



8 Jahre Erfahrung mit Patterns

Wie können Sie von Patterns profitieren?

Wolfgang Keller, Plattform-Management,
Generali Office Service & Consulting AG, Wien

Vorlesung TU Wien, 24.4.2002

Email: wk@objectarchitects.de
<http://www.objectarchitects.de/>



Ausgangslage für Ihre Vorlesung heute ..

- Sie wissen, was Patterns sind ..
 - Sie hören ja gerade eine Vorlesung darüber ..
- Sie haben schon ein paar gesehen
- Nach dieser Vorlesung sollten Sie ein Gefühl dafür haben, warum es sich auch weiter lohnt, wenn Sie sich in der Praxis mit Patterns beschäftigen
- Dafür erzähle ich Ihnen etwas über Erfahrungen, die ich in zwei Firmen mit Patterns gemacht haben...

Fragen, die unter anderem heute beantwortet werden ..



- Erst mal ein kleines „running example“
- Machen Patterns aus Ihnen automatisch einen guten Designer?
 - Erfahrung
 - Die Sache mit dem Vokabular
- Warum ist es gut, Patterns auch mal zu schreiben?
 - Was lernen Sie dadurch für Ihre Reviewkultur
 - Was lernen Sie dadurch für sich als Software Designer
- Können Sie erwarten, dass in Ihrem Unternehmen später 75% der Mitarbeiter selbst Patterns schreiben?
- Beispiele dafür, wo man in der betrieblichen Praxis Muster findet
 - Wem helfen die
 - Was wird dadurch erleichtert?
- Zusammenfassung



Running Example - Produktserver

Running Example - Produktserver

Warum?



- Sie sollen diese Patterns nicht „lernen“
- Aber man kann daran ein paar Dinge erklären, die man aus Patterns lernen kann ...
- meine habe ich nicht genommen, weil ich sie für besonders gut halte, sondern weil ich mich darin auskenne und die Motivation dahinter folglich besser erklären kann ...
- Das Paper sollte für die Vorlesung verteilt sein und vorliegen
 - <http://www.objectarchitects.de/ObjectArchitects/papers/Published/ZippedPapers/inspat03.pdf>

Beispiel aus dem Versicherungsbereich

Architekturmuster Produktserver

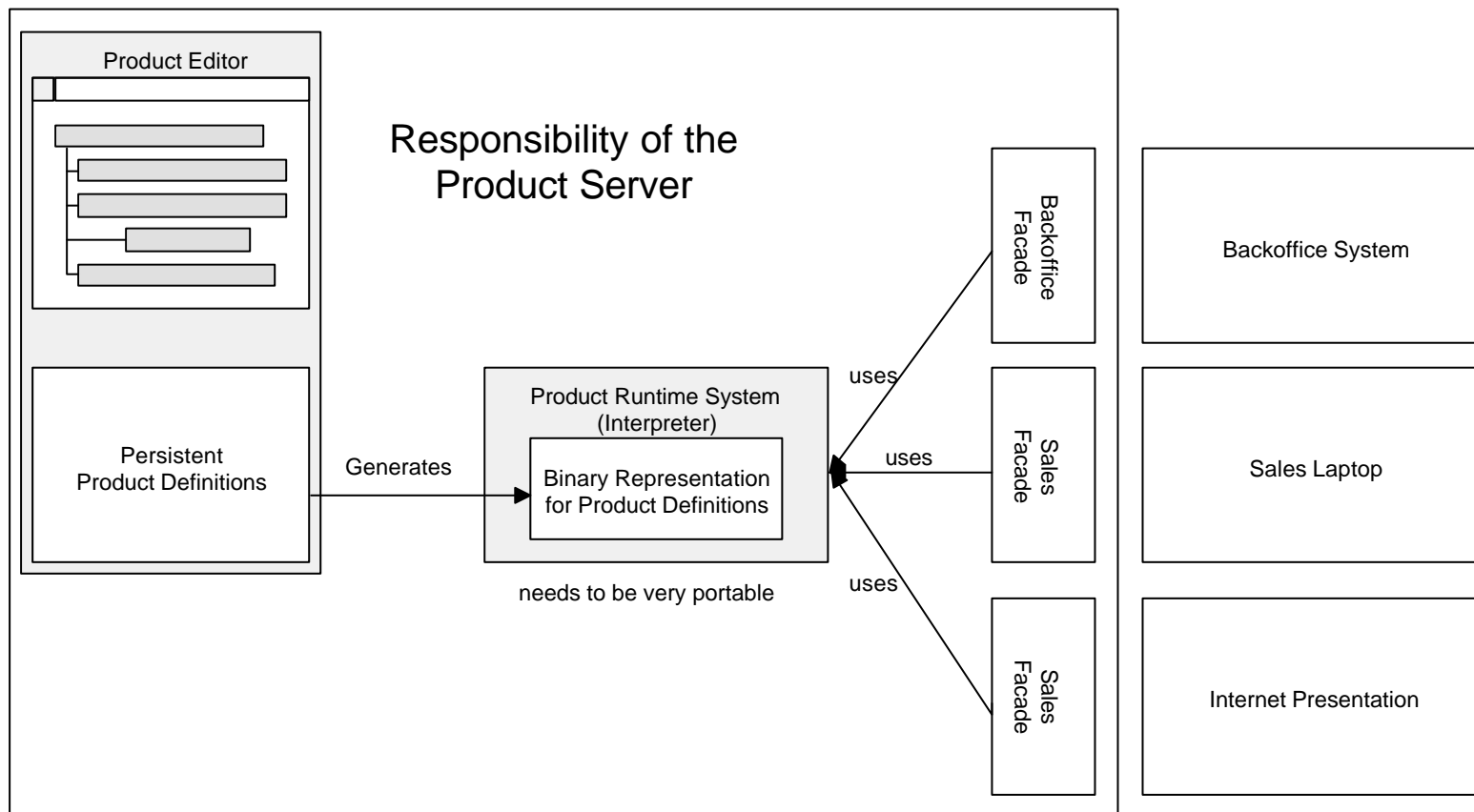


- Problem
 - Neue Versicherungsprodukte sollen innerhalb von Tagen im DV System verfügbar sein. Wie baut man ein System, das dies möglich macht?
- Faktoren, die das Design beeinflussen (exemplarisch ...)
 - Plattformunabhängigkeit **gegen** Kosten: Produktwissen muß Plattformunabhängig zur Verfügung stehen und in vielen Teilsystemen.
 - Kapselung des Produktwissens **gegen** optimales Design von User Interfaces: Gut designte Benutzungsschnittstellen enthalten ebenfalls Produktwissen - eine vollständige Auslagerung in einen Produktserver
 - ... Viele viele weitere ...



Architekturmuster Produktserver

- Lösung

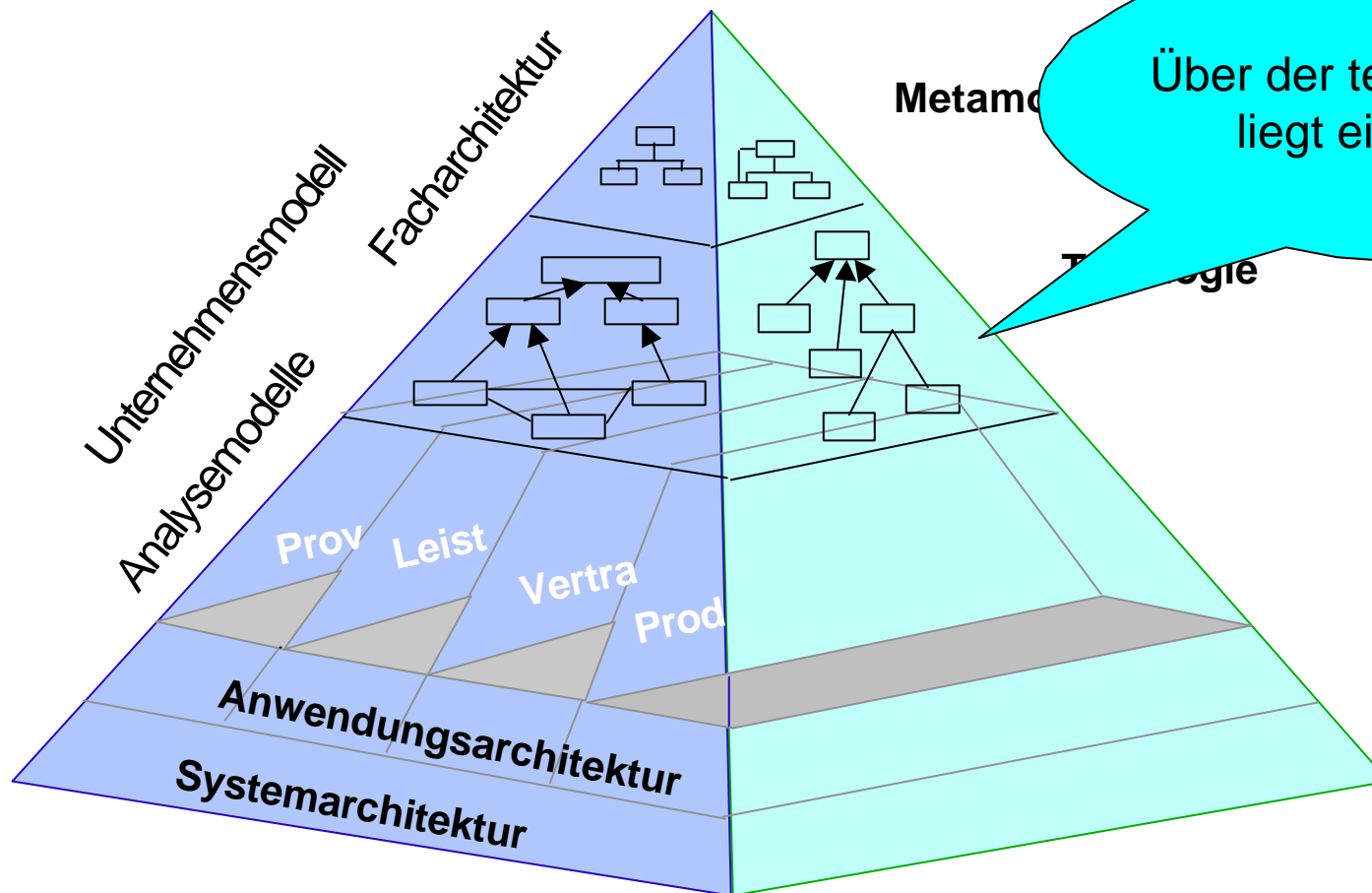




Architekturmuster Produktserver

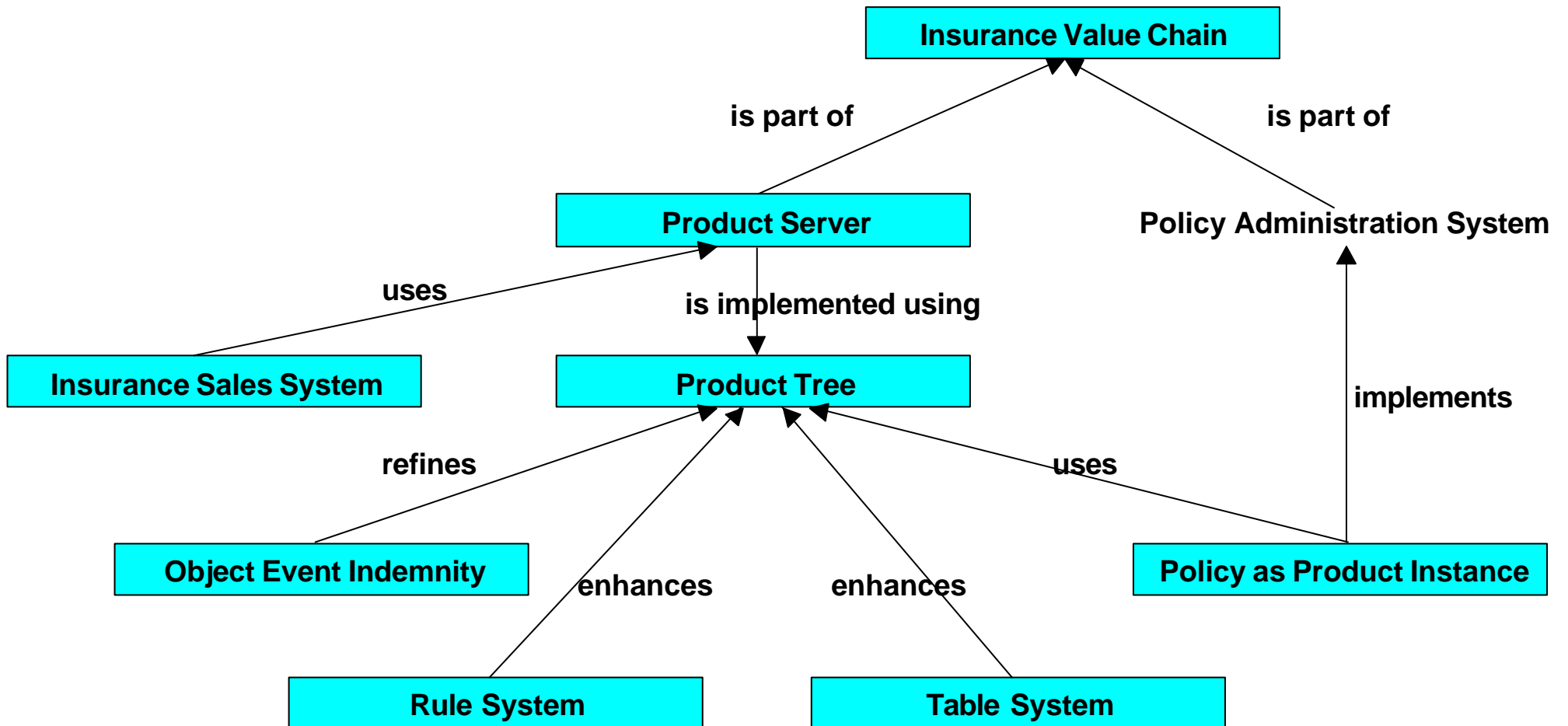
- Auswirkungen (exemplarisch ...)
 - Produktwissen wird plattformunabhängig zur Verfügung gestellt
 - Benutzungsschnittstelle selten voll aus dem Produktwissen zu generieren
 - also hier noch Programmierung bei der Einführung neuer Produkte
 - ... Viele weitere ...
- Zum Nachlesen im Web unter ..
 - Some Patterns for Insurance Systems, Wolfgang Keller, PLoP 1998, http://jerry.cs.uiuc.edu/~plop/plop98/final_submissions/

Wo und wie ... Analyse- und Datenmodell-Patterns



Über der technischen Architektur
liegt eine Facharchitektur

Beispiel: Landkarte bisher selbst beschriebener fachlicher Muster ..





Machen Patterns aus Ihnen automatisch einen guten Designer?

Machen Patterns aus Ihnen automatisch einen guten Designer? Erfahrung!



- **NEIN!**
- Wenn Sie Patterns lesen, können Sie einige Fehler anderer Leute vermeiden
 - auf der anderen Seite liest sich das GOF Buch aber nicht sehr flüssig
 - ich benutze es daher eher als Nachschlagewerk - man sollte die „Thumbnails“ kennen - den Rest kann man nachlesen.
- Richtig gut werden Sie erst, wenn Sie praktisch eine Menge Software geschrieben haben und die Muster darin verwendet haben.
- Patterns machen aus Ihnen also schneller einen guten Designer. Die eigene Erfahrung können Sie allerdings nicht ersetzen

Machen Patterns aus Ihnen automatisch einen guten Designer? Das Vokabular!



Mediator

Factory

Singleton

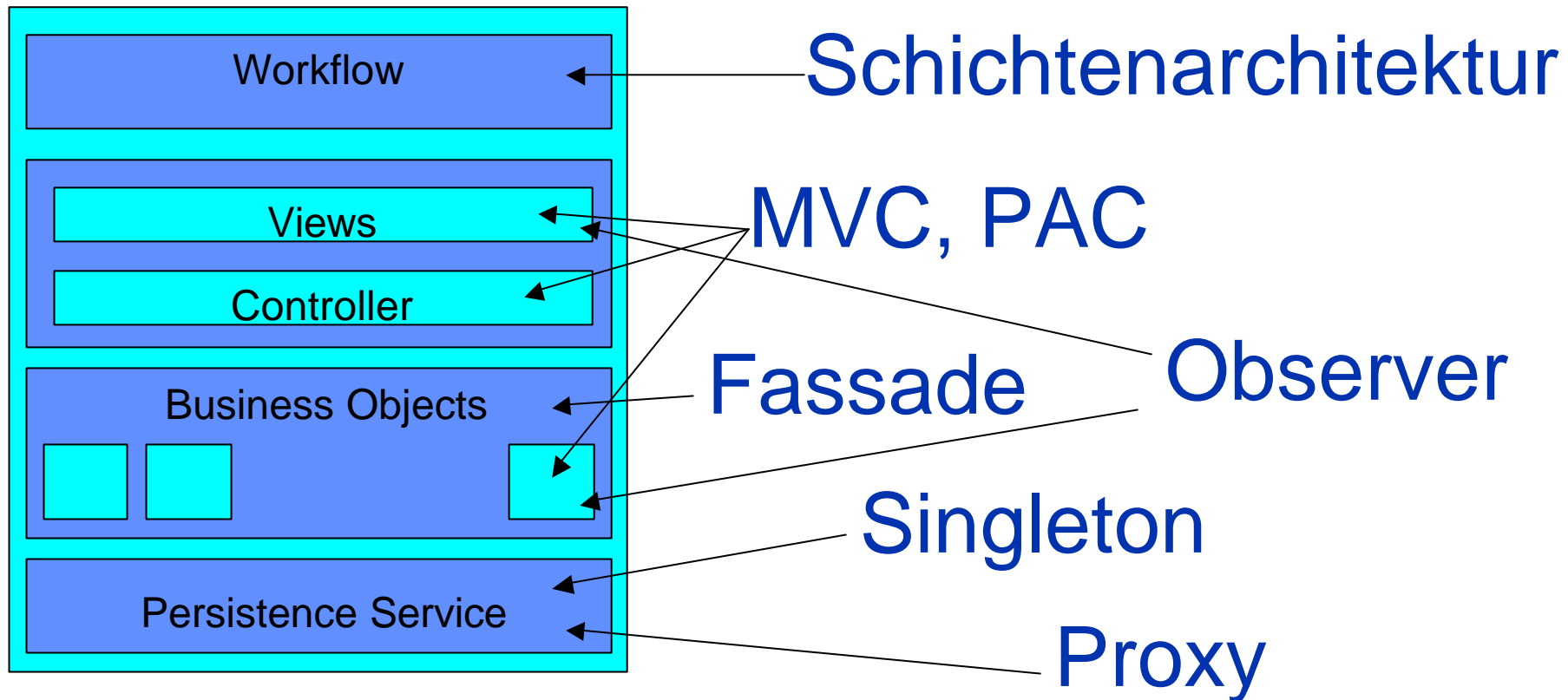
Observer

Composite

Wo und wie ... Design- und Architektur-Patterns



Entwurf und Erklärung von Architekturen



Machen Patterns aus Ihnen automatisch einen guten Designer? Das Vokabular!



- Patterns schaffen ein Vokabular
- Sie können sich damit schneller verständigen
- Damit ist der Name eines Patterns extrem wichtig!
 - Der Name sollte so sein, dass man sich das Design dahinten schnell einprägen kann und das Thumbnail sollte so sein, dass einem aus Name und Thumbnail intuitiv klar ist, was doch gleich die Lösung und das Design dafür war



Warum ist es gut, Patterns auch mal selbst zu schreiben?

Warum ist es gut, Patterns auch mal zu schreiben? Was lernen Sie dadurch für Ihre Reviewkultur?



- Sie werden in diesem Kurs auch noch Patterns schreiben
 - hoffentlich machen Sie es auch - es lohnt sich - wegen der Erfahrung
- Dazu werden Ihre Muster zwei Phasen durchlaufen
 - Shepherding
 - und weiter einen Writers Workshop
- Wenn beides gut gemacht ist, werden sich Ihre Partner nur darauf konzentrieren was Sie machen können, um Ihr Pattern **NOCH BESSER** zu machen



Grundhaltung der Pattern-Kultur

- der Autor ist ein Experte auf seinem Gebiet
 - ich gehe davon aus, dass der Autor weiss worüber er schreibt
 - ich respektiere ihn, seine Person und seine Arbeit
 - ich will ihm helfen, sein WERK noch besser zu machen
-
- Diese Kultur ist sehr stark geprägt durch ..
 - amerikanische Kultur - Think Positive
 - Challenge und nicht Problem
 - positives Formulieren
 - Wertschätzenden Umgang

Grundhaltung „europäischer“ Review-Kultur



- der andere ist eh ein Idiot
- ich werde ihm seine Fehler nachweisen
- ich werde solange seine Fehler suchen, bis ich nachgewiesen habe, dass er unfähig ist ...

- Diese Kultur ist sehr stark geprägt durch
 - das Wort Problem
 - Schwächenorientierung statt Stärkenorientierung
 - Destruktivität
 - ...



Beispiel für positives Formulieren

- (1) Negativ formuliert:
 - In der Einleitung fehlt eine Landkarte aller Patterns, die in diesem Artikel behandelt werden
- (2) Positiv formuliert
 - füge bitte noch eine Landkarte der Patterns ein
- (3) Noch positiver
 - Der Leser wird noch mehr von Deinem Artikel haben, wenn Du eine Landkarte der Patterns einfügst, auf der Du auch die Beziehungen der Patterns untereinander beschreibst, so wie das zum Beispiel GOF machen in

Welche Art der Formulierung wird dem Gereviewten am meisten helfen



- Welchen Kommentar würden Sie am liebsten umsetzen
 - (1), (2) oder (3)
- Durch welchen Kommentar fühlen Sie sich unter Umständen persönlich attackiert?
- Was machen Sie, wenn Sie sich und Ihre Arbeit persönlich angegriffen fühlen
- Wie fühlen Sie sich, wenn Ihnen jemand aber sagt, wie Sie noch erfolgreicher werden können



Was lernen Sie hier also durch Patterns?

- Durch die Pattern-Kultur lernen Sie wichtige Soft- und Social-Skills...
- Sie machen damit sich selbst produktiver
- Sie machen ihr Arbeitsumfeld produktiver
- Sie machen Ihre Kollegen produktiver
- Sie sind hilfreich und nicht destruktiv
- Also selbst, wenn Sie nicht Patterns schreiben, können Ihnen die Techniken der Pattern-Community extrem helfen



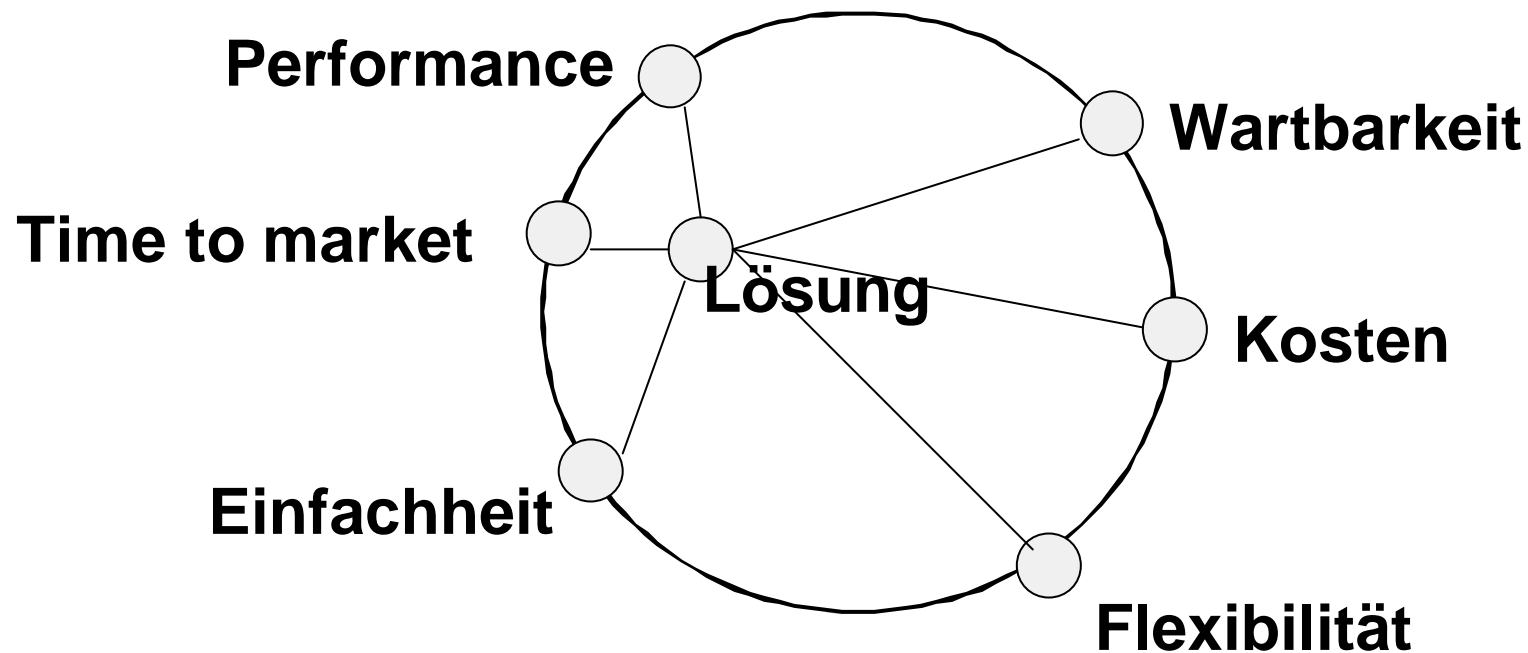
Was lernen Sie dadurch für sich
als Software Designer?



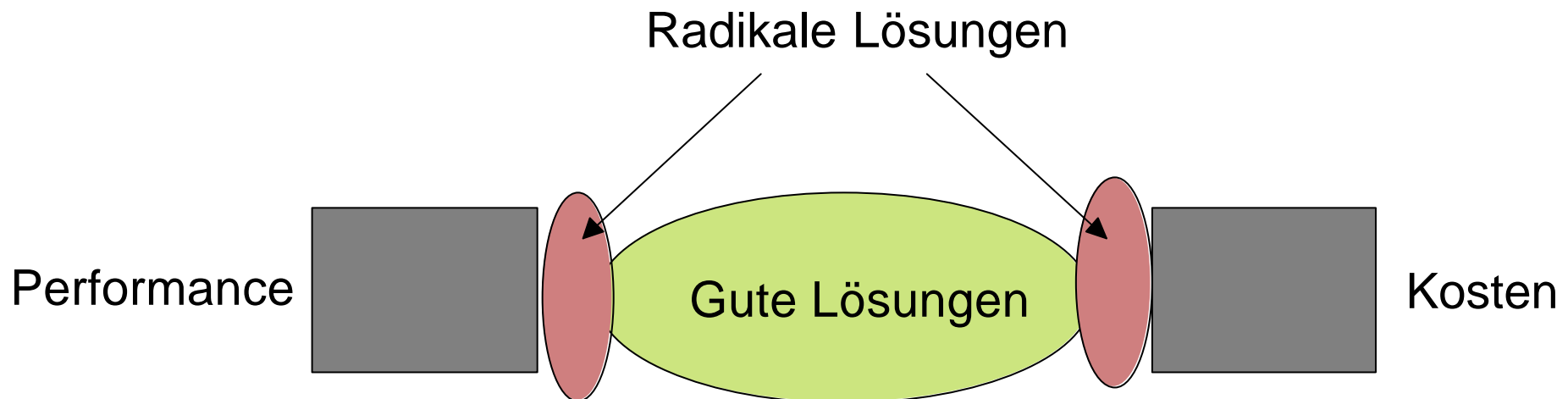
Wenn Sie Software bauen ...

- müssen Sie Funktionalität liefern
- Ihre Software muß aber auch nichtfunktionale Eigenschaften haben ...
 - Performance
 - Wartbarkeit
 - Benutzbarkeit
 - Kosten
 -

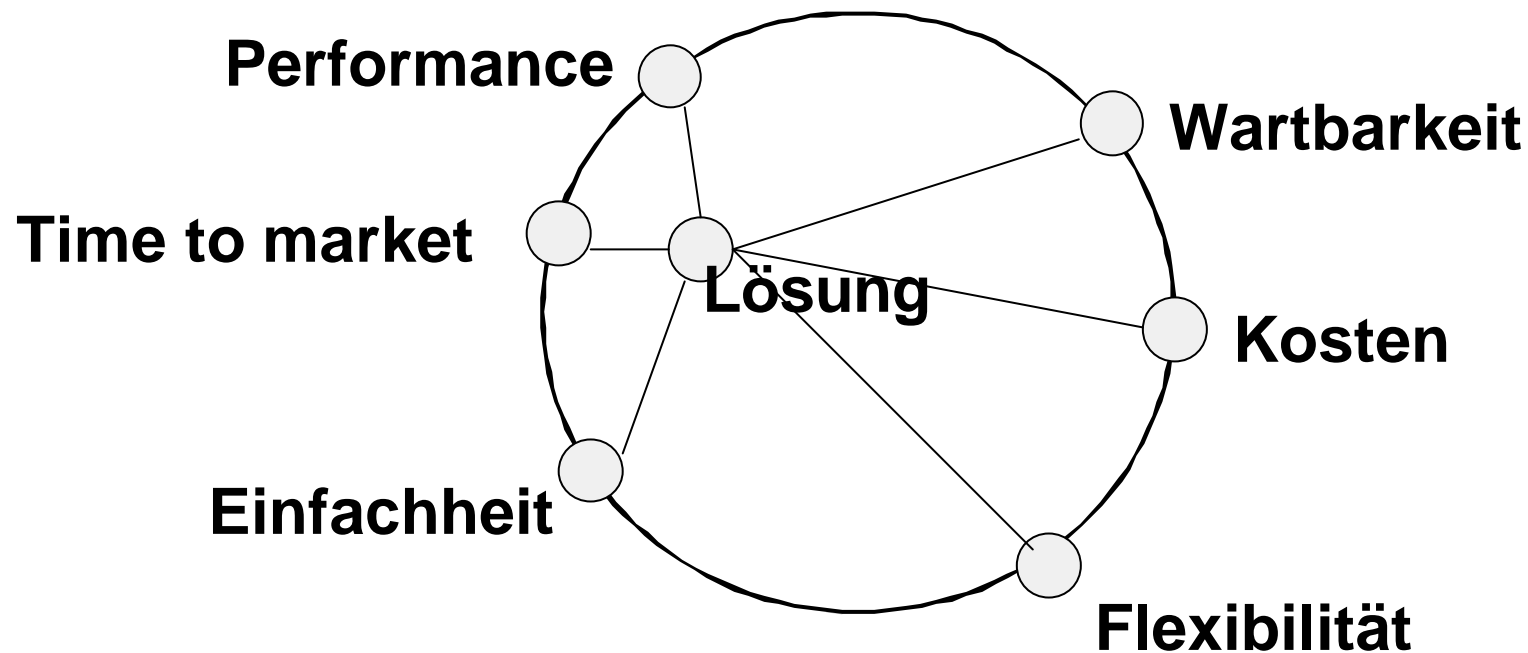
Nichtfunktionale Eigenschaften stehen typisch in Zielkonflikten zueinander ...



Gute Lösungen liegen selten genau auf einem Extrem eines Zielkonfliktes ...



Gut geschriebene Patterns machen neben der Nennung einer Lösung für ein Problem die Zielkonflikte explizit und zeigen die Konsequenzen, die eine Lösung hat auf ...



Und jetzt suchen Sie das bitte mal in den GOF Patterns



142 *STRUCTURAL PATTERNS*

CHAPTER 4

Consequences

Class and object adapters have different trade-offs. A class adapter

- adapts Adaptee to Target by committing to a concrete Adapter class. As a consequence, a class adapter won't work when we want to adapt a class *and* all its subclasses.
- lets Adapter override some of Adaptee's behavior, since Adapter is a subclass of Adaptee.
- introduces only one object, and no additional pointer indirection is needed to get to the adaptee.

An object adapter

- lets a single Adapter work with many Adaptees—that is, the Adaptee itself and all of its subclasses (if any). The Adapter can also add functionality to all Adaptees at once.
- makes it harder to override Adaptee behavior. It will require subclassing Adaptee and making Adapter refer to the subclass rather than the Adaptee itself.



Herausforderung ;-) gesehen?

- man könnte die GOF Patterns noch besser machen, wenn man am Anfang die Forces und Tradeoffs beschreibt, die die Lösung treiben
- Am Anfang sollte man also die Forces explizit nennen
- und hinten in den Consequences sollte man sagen, welche Auswirkungen eine bestimmte Lösung in Bezug auf die Forces hat
- Das kann man sehr einfach und sehr schematisch machen

Aber Vorsicht: Häufiger Kommentar „Your Forces read like requirements“



- Anforderungen sind Dinge, bei denen es normalerweise keinen Tradeoff gibt
 - Die Software muß den Arbeitsstand speichern können!
- Forces sind Dinge, bei denen es **Tradeoffs** gibt
- Man formuliert sie typischerweise in der Art „Speicherverbrauch“ versus „Performance“
 - mit einem Cache verbraucht man mehr Speicher, aber bekommt bessere Performance

Was lernen Sie also für sich als Software-Designer



- Sie gehen mit nichtfunktionalen Anforderungen bewusst um
 - und denken nicht erst an die Performance, wenn das System in Betrieb gehen soll ..
- Sie können als guter Architekt dem Bauherren erklären warum es teurer wird, wenn es schneller sein soll
- Sie sind ein besserer Reviewer
 - weil Sie Dokumente als verbesserungsfähig empfinden, die solche Zielkonflikte nicht explizit darstellen, sondern die Lösung als „gottgegeben“ hinstellen
 - und weil Sie das so kommunizieren, dass es angenommen wird (positiv!)



kleiner Exkurs für Ihr Pattern ..

Three Independent Known Uses

a.k.a. Buschmann's Law



Was ist ein Pattern und was nicht ...

- Patterns sind bewährte Lösungen, die beschrieben werden, weil sie in der Praxis unabhängig voneinander an vielen Stellen zu finden sind
 - **Ein eigenes Design, das man einmal gemacht hat und das man in der Form von Patterns beschreibt ist noch lange kein Pattern (es fehlen die sog. Known Uses)**
 - Eine Lösung zu einem häufig auftretenden Problem ist noch lange kein gut geschriebenes Patterns, wenn die Zielkonflikte und die Konsequenzen der Lösung auf die Zielkonflikte nicht erläutert sind (es fehlen die sog. Forces und Consequences)



Known Uses - Warum

- Patterns sind Muster, die in vielen Designs auftreten
 - Sie sind keine Beschreibungen des eigenen Designs - auch wenn man seine eigene Software mit Hilfe derselben Tricks gut und schnell beschreiben kann
- Daher sollte man „eine Sache“ in der Software von drei verschiedenen Leuten / Teams gefunden haben, die unabhängig voneinander gearbeitet haben
 - sonst läuft man Gefahr etwas zu beschreiben, was zwar gutes Design ist, aber streng genommen kein Muster



Können Sie erwarten, dass in Ihrem Unternehmen später 75% der Mitarbeiter selbst Patterns schreiben?



Leider nein ...

- Nur wenige Softwareentwickler schreiben auch gerne
- Von denen die schreiben, schreiben nur die wenigsten Patterns
- Die Zahl der Pattern-Schreiber vermehrt sich nur langsam

- Guter Zugang ist allerdings der über die Review-Kultur
 - siehe vorne - Si können den Leuten demonstrieren, wie positiv und produktiv Shepherding oder Writer's Workshops funktionieren auch an anderen technischen Papieren

Beispiel Produktserver wieder: Eigene fachliche Patterns Potentiale noch bei ...



- Produktmodellierung
 - Bausteinbildung
 - Struktureller Aufbau von Produktmodellen
- Referenz-Objektmodelle für versicherungstechnische Systemteile
 - Aufbau von Vertragsservern
 - Technische Basisfunktionen für ...

Diese Patterns hat seit 1999 keiner bei uns im Unternehmen geschrieben, obwohl jeder sie brauchen würde



Beispiele dafür, wo man in der betrieblichen Praxis Muster findet

Gibt es mehr als Design Patterns

Wo und wie ...

Analyse- und Datenmodell-Patterns



- Facharchitektur beschreibt fachliche Referenz-Objekt-Modelle
- Auch hier gibt es inzwischen viele Muster, die Sie sicher auch verwenden ...
 - Partner-Rollen (Partnersysteme)
 - Gespiegelte Stückliste (produktflexible Vertrags...
 - Historien (Vertragssysteme)
 - Buchungen (in vielen Feldern)

Kennt man zum Teil aus Martin Fowler's
Analysis Patterns

Wo und wie ...

Eigene fachliche Muster



- Verwenden wir, um den versicherungsfachlichen Teil der Facharchitektur zu beschreiben
- Zum Beispiel
 - Wertkette der Versicherung
 - Produktserver
 - Außendienst-Shell
 - Produktbaum
 - Objekt / Ereignis / Leistung
 - Gespiegelte Stückliste ...
 - ... Weitere

FAQ: Wo helfen uns Patterns also in der betrieblichen Praxis? (fast Wiederholung)



- Patterns schaffen auch hier eine **Sprache für Design**
- Das Problem und die Kräfte werden explizit gemacht - nicht nur die Lösung.
- Man bekommt fertige Skizzen für gutes Design
- Die kann man schneller kommunizieren
- Man kann sein eigenes Design besser dokumentieren
- Man wird (wie schon gesagt) schneller ein guter Designer



Zum Schluß noch ...
Warum sitzen Sie in diesem Kurs?

FAQ: Soll ich alle meine DV-Fachkräfte auf Pattern-Schulungen schicken?



- Man sollte besser von DV-Fachkräften erwarten, daß sie sich in Ihrem Beruf selbst auf dem laufenden Stand halten - und Patterns gehören dazu.
- Initialzündungen und kurze Veranstaltung sind ok, wochenlange Schulungen sollte man vermeiden.
- Besser auf die Eigeninitiative setzen und Patterns bei Bedarf zugänglich machen. Patterns bei Bedarf verwenden und immer wieder verwenden.
- Bei Externen voraussetzen.

Frage aus einem Vortrag für
EDV-Manager von Versicherungen
1999



Für Sie übersetzt heißt das

- lernen Sie was Forces sind
 - und lernen Sie Zielkonflikte im Softwaredesign explizit zu machen, zu dokumentieren und verlangen Sie auch solche Doku
- nehmen Sie die positive Kultur der Pattern-Community mit
 - und werden Sie ein besserer Reviewer
 - sein Sie hilfreich, indem Sie positiv formulieren
 - und vermeiden Sie dadurch unnötige Konflikte in Ihrem Arbeitsumfeld, in dem Sie hilfreich sind, statt negativ
- lernen Sie das, indem Sie den Prozeß einmal „mitmachen“
 - kann sein, dass es Ihr Hobby wird ...