, analyze the existing solutions to find occurrences of new patterns.

Summary

We can use the PBE Patterns and Guidelines to help us identify, produce, consume, and manage patterns. This chapter has provided an overview of all of the PBE Patterns and Guidelines. When we initially review the list, it serves as an introduction to the set and the support provided for our PBE efforts. Once we've gained some familiarity with the patterns and guidelines, we can use this chapter as a reference that can quickly and easily be consulted.