Ten Design Pattern Myths

Jim Fawcett

condensed from

Pattern Hatching, John Vlissides,

Addison-Wesley, 1998

Myth 1 - A Pattern is a Solution to a Problem in Some Context

- Yes, three essential parts of a pattern are Problem, Solution, and Context, but three other things are also required:
- Relevant Recurrence:
  This solution is relevant in other situations.
- Vehicle for Teaching:
  The solution conveys some principle or observation worth passing on to others. It is not immediately obvious, nor is it so complex that there is little value in passing on.
- It must have a Name:

Myth 2 - Patterns are just rules, tricks, and data structures

- Patterns introduce few new terms
  - A good pattern is inherently accessible to its audience.
- Patterns are not rules you can apply mindlessly nor are they limited to programming tricks
- Each pattern captures a useful principle or method that can be used "as-is" or bent and modified to suit your current design needs.
- Most, but not all, patterns focus on issues that are independent of a specific programming language.
- Structure may capture the essence of a pattern but:
  - structure usually relates to module and/or class relationships
  - the value of many patterns lies in their description of dynamic, not static, relationships. That is, collaborations are more important than structure.

Myth 3 - Seen one, Seen them all

- Patterns come in a variety of application domains, content, scope, and styles. Patterns are used to capture:
- Design paradigms
  - Object-Oriented Design techniques
- Programming strategies
- Domain specific techniques
  - database, distributed processing, …
- Specific technology issues
  - security, component models, …
- Architectures for user interfaces
- Management strategies

Myth 4 - Patterns need tool or Methodology Support to be Effective

- The benefit from patterns comes mostly from applying them as they are - with no support of any kind.
- Patterns provide a way to pass on expertise from the seasoned expert to the entry level developer.
- Patterns also provide a means to save, catalog, review, and reason about technical issues and really bright ideas.

The benefits of patterns are:

- they capture expertise and make it accessible
- their names form a vocabulary that makes communication precise
- they help people understand a system more quickly when it is documented with the patterns it uses
- they help to restructure a system by helping us to focus on the main issues

Myth 5 - Patterns Guarantee Reusable Software, Higher Productivity

- Patterns don’t guarantee anything. They don’t even make benefit likely.
- Patterns do nothing to remove the human from the creative process.
- Patterns bring the hope of empowerment to the uninitiated but otherwise capable and creative person.
- Patterns are just another weapon in the developer’s arsenal.

Myth 6 - Patterns Generate Whole Architectures

- Patterns themselves don't generate anything. People do - and they do it well only if they and the patterns they use are effective.
- Patterns are often catalysts that set the path to a complete architecture. Use of patterns can establish a strong foundation on which to build.
- Patterns are unlikely to cover every aspect of an architecture. Any non-trivial design has lots of aspects that no pattern addresses. It's up to you to fill in the spaces with your own creativity.
- The intent and especially motivation sections should deal with the "forces" at work in a design problem and the "resolutions" of those forces that the pattern provides.
  - A clear view of the motivation behind a pattern helps you to bend it to the needs of your current application.

Myth 7 - Patterns are only for Object-Oriented Design/Implementation

"Patterns are nothing if they don't capture expertise.

- The nature of that expertise is left open to the pattern writer."

Patterns are useful in:

- analysis
- Design
  - functional, structured, declarative, object-oriented, …
- implementation and maintenance
- testing
- documentation
- organization and management

Myth 8 - There’s no Evidence that Patterns Help Anybody

- “People are reporting benefits from patterns in journals [Software - Practice and Experience] and conferences [OOPSLA].”

- Pattern Languages of Program Design conference proceedings are filled with scores of papers describing benefits of applying and modifying existing patterns as well as reporting new pattern discoveries.

Myth 9 - The Pattern Community is a Clique of Elites

- The four authors of the Design Patterns book are from university faculty, research lab, and industry.
- Most of the contributors to the PLOP conferences are working software developers in industry.

Myth 10 - The Pattern Community is Self-Serving, even Conspiratorial

- "…, I can confirm that we [authors of Design Patterns] were as surprised as anyone by the reaction to Design Patterns … even the publisher was caught off-guard by the demand."

- "… if you read the works of leading pattern authors carefully, you'll sense a common and overarching desire: to take hard-earned expertise, best practices, even competitive advantage -- the fruits of years of hands-on experience -- and not just disclose it but *impart* it to all comers."