

The Pros and Cons of Adopting and Applying Design Patterns in the Real World

Tina Andersen

Benefits

Coordinate process and community: Uniformity allows the developers to communicate at a higher level of abstraction.

Can be used reactively: Documentation tool to classify the fragments of a design.

Can be used proactively: Build robust designs with design-level parts that have well understood trade-offs.

Can give the software a hinge: Future changes consistent with a hinge are relatively inexpensive.

Benefits continued

Can turn a trade-off into a win-win situation: Adaptability must be explicitly designed into the software in designated places.

Constrain maintenance programmers: Require maintenance programmers to understand and preserve the integrity of the design patterns during maintenance changes.

Let management reward self-directed designers: Differentiating self-directed developers from the others is an important management problem.

Inhibitors

Oversold: The overall software development process has to be modified to take design patterns into account.

Unnecessarily difficult to learn: Murky, obfuscated design patterns tend to distract people from the clarity of many writers and the depth of their design patterns.

Classifications are not yet useful for practitioners: Some categories are not helpful to software practitioners while they learn patterns.

Marshall's conclusion

Benefit!

My own conclusion

I have no doubt that one will get benefit from successfully adopting and applying design patterns. The question is just what is the cost in the process?