HARMONISIERUNG VON SOFTWARE-ARCHITEKTUREN UND PLATTFORMEN (HAP)

ERFAHRUNGEN AUS DEM INDUSTRIELLEN KONTEXT BEI SIEMENS

Dr. Michael Gloger, Dr. Stefan Jockusch, Norbert Weber
Siemens AG
Zentralabteilung Technik
München

Siemens AG, ZT SE 3
Otto-Hahn-Ring 6
D-81739 Muenchen
Email: Michael.Gloger@mchp.siemens.de
Tel.: +49 89 636 42161
Fax.: +49 89 636 44424

Viele Produktdomänen zeigen ein ausgeprägtes „Heterogenitätsproblem“: Aufgrund unterschiedlicher Produktstrukturen können Synergien nicht genutzt werden. Die produktübergreifende Verwendung von gemeinsamen Komponenten verlangt, daß eine gemeinsame Referenzarchitektur definiert wird, die auch die Standardschnittstellen vorgibt. Zu diesem Zweck ist das Verfahren „Harmonization of Software Architectures and Platforms“ (HAP) in der Siemens AG entwickelt und mehrfach angewendet worden. Dieser Beitrag stellt die Grundprinzipien von HAP und seine methodischen Arbeitsschritte vor. Darüber hinaus beschreibt er die Erfahrungen, die bei der Anwendung von HAP im Rahmen mehrerer Harmonisierungsprojekte gemacht wurden.

Schlüsselwörter: Software Architekturen, Systemplattformen, Harmonisierung von Architekturen, Domain Engineering

1 Die Herausforderung

Als wichtiger Faktor für die Steigerung der Wettbewerbsfähigkeit unseres Unternehmens kristallisiert sich verstärkt die Fähigkeit heraus, Synergien in der Softwareentwicklung zu erkennen und zu nutzen. Die Produktdomänen vieler Geschäftsgebiete zeigen jedoch ein ausgeprägtes „Heterogenitätsproblem": aufgrund heterogener Systemstrukturen bleiben Synergiepotentiale ungenutzt. Obwohl Produkte im gleichen Anwendungsgebiet angesiedelt sind und viele Gemeinsamkeiten haben, setzen sie aufgrund ihrer Historie auf unterschiedlichsten Plattformen auf, enthalten funktional gleiche Komponenten in separaten Implementierungen oder stellen verschiedene Schnittstellen nach außen bereit. Diese mangelnde Einheitlichkeit führt zu erhöhten Entwicklungs-, Wartungs- und Servicekosten; das Erscheinungsbild der Produkte und ihre Akzeptanz beim Kunden leiden.

Oft ist das Problem zwar bereits erkannt und es werden Maßnahmen zur Harmonisierung der Produkte ausgearbeitet. In vielen Fällen jedoch werden sie nicht umgesetzt, weil für die vorgeschlagenen Lösungen keine Akzeptanz bei den Betroffenen erzielt werden kann. Diese Erfahrung zeigt, daß sich die Harmonisierung von Software-Architekturen und -Plattformen nicht auf die Entwicklung „korrekter“ technischer Lösungen allein beschränken darf. Vielmehr muß sie sich auf eine systematische Vorgehensweise stützen, die die Akzeptanz und Umsetzung der Maßnahmen sichert, indem sie von Anfang an alle betroffenen Abteilungen und Produktbereiche einbezieht, die Maßnahmen auf deren Anforderungen an die Produkte und ihre Entwicklung abstimmt und den Nutzen der Maßnahmen transparent macht.

Für die Harmonisierung heterogener Produktdomänen ist das Verfahren *„Harmonization of Software Architectures and Platforms“ (HAP)* entwickelt worden, das diese entscheidenden Aspekte zu einem strukturierten Vorgehen integriert. Inhaltlich umfaßt es die Analyse und Verbesserung des Status-Quo, geht aber darüber hinaus. Da Architekturen eine entscheidende Investition in die Zukunft der Produkte und der Produktdomäne sind, bezieht das Vorgehen systematisch die zukünftigen Markt- und Technologietrends mit ein. Die Harmonisierung bildet so

die Grundlage für eine umfassende Produktinnovation, deren Intention die Tragfähigkeit der Architektur bzw. Plattform für die Zukunft ist [KMK 1993].

Die Zielsetzung, die das Verfahren HAP verfolgt, ist zunächst die Definition einer Referenz- bzw. Plattformarchitektur für die verschiedenen Produkte eines Anwendungsgebietes. Dazu gehört auch die Festlegung von gemeinsamen Komponenten und den zugehörigen Standardschnittstellen. Da in der Regel die betroffenen Produkte aus wirtschaftlichen Gründen nicht sofort restrukturiert werden können, ist ein weiteres Ziel die Definition einer „Roadmap", d.h. eines Migrationsplans, nach dem die einzelnen Produkte auf die gemeinsame Architektur bzw. Plattform schrittweise überführt werden können.

Das Thema „Architekturen von Software-Systemen" hat in den letzten Jahren zunehmend an Bedeutung in der Industrie und in der Forschung gewonnen (z.B. [BCK 1998], [GP 1995], [SG 1994]). Ein Schwerpunkt ist dabei die Entwicklung von Architekturen für Produktdomänen (z. B. [GOM 1995], [WEI 1998], [BRO 1998], [OWZ 1998]). Bislang ist dabei jedoch wenig diskutiert worden, wie die Harmonisierung von heterogenen Produktdomänen als Prozeß organisiert und durchgeführt werden kann. Dieses Thema ist der Gegenstand dieses Beitrags.

2 Das Verfahren

2.1 Grundprinzipien

Das Verfahren *HAP* für die Harmonisierung von Architekturen und Plattformen ist durch drei Grundprinzipien charakterisiert:

- **Alle betroffenen Abteilungen in das Vorgehen konsequent einbeziehen:**

  Durch Workshops und Feedbacksitzungen mit Vertretern aller Organisationeinheiten, deren Produkte in die Harmonisierung mit einbezogen werden sollen, fließt ihr Know-How ein und können die Interessen der einzelnen Organisationseinheiten koordiniert werden.

  Eine gemeinsame Architektur bzw. Plattform für eine Produktdomäne entsteht darüber hinaus im Spannungsfeld einer Marketing- und Produktstrategie einerseits und den technischen Möglichkeiten andererseits. Dies bedeutet, daß die technischen Entscheidungen im Konsens zwischen Entwicklung, Marketing, Vertrieb, Service und dem Management getroffen werden müssen. Aus diesem Grund werden Experten aus allen genannten Funktionsbereichen in das Vorgehen einbezogen und gezielt zu einzelnen Workshops eingeladen.

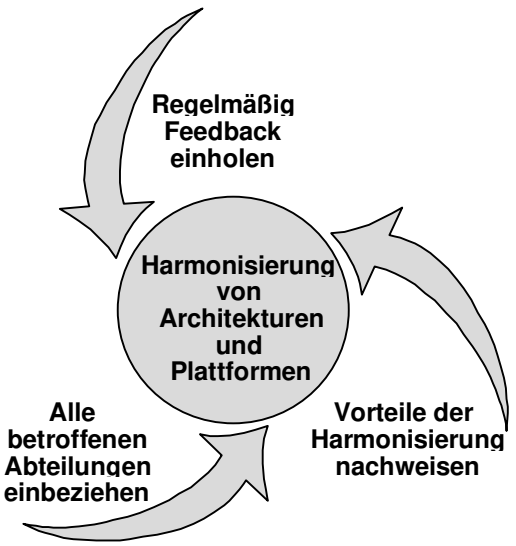

Abbildung 1: Grundprinzipien des Vorgehens

- **Vorteile der Harmonisierung nachweisen:**

  Der Nutzen einer produktübergreifenden Harmonisierung von Architekturen bzw. Plattformen sollte frühzeitig abgeschätzt und regelmäßig detailliert werden. Ziel dabei ist nachzuweisen, daß sich die Harmonisierung lohnt und die Entwicklung produktübergreifend zusammengeführt werden kann. Außerdem kann nur so der Schwerpunkt der Arbeit auf diejenigen Themen fokussiert werden, die das größte Kosten-/Nutzenpotential aufweisen.

  Dabei muß der Nutzen in einem umfassenden Zusammenhang bewertet werden. Er drückt sich nicht allein in Einsparungen bei Kosten aus. Andere Gesichtspunkte, die ebenfalls eine wichtige Rolle spielen können, sind z.B. ein verringertes Time-to-Market oder die Vereinheitlichung des Erscheinungsbildes beim Kunden. Daher müssen die Maßnahmen des Harmonisierungsprojekts vor dem Hintergrund aller Anforderungen bewertet werden, die an die Produkte und ihre Entwicklung gestellt werden.

- **Regelmäßig Feedback einholen:**

  Die Konzepte und Maßnahmen sollten frühzeitig und regelmäßig dem Management, den Verantwortlichen und Fachexperten vorgestellt werden, um die Diskussion und Steuerung des Projekts zu ermöglichen und um möglichst viele relevante Aspekte im Projekt berücksichtigen zu können.

2.2 Arbeitsphasen

Das Vorgehen ist in drei Phasen gegliedert (siehe Abbbildung 2). Dadurch ergeben sich natürliche Einschnitte im Projekt, an denen die Ergebnisse den Verantwortlichen und Entscheidern vorgestellt werden können. Dabei kann bzw. muß ihr Feedback eingeholt werden, da die Projektinhalte, Vorschläge und Maßnahmen unmittelbar ihren Verantwortungsbereich betreffen. Auf diese Weise besteht immer wieder die Möglichkeit, die Ausrichtung des Projektes zu überprüfen und - falls notwendig - zu korrigieren.

Die Organisation und Durchführung des Harmonisierungsprojekts liegt in der Verantwortung eines Kernteams aus Mitarbeitern der Entwicklungsabteilungen, die das System-Know-How mitbringen, sowie *HAP*-Experten, die Erfahrungen mit dem Verfahren aus früheren Projekten haben. Das Team wird möglichst klein gehalten, damit wesentliche Architekturentscheidungen schnell und effizient getroffen werden können.

Phase 1: Ist-Zustand und Kosten-/Nutzenfaktoren

In Phase 1 geht es zunächst darum, den Ist-Zustand zu analysieren und festzustellen, was zusammengeführt werden kann. Um das Vorgehen zu strukturieren und zu fokussieren, wird ein Fragen- bzw. Themenkatalog verwendet, der die wichtigsten architektur- und plattformrelevanten Fragestellungen und Themen für die Untersuchung zusammenfaßt. Ziel ist es vor allem,

- eine Übersicht über die Struktur der Produkte, ihre Komponenten und Schnittstellen zu erhalten,
- ihr Erscheinungsbild beim Kunden auf Einheitlichkeit hin zu prüfen,
- funktional gleiche Komponenten, die aber getrennt entwickelt werden, zu bestimmen,
- diejenigen Produkte zu identifizieren, für die sich eine Harmonisierung lohnt.

Diese Untersuchung wird im Rahmen von Workshops des Kernteams durchgeführt und, falls die Notwendigkeit besteht, durch Interviews mit ausgewählten Interviewpartnern aus der Entwicklung, dem Marketing oder dem Vertrieb ergänzt. Das Resultat der Bestandsaufnahme ist:

- die Auflistung der Produkte, die harmonisiert werden können und sollen,
- die Beschreibung ihrer Gemeinsamkeiten hinsichtlich ihrer Architekturen,
- die Identifikation des Harmonisierungspotentials für diese Produkte und
- die Definition erster Maßnahmen für die Harmonsierung der Produkte und ihrer Architekturen.

Darüber hinaus wird bereits in dieser Phase des Projekts abgeschätzt, welchen Nutzen diese ersten Maßnahmenvorschläge nach ihrer Umsetzung zeigen. Dazu gehört das Einsparpotential, das erzielt werden kann. Der Nutzen einer Harmonisierung besteht aber in der Regel auch darin, die Realisierung weiterer wichtiger Anforderungen an die Produkte und ihre Entwicklung zu vereinheitlichen. Beispiele dafür sind etwa die Einheitlichkeit des Erscheinungsbildes beim Kunden oder die Verbesserung der Wartbarkeit durch eine Reduktion der Komponentenvielfalt einer Produktdomäne. Auch wenn man diese Anforderun-

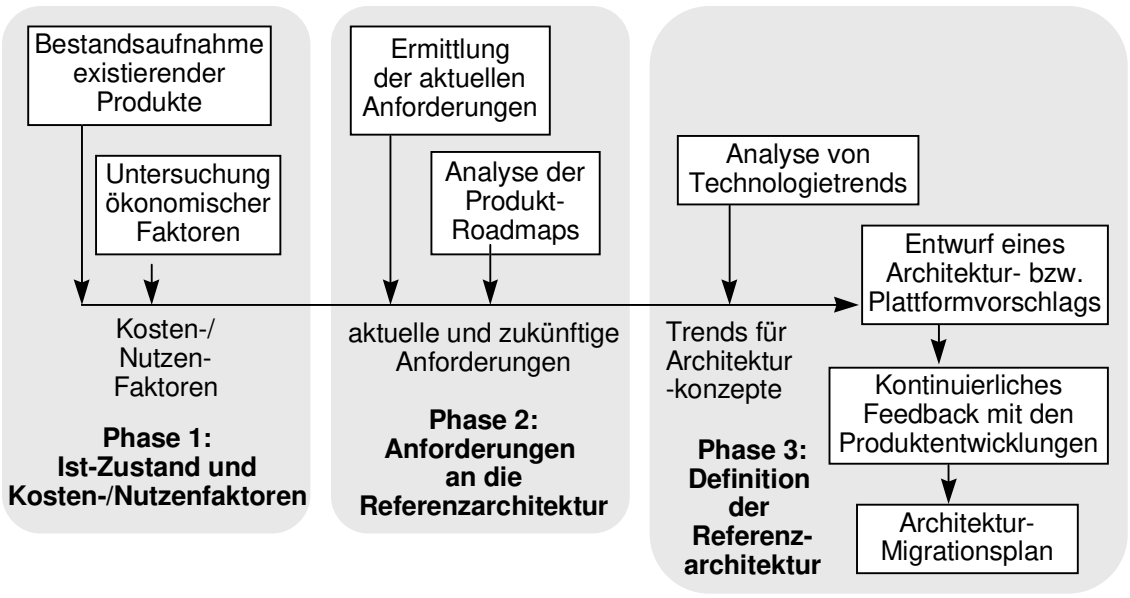

Abbildung 2: Das Verfahren im Überblick

gen zu diesem Zeitpunkt der Untersuchung noch nicht systematisch in den Blick genommen hat (siehe Phase 2), sollten sie hier berücksichtigt werden.

Das Ergebnis der Kosten-/Nutzenanalyse für eine Harmonisierungsmaßnahme wird in einem Übersichtsdiagramm zusammengefaßt (siehe Abbildung 3), das qualitativ die Einschätzungen wiedergibt. Den einzelnen Balken liegen detaillierte Analysen zugrunde, z. B. über die erwarteten Kosteneinsparungen. Liegt ein Balken z. B. auf der Höhe eines „+“ so überwiegt der Nutzen, der durch die Maßnahme erzielt wird, signifikant die Kosten, es wird also ein gutes Kosten-/Nutzenverhältnis für die Maßnahme erwartet. Natürlich kann es passieren, daß bei einer Maßnahme bzgl. eines bestimmten Kriteriums die Nachteile überwiegen. Dies wird durch einen Balken auf der Höhe des „-“ oder gar des „--“ angezeigt.

Phase 2: Anforderungen an die Referenzarchitektur

Eine Referenzarchitektur bzw. eine Plattform für eine Produktdomäne entsteht auf der Grundlage einer Marketing- und Produktstrategie. Diese definiert die Anforderungen, die an die Produkte und ihre Entwicklung jetzt und in der Zukunft gestellt werden. Während Phase 2 werden die Anforderungen systematisch erhoben und auf ihre Bedeutung für die Definition der Referenzarchitektur bzw. der Plattform untersucht. Im Mittelpunkt stehen die folgenden Fragen:

- Welche Markttrends zeichnen sich im betrachteten Produktsegment ab?
- Welche Anforderungen sind bereits in den Produkten realisiert?
- Welche Anforderungen ergeben sich aus den aktuellen Produktplanungen für die nächste Zukunft?

Zusammen mit Experten des Marketings und des Vertriebs werden Anforderungen ermittelt, strukturiert (basierend auf QFD; vgl. dazu [ZUL 1992]) und nach ihrer Bedeutung für den Markterfolg priorisiert (vgl. [SAA 1990]). Besonderes Gewicht liegt dabei auf den Fragen, welches die gemeinsamen und welches die unterschiedlichen Anforderungen an die Produkte sind und welche Anforderungen sich daraus an die Referenzarchitektur bzw. an die Plattform ergeben.

Die Ergebnisse dieser Phase sind:

- Liste der Anforderungen an die Referenzarchitektur bzw. an die Plattform,
- Gewichtung der Anforderungen nach ihrer Bedeutung für das Produktgeschäft.

Phase 3: Definition der Referenzarchitektur

In Phase 3 werden schließlich die Realisierungsmöglichkeiten bestimmt, die den verschiedenen Produkten zugrundeliegen sollen und die die Referenzarchitektur bzw. die Plattform festlegen. Dabei geht es im wesentlichen um folgende Themen:

- Welche Technologietrends existieren in dem Produktumfeld?
- Was sehen die produktbezogenen Innovationsplanungen vor?
- Wie können die bestehenden Realisierungen der einzelnen Produkte aneinander angeglichen werden?

Allein schon aufgrund der Heterogenität der Produkte gibt es in der Regel eine Vielzahl von Realisierungsvarianten. Um die bestehenden Mög-

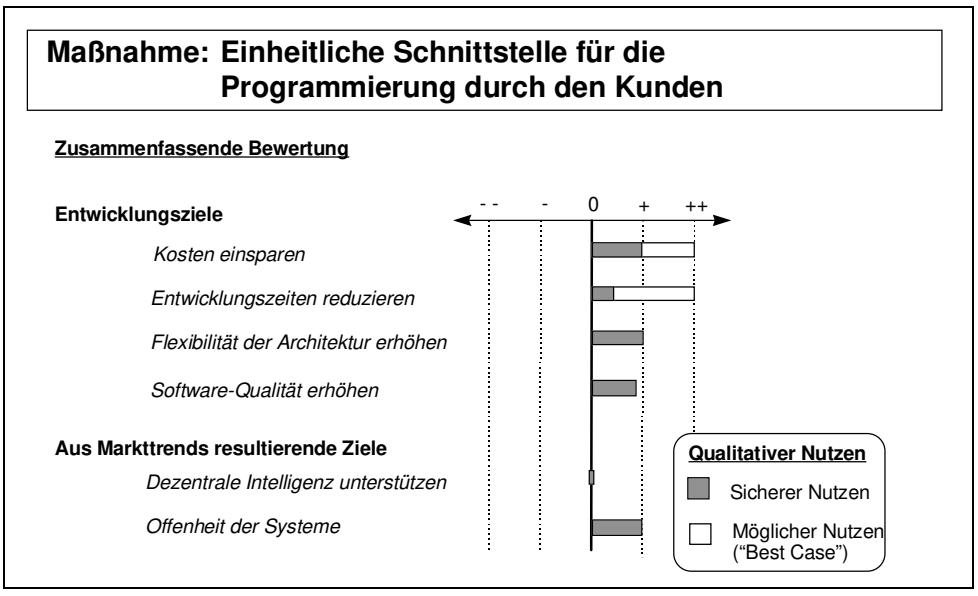

Abbildung 3: Resultat der Kosten-/Nutzenanalyse für eine Maßnahme

lichkeiten zu strukturieren und die Entscheidungsfindung transparent zu machen, werden daher in dieser Phase methodische Schritte von *System Architecture Analysis (SAA)* (vgl. [GJW 1996], [GJW 1997]) eingesetzt.

Eine Referenzarchitektur kann in der Regel nicht unmittelbar und sofort eingeführt werden. Ein solche abrupte Umsetzung der Architektur würde zu hohe Kosten verursachen und die verfügbaren Kapazitäten durch Arbeiten binden, bei denen u. U. keine für den Kunden sichtbare, neue Funktionalität entstünde. Daher wird in Phase 3 außerdem ein Migrationsplan erstellt, in dem die Einführung der neuen Referenzarchitektur auf die aktuellen Produktplanungen abgestimmt wird.

Wichtig ist in diesem Zusammenhang auch, daß die technischen Konzepte, die der gemeinsamen Referenzarchitektur zugrundeliegen sollen, und die Migrationsstrategie für ihre Umsetzung mit den Verantwortlichen und Experten kontinuierlich abgestimmt werden bzw. daß deren Feedback dazu eingeholt wird. Dieser Aspekt wurde bereits in Abschnitt 2.1 „Grundprinzipien“ angesprochen. Systematisches Feedback und kontinuierliche Abstimmung sind entscheidende Elemente, um die Transparenz der Architekturentscheidungen und ihre Akzeptanz abzusichern.

Darüber hinaus müssen die organisatorischen Voraussetzungen für die Einführungen der Referenzarchitektur geklärt werden. Entscheidend ist hier, daß die Verantwortlichkeiten für die Umsetzung und Pflege der Referenzarchitektur zwischen den entwicklenden Organisationseinheiten definiert werden. Einen konkreten Vorschlag macht hierzu der Abschnitt 3.2 „Prozeß-Harmonisierung - eine notwendige Voraussetzung“.

Die Resultate dieser Phase sind:

- Referenzarchitektur bzw. Plattform (Festlegung der Realisierungsmöglichkeiten sowie der gemeinsamen Komponenten und Schnittstellen),
- Bewertung der Realisierungsmöglichkeiten auf der Basis einer verfeinerten Kosten-/ Nutzenanalyse,
- Migrationsstrategie und organisatorische Voraussetzungen für die Einführung der Referenzarchitektur.

3 Erfahrungen aus den Harmonisierungsprojekten bei Siemens

3.1 Ein Projektbeispiel

Im Umfeld der Automatisierungssysteme sind aus Software-Engineering-Sicht zwei wichtige Trends zu beobachten. Zum einen gibt es dort eine zunehmende Dezentralisierung der Intelligenz, d.h. immer mehr Funktionalität wandert in der Peripherie der Systeme. Zum anderen - als Folge davon - steigt damit der Software-Anteil in den peripheren Geräten erheblich. Da aber lange Zeit diese Software nur eine kleine Rolle spielte, wurde ihrer Entwicklung kaum Beachtung geschenkt. Eine Abstimmung zwischen den einzelnen Produkten gab es oftmals nicht und es entstanden heterogene Systeme, die zwar funktional gleiche Komponenten enthielten, jedoch in unterschiedlichen Implementierungen.

Aus diesem Grunde wurde im Oktober 1996 ein Harmonisierungsprojekt gestartet, das für eine Produktdomäne im Umfeld der Regelungs- und Steuerungssoftware das Harmonisierungspotential untersuchen und eine gemeinsame Referenz-Architektur definieren sollte. Die Laufzeit des Projekts betrug ein Jahr und endete im Oktober 1997. Die Organisation und Durchführung lag in der Verantwortung eines Kernteams, das sich aus Experten aus der Vorfeldentwicklung und den einzelnen Entwicklungsabteilungen zusammensetzte.

Die Bestandsaufnahme zu Beginn des Projektes zeigte bereits:

- trotz der Verschiedenheit der Produkte läßt sich eine gemeinsame Struktur identifizieren,
- es besteht ein signifikantes Harmonisierungspotential,
- der Nutzen - hinsichtlich z. B. Kostenreduktion, Verkürzung der Entwicklungszeiten oder Harmonisierung des Erscheinungsbildes beim Kunden - ist deutlich höher als die Kosten.

Die Referenzarchitektur legte die neue Software-Struktur in vor allem zwei Themenbereichen fest:

- Standard-Module:

  Ein großer Teil der Steuerungs- und Regelungssoftware kann durch produktübergreifende Standard-Module realisiert werden. Für diese Module wurde ein sogenanntes Body-Interface-Konzept spezifiziert, das eine Trennung von Implementierung (= Body) und Schnittstelle (= Interface) vorsieht, die Integration der Module in die einzelnen Systeme klärt und für beide Bestandteile Standard-Vorlagen festlegt.
- Kommunikationsschnittstelle:

  Da die Systeme oft dieselben Kommunikationsschnittstellen zur Umgebung bereitstellen müssen, wurde in diesem Themenschwerpunkt das Datenaustauschformat spezifiziert, auf dem die Kommunikation beruht.

Im folgenden werden nun einige Erfahrungen angesprochen, die uns im Hinblick auf die durchgeführten Projekte besonders wichtig erscheinen.

3.2 Prozeß-Harmonisierung - eine notwendige Voraussetzung

Zu einer erfolgreichen Architekturharmonisierung gehört nicht allein die Definition der übergreifenden Architekturkonzepte und der gemeinsamen Komponenten und Schnittstellen. Da die Produkte in unterschiedlichen Organisationseinheiten entwickelt werden, müssen darüber hinaus ihre Produktplanungen und Entwicklungsprozesse einander angeglichen und das Entwicklungsgeschehen synchronisiert werden. Insbesondere folgende Themenbereiche, die die Planung der Produkte und ihre Entwicklung betreffen, spielen dabei eine Rolle:

- Produktmanagement

  Die beteiligten Organisationseinheiten müssen neue Anforderungen und ihre Spezifikationen, aber auch die Versionsplanung der Produkte aufeinander abstimmen. Dieser Abstimmungsprozeß sollte institutionalisiert sein, indem z.B. ein Gremium eingesetzt wird, in dem die verantwortlichen Produktmanager vertreten sind. Die Aufgabe dieses Gremiums besteht darin, neue Anforderungen und geplante Innovationen aufeinander abzustimmen, Gemeinsamkeiten zu identifizieren und die Verantwortung für Realisierung gemeinsamer Komponenten festzulegen.

- Realisierung der gemeinsamen Komponenten

  Für die Entwicklung der gemeinsamen Komponenten sind verschiedene organisatorische Szenarien möglich. Ist der Anteil der gemeinsamen Komponenten sehr umfangreich, dann sollte eine eigene Organisationseinheit eingerichtet werden, die für die Entwicklung der Komponenten und ihre Wartung verantwortlich ist und die sie als Lieferant den einzelnen Produktentwicklungen bereitstellt. Bei einem geringeren Umfang der gemeinsamen Komponenten ist es aber ebenso denkbar, daß eine solche Querschnittsabteilung nicht eingerichtet wird, sondern daß das oben genannte Gremium festlegt, welche der bestehenden Produktentwicklungseinheiten jeweils eine gemeinsame Komponente entwickelt und den anderen Einheiten zur Verfügung stellt.

- Erstellungsprozeß

  Um die Wiederverwendbarkeit und Portierbarkeit von Entwicklungsergebnissen zu vereinfachen, sollten die eingesetzten Methoden, Werkzeuge und vereinbarten Konventionen vereinheitlicht werden.

- Konfigurationsmanagement

  Um die Konsistenz der produktübergreifend eingesetzten Komponenten sicherzustellen, muß im Rahmen eines übergreifenden Konfigurations- und Änderungsmanagements die Bearbeitung von Fehlermeldungen und „Change Requests“ synchronisiert werden. Hauptverantwortlich sollte auch hier das oben genannte Gremium sein, das hier als ein übergeordnetes „Change Control Board“ fungiert und das die verantwortliche Entwicklungseinheit für die Umsetzung benennt. Im Fall eiliger Fehlerkorrekturen muß es allerdings möglich sein, daß diese ohne vorherigen Abstimmungsaufwand lokal bearbeitet werden und die resultierenden Änderungen erst später in den anderen Produkten eingebracht werden.

3.3 Transparente Entscheidungen erleichtern die Konsensfindung

Um einen Konsens für die Maßnahmenvorschläge erzielen zu können, muß transparent werden, auf welcher Grundlage die Entscheidungen über technische Realisierungskonzepte gefällt werden. Hier haben sich die eingesetzten methodischen Elemente von *System Architecture Analysis (SAA)* bewährt:

- Offenlegung der Bewertungskriterien

  Die Bewertungskriterien werden aus den Anforderungen abgeleitet, die vom Kunden, vom Markt oder von den entwickelnden Organisationseinheiten an die Produkte und ihre Entwicklung gestellt werden.

- Bewertung der Referenzarchitektur und der Harmonisierungsmaßnahmen

  Die Harmonisierungsmaßnahmen und die Realisierungsmöglichkeiten für die Referenzarchitektur werden jede für sich analysiert, inwieweit sie in der Lage sind, die Bewertungskriterien zu erfüllen, d. h. die gestellten Anforderungen umzusetzen.

Durch dieses Vorgehen wird sichergestellt, daß die Harmonisierungsvorschläge und die Referenzarchitektur optimal auf den Markt und den Kunden abgestimmt sind. Die Ergebnisse dieser Bewertung sind in der Kosten-/Nutzenanalyse zusammengefaßt.

Die Grundlagen für die getroffenen Entscheidungen werden somit transparent und ermöglichen die Konsensfindung.

3.4 Hemmnisse für eine Architektur- und Plattform-Harmonisierung

In vielen Bereichen unseres Unternehmens findet man heute vertikalisierte Orgainsationsstrukturen vor, die sich an den verschiedenen existierenden Produkten orientieren. Eine Harmonisierung der Produktstrukturen steht dieser Organisationsform zumindest teilweise entgegen, da mit einer gemeinsamen Architektur und Plattform auch Entwicklungsabläufe übergreifend koordiniert werden müssen (siehe Abschnitt 3.2). Verantwortlichkeiten müssen neu und übergreifend definiert werden.

Bei einer Harmonisierung müssen außerdem lokale Interessen, die die Entwicklung eines einzelnen Produktes optimieren wollen, und globale, produktübergreifende Interessen aufeinander abgestimmt werden.

Das Verfahren *HAP* integriert die verschiedenen Interessensgruppen, bezieht sie explizit in das Vorgehen mit ein und wägt den Nutzen gegen die entstehenden Kosten und gegen den Koordinierungsaufwand ab.

5 Ausblick

Dieser Beitrag hat das Verfahren *HAP* für die Harmonisierung von Software-Architekturen und Plattformen für Produktdomänen vorgestellt. Die Herausforderungen eines solchen Projekts liegen in der Definition gemeinsamer technischer Realisierungskonzepte einerseits und in der zielführenden und transparenten Vorgehensweise andererseits.

Die Erfahrungen der vorgestellten Projekte zeigen, daß produktübergreifende Synergien genutzt werden können und - wegen des bestehenden Verbesserungspotentials - auch genutzt werden sollten.

6 Literatur

[BCK 1998], Bass, L.; Clements, P.; Kazman, R.: Software Architecture in Pratice, Addison-Wesley, Reading, Massachusetts, 1998.

[BRO 1998], Brownsword, L.: CelsiusTech - A Case Study in Product Line Development, In: Bass, L.; Clements, P.; Kazman, R.: Software Architecture in Pratice, Addison-Wesley, Reading, Massachusetts, 1998, S. 345 - 374.

[GP 1995], Garlan, D.; Perry, D. E.: Introduction to Special Issue on Software Architecture, IEEE Transactions on Software Engineering, Vol. 21, No. 4, 1995.

[GJW 1996], Gloger, M.; Jockusch, S.; Weber, N.: Assessment and Optimization of System Architectures, Proceedings of Second IEEE International Conference on Engineering of Complex Computer Systems, pp. 400-407, 1996.

[GJW 1997], Gloger, M.; Jockusch, S.; Weber, N.: System Architecture Analysis - Optimizing Architectures in the Industrial Context at SIEMENS, European Software Engineering Process Group Conference, Part C308, 1997.

[GOM 1995], Gomaa, H.: Reusable Software Requirements and Architectures for Families of Systems, Journal of Systems Software, Vol. 28, 1995.

[KMK 1993], Kolland, M.; Mehner, T.; Kuhn, K.-J.: Software-Plattformen - Mehr als ein Schlagwort, Wirtschaftsinformatik, Nr. 35, pp.23-31, 1993.

[OWZ 1998], Oberndorf, P.; Wallnau, K.; Zaremski, A. M.: Product Lines - Reusing Architectural Assets within an Organization, In: Bass, L.; Clements, P.; Kazman, R.: Software Architecture in Pratice, Addison-Wesley, Reading, Massachusetts, 1998, S. 331 - 344.

[SAA 1990], Saaty, T. L: How to make a decision? The analytical hierarchy process, European Journal of Operational Research, Vol. 48, No. 1, pp. 9-26, 1990.

[SG 1994], Shaw, M.; Garlan, D.: Software Architecture: Perspectives on an Emerging Discipline, Prentice Hall, 1994.

[WEI 1998], Weiss, D.: Commonality Analysis - A Systematic Process for Defining Families, 2nd International Workshop on Development and Evolution of Software Architectures for Product Families, February 1998.

[ZUL 1992], Zultner, R. E.: Quality Function Deployment (QFD) for Software - Satisfying Customers, American Programmer, February 1992, pp. 1-14.