

# Patterns in der Produktkonfiguration

Matthias Plietz

ORISA Software GmbH Jena  
Friedrich-Schiller-Universität Jena

## Zusammenfassung

Ausgehend von den Erfordernissen zur Einführung eines Produktkonfigurators wird ein ganzheitlicher Einführungsprozess skizziert und durch einen Pattern-Ansatz unterstützt.

## 1 Einleitung

Ein Produkt- und Angebotskonfigurator wird zum einem entscheidenden Wettbewerbsvorteil moderner Unternehmen des Maschinen- und Fahrzeugbaus sowie anderer Branchen. Einerseits zwingt der Wettbewerbsdruck zu individualisierten Produkten - andererseits erfordert der Kostendruck machbare und wiederverwendbare Lösungen. Die wachsende Komplexität dieses Widerspruchs wird durch einen Konfigurator beherrschbar, indem er die Machbarkeit auf Basis von Standards prüft, den Angebots- und Auftragszyklus verkürzt sowie die korrekte Preisfindung und Angebotserstellung ermöglicht.

Um den Einsatz und den Betrieb eines Konfigurators zum Erfolg zu führen, muss der Konfigurator optimal konzipiert und gestaltet werden. Durch die Ausarbeitung und Bereitstellung von konkreten Hilfestellungen, Einführungsmethodiken, Szenarien und Fallbeispielen sollen Projekte erfolgreich abgeschlossen werden.

## 2 Einführungsmethodiken für Konfiguratoren

Marktstudien und Auswahlkriterien von Standardsoftware für Produktkonfiguratoren bieten wichtige Informationen bei der Softwareauswahl und dienen der Markttransparenz. Sie sind einzubeziehen, um Software nach den konkreten Bedürfnissen auszuwählen. Im praktischen Vorgehen müssen diese Bedürfnisse in vorangestellten Entscheidungsprozessen erarbeitet, inhaltlich konkretisiert und weitgehend unabhängig von der konkreten Software betrachtet werden.

## 2.1 Aspekte der Einführung

Inhaltliche Herangehensweisen und Aspekte der Konfigurator-Einführung beschäftigen sich mit Fragestellungen der technischen und bediengerechten Gestaltung, der Integration, der Kosten-Nutzen-Betrachtung u.a.m. Meist sind solche Aspekte separat betrachtet worden. Beispielsweise führt Henseler eine Methode für die Ablage spezialisierten Konfigurationswissens ein (Henseler, 2004); Ghoffrani beschreibt die Integration in CAD-Systeme (Ghoffrani, 2007).

Zunehmend sind Arbeiten zu finden, die die gemeinsame und integrative Betrachtung verschiedener Aspekte zum Ziel haben. Herrmann beschreibt Vorgehensweisen zur Gestaltung von Konfiguratoren unter betriebswirtschaftlicher Sicht (Herrmann & Heitmann, 2006). Wüpping betrachtet das Variantenmanagement und die Variantenmodellierung unter Effizienz- und Nutzen-Gesichtspunkten (Wüpping, 2010). Drews erkennt, dass die Designer von Konfiguratoren Kenntnisse über Gestaltung, Produkt und Unternehmenszielen haben müssen, um einen Konfigurator erfolgreich im Markt zu platzieren (Drews, 2008). Er benennt diese Aspekte in Anlehnung an andere Autoren als technische Anforderungen, Kundenbedürfnisse und Geschäftsziele.

Für Drews ist der Einsatz des Konfigurators im Online-Vertriebskanal vorgegeben; auch werden die Einflüsse, die sich durch Einsatz einer standardisierten Software ergeben, nicht betrachtet. Daher ist ein weiterer, wesentlicher Aspekt der Konfigurator-Einführung zu betrachten: die Integration des Konfigurators in die bestehende Softwarearchitektur des Unternehmens muss innerhalb der Einführung analysiert und entschieden werden, um den Erfolg des Konfigurators zu ermöglichen. Aus dieser Sicht ist ein Konfigurator nie Selbstzweck, sondern ordnet sich als Instrument den Einsatzzielen als Erweiterung der Geschäftsziele unter. In diesem Kontext umfasst die Integration eher die Ausprägungen der Software und deren Bedien- und Prozessflüsse im Zusammenspiel anderer Softwarekomponenten.

Zusammenfassend lassen sich wesentliche Aspekte der Konfigurator-Einführung erkennen.

<b>Aspekt</b>	<b>Beschreibung und Konkretisierung</b>
Technische Anforderungen	umfassen die Eigenschaften, Struktur und Potenzial des zu verkaufenden Produktes, das Produktmodell für einen Konfigurator sowie das Variantenmanagement mit Beherrschung der Varianz
Kundenbedürfnisse	der Kunde wird als Nutzer des Produktes und als Anwender der Software angesehen; umfassen auch die Software-Erstellung und Anforderungen an ihre Gestaltung
Geschäftsziele	umfassen die vertrieblichen Ziele, die Kosten und Nutzenpotenziale, die Einsatzziele, die Anwendergruppen und (Vertriebs-)Kanäle sowie Return-on-Invest-Schätzungen

Integration	umfasst Prozessabläufe und Datenflüsse, Einsatzszenarien sowie die Interaktion mit vorhandener Software
-------------	---

*Tabelle 1: Aspekte der Konfigurator-Einführung (vgl. Drews, 2008)*

## 2.2 Gesamtprozess der Einführung

Die beschriebenen Aspekte sind immer im Zusammenhang zu betrachten und in ihrer Wirkung zum Teil gegenläufig. Drews folgert die notwendige gemeinsame Betrachtung der Aspekte und ihrer Wechselwirkungen im Zusammenhang, um „dabei Kundenbedürfnisse, Ziele des Anbieters [hier: das Kundenunternehmen] und technische Möglichkeiten in Einklang zu bringen“ (Drews, 2008, 368).

Die ganzheitliche Betrachtung dieser Aspekte ist entscheidend für die Akzeptanz einer Konfiguratorsoftware (vgl. ORISA, 2011). Diese Studie beschreibt, dass sich bei Verwendung von Standardsoftware der Implementierungsaufwand auf Spezialfunktionen und Anpassungen verlagert und sich im Verhältnis zum Gesamtprozess der Konfigurator-einführung verkürzt. Der wesentliche Aufwandsanteil aktueller Einführungsprojekte wird auf Seiten des Kundenunternehmens generiert, um mit Beratungsunterstützung Entscheidungen zu treiben und verantwortlich zu bearbeiten wie:

- die Festlegung der Benutzergruppen und Integration der Anwendung
- die Analyse, Strukturierung und Pflege (Eingabe) der Produktdaten in den Konfigurator
- die Erstellung der Bedienoberfläche und der Bedienabläufe

Praxis ist auch - vor allem bei kleineren und mittleren Unternehmen (KMU) -, dass die benannten Aufgaben mit geringem Einführungsetat und mit einem Minimum an externer Beratung durchgeführt werden.

Andererseits wird auf Grund dieser Aufgaben-Verlagerung das Kundenunternehmen von den Anwendern als Software-Lieferant bewertet. Ein Anwender kann nicht unterscheiden, ob eine spezielle Funktion durch eine Eigenschaft des zu konfigurierenden Produktes, durch eine Besonderheit in der Abbildung innerhalb des Konfigurators oder durch eine Funktionalität der Standardsoftware selbst hervorgerufen wurde. Durch diese nicht differenzierbare Situation hat die konkrete Implementierung einen wesentlichen Einfluss auf die Akzeptanz des konfigurierbaren Produktes und auf die erreichbaren Geschäftsziele des Kundenunternehmens (vgl. Drews, 2008, Pogoll & Piller, 2002).

Es sind Einführungsmethodiken und Handlungsanweisungen, basierend auf Erfahrungen, Beispielen und best practice, notwendig, um praxisrelevante Unterstützung in adäquatem Abstraktionsgrad geben zu können.

### 3 Pattern als Einführungsmethodik

Eine Methodik zur Sammlung, Ablage und Reproduzierbarkeit von Produktwissen, Anforderungen und Ausführungen liefert das Pattern-Konzept.

#### 3.1 Historie und Einordnung

Seit Alexander wird der Begriff „Design Pattern“ in der Architektur genutzt (Alexander, Ishikawa et al, 1977). Mit dem Pattern-Begriff definierte er eine Beziehung zwischen Problem eines Nutzers, der in einem Kontext agiert und nach einer Lösung für das aufgetretene Problem sucht. "Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution." (Alexander, 1979, 247).

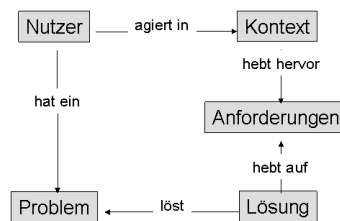


Abbildung 1: Pattern-Konzept (vgl. Kohls 2007)

Dieser Ansatz schafft die Voraussetzungen

- zur Definition einer einheitlichen Ausdrucksweise für Probleme und deren Lösungen innerhalb eines abgrenzten Kontextes und
- zur Sammlung von Lösungen für typische, immer wiederkehrende Probleme.

Durch Arbeiten von Gamma, Helm, Rising und anderen wurden Patterns in die objektorientierte Softwareentwicklung eingeführt (Gamma, Helm et al, 2004, Rising, 1998). Tidwell führt Human Interface Patterns ein und wendet sie auf die Interaktion zwischen Mensch und Anwendung an (Tidwell, 2005).

Patterns besitzen auch im Bereich des Wissensmanagements das Potenzial der Vereinfachung und Strukturierung. Kohls strukturiert Erfahrungs- und Expertenwissen und stimmt Vorgehensweisen durch Patterns ab (Kohls, 2007). Kerth betrachtet Patterns zur Anforderungsdefinition und zur Umsetzung von Anforderungen in die Realisierung von Softwaresystemen und liefert damit einen Ansatz zum Softwaredesign auf Basis von Patterns (Kerth, 1995).

## 3.2 Kriterien und Formalisierung

Damit Patterns bewährte praktische Lösungen beschreiben können, sind sie durch Analyse oder Experiment herzuleiten und zu abstrahieren. Einige wesentliche Kriterien, die an Patterns gestellt werden, sind nachfolgend aufgeführt.

Nach Alexander sollen Patterns Probleme lösen, erprobte Konzepte bieten, über das rein Offensichtliche hinausgehen und Beziehungen auf tiefere Strukturen aufzeigen. Alexander spricht in diesem Zusammenhang von „enfolding“ (vgl. Alexander, Ishikawa et al, 1977). Gamma und Mitarbeiter betonen ein gemeinsames Verständnis der beteiligten Personen durch Benennung und Beschreibung sowie die potentielle Reduktion der Komplexität durch Verlagerung in Patterns (Gamma, Helm et al, 2004).

Von den 10 Gütekriterien, die Kohls benennt, sind beispielhaft aufgeführt (vgl. Kohls, 2007):

- Validität: löst es wirklich das Problem?
- Plausibilität: ist es überzeugend formuliert und akzeptiert?
- Relevanz: geht es über Bekanntes hinaus?

Fowler erweitert den Pattern-Ansatz und stellt den Mustern, die bis dahin implizit als Entwurfsmuster (design patterns) angesehen wurden, Analysemuster (analysis patterns) voraus (Fowler, 1997). Analysis Patterns unterstützen bei Anforderungs- und Analyseaufgaben, die formal dokumentiert werden und als Basis für Entwicklungsschritte dienen sollen. Hahsler beschreibt das Zusammenwirken von Analyse und Design Patterns im Gesamtentwicklungsprozess, geteilt in Analyse- und Design-Phase, an konkreten Beispielen (Hahsler, 2001).

## 3.3 Patterns für die Produktkonfiguration

Durch diese Erfahrungen motiviert, soll erarbeitet werden, wie auch im Kontext eines Produktkonfigurator Einführungsmethodiken, best practice und Handlungsanweisungen durch Patterns beschrieben und unterstützt werden können (vgl. Plietz, 2010).

Bezogen auf dieses Exkursgebiet dienen Patterns folgenden Zielen:

1. die strukturierte Aufarbeitung und Darstellung notwendiger Schritte und Aspekte zur Einführung eines Konfigurator, Unterstützung des gemeinsamen Verständnisses aller beteiligten Personen durch Vorgehensweisen und Terminologien
2. die vereinfachte Darstellung und Pflege des Produktwissens durch eine inhaltsgetriebene Struktur; die Reduktion des Datenumfanges durch Verschiebung der Komplexität in Patterns und deren Wiederverwendbarkeit

Der Einsatz von Patterns innerhalb der Produktkonfiguration wird noch von wenigen Autoren beschrieben. Lee und Mitarbeiter beschäftigen sich mit Pattern zur Beschreibung und Normalisierung von Produktmodellen für den Systemaustausch (Lee, Eastman et al,

2007). Drews untersucht die Nutzung von interaction patterns bei der Gestaltung des Dialog- und Interaktionsverhaltens von Konfiguratoren (Drews, 2008). Andererseits lassen sich Hilfestellungen und Methoden, die bereits in der Literatur beschrieben sind, als Patterns auffassen und entsprechend formalisieren (vgl. Kohls, 2007).

### 3.3.1 Patterns für die Aspekte der Einführung

Solche Patterns dienen der Beschreibung eines Teilbereiches (Aspektes) innerhalb des Einführungsprozesses. In der folgenden Tabelle werden, zugeordnet zu den oben beschriebenen Aspekten, exemplarische Patterns beschrieben.

Aspekt	Pattern und Problemstellung (exemplarisch)
Technische Anforderungen	<p>nstufigeKonfiguration: auf welcher Ebene sind Komponenten noch separat verkaufbar oder anpassbar?</p> <p>Paketierung: sind Pakete bzw. Vorzugsvarianten definiert?</p> <p>DomainVerträglichkeit: welche Abhängigkeiten, Ausschluss und Herleitung von Optionen und Komponenten sind zu betrachten?</p>
Kundenbedürfnisse	<p>Direktionalität: soll die Konfiguration sequentiell oder wahlfrei durchlaufen werden?</p> <p>Konfliktbehandlung: sollen konfliktierende Optionen angezeigt werden und wählbar sein?</p>
Geschäftsziele	<p>Belegeffektivierung: welche Belegart (Schätzung, Angebot, Auftrag) soll vorrangig verbessert werden?</p> <p>Varianten: sollen Pakete bzw. Vorzugsvarianten bevorzugt verkauft werden?</p>
Integration	<p>Vertriebskanal: welche Vertriebskanäle soll der Konfigurator erschließen?</p> <p>VertriebFertigungsSicht: soll der Konfigurator ERP-konforme Daten generieren?</p>

*Tabelle 2: Pattern und Beispiele zu Einführungsaspekten*

### 3.3.2 Patterns für den Gesamtprozess der Einführung

Der Anspruch, den Prozess der Einführung ganzheitlich zu betrachten, verlangt die Erweiterung und Integration der beschriebenen Aspekte in ihren Zusammenhängen in einen Gesamtprozess.

Einerseits werden solche Zusammenhänge durch Referenzen zwischen den Patterns selbst dargestellt. Bereits Gamma und Mitarbeiter konnten solche Verknüpfungen als

Netzstrukturen verwalten und deren Referenzierungsart an den Kanten darstellen (vgl. Gamma, Helm et al, 2004).

Fasst man diese Kanten als bedingte Vorgänger-/Nachfolger-Relationen auf, so lassen sich Bearbeitungsreihenfolgen definieren. Alexander hat im Ansatz des „enfolding“ (vgl. Alexander, 1977) eine Referenzierungsart zwischen Patterns erarbeitet, bei der die Kanten zwischen Patterns eine thematische Unterordnung und Konkretisierung (Spezialisierung) darstellen. Referenzen zwischen Patterns können auch Assoziationen aufeinander sein, um einander beeinflussende Strukturen oder Entscheidungen anzuzeigen. In diesem Fall sind die Patterns einander gleichberechtigt und beeinflussen den Einführungsprozess kooperativ.

Andererseits können zusätzliche strukturierende Patterns die Gliederung und die Steuerung des Einführungsprozesses unterstützen.

Referenz	Problemstellung am Beispiel (exemplarisch)
Vorgänger-Nachfolger	Sequentialisierung von Patterns Beispiel: Pattern „Geschäftsziele→Belegeffektivierung“ entscheidet über „Integration→VertriebFertigungsSicht“
Spezialisierung	Ausprägung oder Konkretisierung von Daten Beispiel: Pattern „stufigeKonfiguration“ wird durch „Domainverträglichkeit“ präzisiert
Assoziation	Zusammenhang zwischen konkurrierenden oder kooperierenden Patterns Beispiel: Zusammenhang von „Technische Anforderungen →Paketierung“ mit „Geschäftsziele→Varianten“

Tabelle 3: Pattern-Referenzen

Diese Beispiele zeigen den grundsätzlichen Lösungsansatz auf; für eine umfassende Beschreibung und Unterstützung des Einführungsprozesses sind weitere Forschungsarbeiten notwendig.

### Literaturverzeichnis

- Alexander, C. (1979). *The timeless way of building*. New York, NY: Oxford Univ. Press.
- Alexander, C., Ishikawa, S. & Silverstein, M. (1977). *A pattern language* (2). New York, NY: Oxford Univ. Press.
- Drews, M. (2008). Interaction Patterns für Produktkonfiguratoren. In M. Herczeg & M. C. Kindsmüller (Hrsg.), *Mensch & Computer 2008: Viel Mehr Interaktion* (S. 367-376). München: Oldenbourg Verlag.
- Fowler, M. (1997). *Analysis Patterns: Reusable Object Models*. Addison-Wesley.
- Gamma, E., Helm, R. e, Johnson, R. & Vlissides, J. (2004). *Entwurfsmuster - Elemente wiederverwendbarer objektorientierter Software*. München [u.a.]: Addison-Wesley.

- Ghaffari, M. (2007). *Entwicklung und Einführung eines flexiblen Softwaresystems zur Konfigurierung virtueller Produkte*. Ruhr-Universität Bochum.
- Hahsler, M. (2001). *Analyse Patterns im Softwareentwicklungsprozeß*. Wirtschaftsuniversität Wien.
- Henseler, P. (2004). *Die Konfigurations- und Verträglichkeitsmatrix als Beitrag für eine differenzierte Betrachtung von Konfigurierungsproblemen*. ETH Zürich.
- Herrmann, A. & Heitmann, M. (2006). Produktkonfiguration als Präferenzkonstruktion. *Die Unternehmung (Swiss Journal of Business Research and Practice)*, 03/2006, 223 – 238.
- Lee, G., Eastman, C. M. & Sacks, R. (2007). Twelve Design Patterns for Integrating and Normalizing Product Model Schemas. *Computer-Aided Civil and Infrastructure Engineering* (22), 163-181.
- Kerth, N. L. (1995). Caterpillar's Fate: A Pattern Language for Transformation from Analysis to Design. *Pattern Languages of Program Design* (S. 293-320). Addison-Wesley.
- Kohls, C. (2007). *Design Patterns zur Dokumentation von Erfahrungswissen*. Gefunden am 20.12.2010 unter [www.bildungstechnologie.net/Members/hg/bt-talks/Christian\\_Kohls\\_-\\_Design\\_Patterns\\_zur\\_Dokumentation\\_von\\_Erfahrungswissen.pdf](http://www.bildungstechnologie.net/Members/hg/bt-talks/Christian_Kohls_-_Design_Patterns_zur_Dokumentation_von_Erfahrungswissen.pdf)
- ORISA Software GmbH (2011). *Projektauswertung CREALIS (internes Papier)*.
- Plietz, M. (2010). *Structured Development Process of Configuration Models*. Beitrag zum ECAI 2010 Workshop on Configuration, Lisbon.
- Pogoll, T. & Piller, F. (2002). *Konfigurationssysteme für Mass Customization und Variantenproduktion*. Marktstudie: Eigenverlag.
- Rising, L. (1998). *The patterns handbook: techniques, strategies, and applications* (13). Cambridge: University Press.
- Tidwell, J. (2005). *Designing interfaces. Patterns for effective interaction design*. Beijing [u.a.]: O'Reilly.
- Wüpping, J. (2010). *Spannungsfeld Variantenvielfalt und Produktkostensenkung*, Beitrag zum VDMA-Informationstag, Frankfurt, 16. März 2010.

### **Kontaktinformationen**

Dipl.-Ing. Matthias Plietz  
ORISA Software GmbH Jena  
Humboldtstrasse 13  
D-07743 Jena  
eMail: [plietz@orisa.de](mailto:plietz@orisa.de)

Friedrich-Schiller-Universität Jena  
Lehrstuhl für Wirtschaftsinformatik  
Carl-Zeiß-Straße 3  
D-07743 Jena