System Lines -
Elements of the Software Product Lines Approach for the Construction of Corporate Information Systems

Christian Wege und Frank Gerhardt
DaimlerChrysler AG
{wege, fg}@acm.org

Abstract

The software product lines approach has some shortcomings when developing a corporate information system. We propose the system line approach which leverages ideas of the software product lines approach but with modifications which makes it fit the specific needs of corporate information systems.

Introduction

Existing literature about software product lines contains a large number of examples from practice. However it is hard to find examples covering the use of the software system lines approach for the development of corporate information systems[1]. The reason for this is that existing software product line approaches don’t cover the specific challenges which exist when creating corporate information systems.
To proof this we first discuss the needs of information systems in a large corporation.– especially the needs specific for a software product line approach. We show the missing elements of the common software product line approaches. This leads to the system line approach which leverages ideas of the software product line approach for the construction of corporate information systems. We conclude with some examples out of our practice.

Role of the IS department

The primary task of the information systems is to support the process chains of the main products of a company. Those processes each consist of a set of individual tasks which help reach the strategic goals. [FeEtAl92]. The sum of these processes is the essence of large corporation. [SuEtAl97]. (see Diagram 1)
Those processes are normally separated from the organization. Inside the information systems the data are the backbone of the organization. In many cases there is a large infrastructure already in place. Normally new systems have to leverage this existing infrastructure (e.g. for strategic or other cost related reasons). A new system must be able to integrate into this existing system environment [AcbEtAl94].

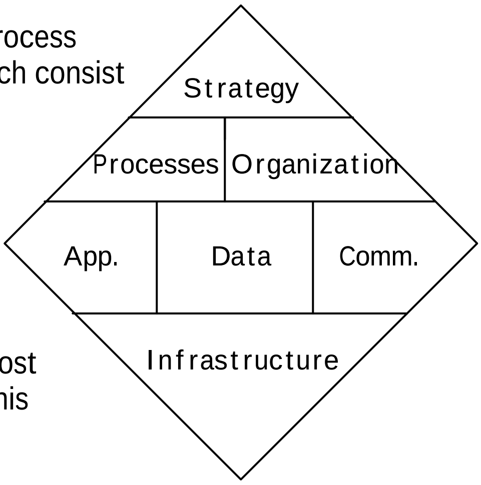

Diagram 1

The challenges for system development in such an environment grow non-linearly with the size of the corporation. The complexity explodes.

[1] Information systems in are large corporation are not the same as large information systems in general. Large information systems are typically found in large companies. But mediums sized and small information systems are found as well. However they all share the needs for operability, product-specific configuration know-how, existing infrastructure, related costs and strategy.

Goals of the software product line approach

The classical goals of the software product line approach are reduction of development costs, improvement of quality, shorter time-to-market, and reduction of maintenance costs [Bos00]. Software product line approaches are mainly found in areas where similar products have to be brought to market fast and one after the other. The delta of the new product compared to the previous one is often small – sometimes only some new features are added (e.g. the different models of a cell phone manufacturer).
One possibility to reach the envisioned goal is reuse. The software product line approach shows a way to enable reuse. This reuse means the use of one component in more than one application or product.

Reuse in a large corporation means to a large degree reuse of experience made in practice. This experience covers:

- the used technology,
- the used products,
- the integration into the existing infrastructure and
- the knowledge about the related business processes and the organization.

The software product line approach however aims at the reuse of software. For the construction of large corporate information systems an approach has to be found which is tailored to these specific needs.

System line approach

Normally not all business supporting applications are developed at a central location. Therefor there must be another way to ensure that new systems integrate into the existing infrastructure and are interoperable with existing systems.

On the system level we talk about the application oriented view and the technical view (compare to the 4+1 views of Kruchten [Kru95]). The application oriented view represents the system from the point of view of the customer. It contains the functional model, the data model, the execution model and the integration model. The technical view contains the programmatic realization of the functional model, the layer model, the distribution model and the integration model [AcEtAl94] (see Diagram 2).

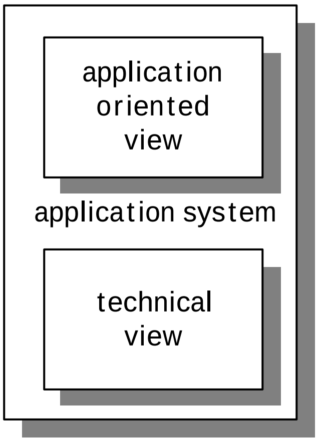

Diagram 2

One of the main goals of the application system architecture is to ensure that the non-functional requirements are met [BaEtAl98] (like portability, security, maintainability, etc.).
Those non-functional requirements are more and more ensured by infrastructure components (middleware). Application development can concentrate on the implementation of the business logic [OrHa98]. This means the application architecture is determined more and more by the existing infrastructure which is used by the new system.

Technical Architecture

The infrastructure contains a set of technologies (respectively products which implement those technologies), which have a strong influence on the architecture of the application systems which are built on top of them (e.g. communications middleware or message oriented middleware). The technical architecture is the sum of all these products and technologies.

Selection criteria for the technical architecture

The selection criteria for components of the technical architecture can be categorized as follows[2]:

- Organizational factors
  - Operation capabilities in the corporation
  - Support capabilities in the corporation
  - Is the complexity of the product manageable by the development team and the operation team?
- Technological factors
  - Technology fits to existing infrastructure
  - Is technology established in the industry?
- Product-related factors
  - Quality of the product (performance, stability, etc.)
  - Stability of product vendor
  - Support of standards
  - Costs for purchase, development and operation
  - Support capabilities of vendor and internal organization

Features of the technical architecture

The technical architecture is the base for the design of information systems. The definition of the elements of the technical architecture enables the seamless deployment and the operation of an information system in the corporation. Changes must be restricted to small increments so that running applications are not affected. In a large corporation you normally find a big amount of experience how the used products have to be installed, configured and operated and the related processes are tailored accordingly. The enduring knowledge about the running systems. enables their integration with new systems. But to make this happen new systems must integrate into the existing environment. Generally the technical architecture and the related processes have a tendency to slow down innovation.

Scoping of system lines

The system line approach emphasises the horizontal dimension of the domain. I.e. the standardized components cover several systems [CzEi00]. The vertical dimension is more or less similar to the canonical architecture of information systems (see diagram 3).

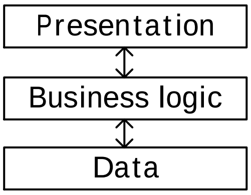

Diagram 3

Case studies

A framework to streamline applications

To support the business processes between the car service stations and the corporation several applications were developed throughout several years. The clean-up these applications a framework was created during the last two years which streamlines the applications in the areas of::

- Communication between car service station and head quarter
- Architecture of the server based part of the application
- Presentation to the user

An example for the application of this framework is the parts catalog, which enable a more comfortable and more efficient use compared to the terminal based predecessor.

[2] Hofmeistet al use the same categories for the selection of a concrete software architecture.

At first sight all these applications have the same GUI. But also the lower layers base on the same components and services. The architectures of the different systems have much in common and only small differences.

Component collection for applications with Java and HTML based user agents

Based on the experiences with the above described framework we currently develop some components which provide often used services like security, logging, etc.
These components provide access to the existing infrastructure. E.g. the security component allows to access the corporate-wide LDAP directory and to other security related systems for authentication and authorization. This facilitates the user management for applications with many users – the users are already in the "corporate directory" and are maintained by existing business processes. The components are usable for diverse kinds of applications (fat client oder thin client). Through the use of this component suite the application gains more independence from the used products and has an optimal access to the existing systems in the corporation. The operation of these systems is ensured and the related processes are adapted accordingly. E.g. the user management in the corporate directory can be coupled to the relevant HR systems. The architecture of new systems of more or less a copy of existing systems which use the same components. This pays back especially in the case of outsourcing the application development The use of the components established by the corporation enables the applications be integrate into the existing infrastructure. We have – so to say – a product line for fat client and thin client applications.

Deployment of a web application server

Web applications have to have a high degree of availability, security, performance, etc. These non-functional requirements should be covered by the use of a web application server. The Java 2 Enterprise Edition [J2EE00] is an established de-facto standard in this respect. However on top of this you have to standardize on one product (or a small number of products). The increasing delegation of responsibility to such a product results in a bigger complexity of the product. The development team and the operation team have to be capable understand and manage this complexity. Without corporate standard each outsourcer would bring in his or her own infrastructure. The issue of system integration starts to become a beast. Despite the fact that web applications can be built on top of many different products and technologies, the product related factors are more and more important. The architecture of the new system depend to a large degree from what is manageable in the corporation.

Summary

This paper shows a gap in the software product line approach for the construction of large corporate information systems. We show some features typical for those systems relevant for a software product line.
Some examples document the system line approach in practice.

Provocatively said a software product line approach is adequate for green field development but fails facing a complex system infrastructure.

Literature

[AcEtAl94] Ackermann, Altmeier, Ayerle, Helmbrecht, Mett, Reimann, Reitter, Schwarzmüller, Seeberger: Wege zu IV-Anwendungsarchitekturen, 1994. Interner Bericht der DaimlerBenz AG.

[BaEtAl98] Len Bass, Paul Clements, Rick Kazmann: Software Architecture in Practice. Addison-Wesley, 1998.

[Bos00] Jan Bosch: Design and Use of Software Architectures. Addison-Wesley, 2000.

[CzEi00] Krzysztof Czarnecki, Ulrich W. Eisenecker: Generative Programming. Addison-Wesley, 2000.

[FeHu92] Feiler, P and Humphrey, W. Software process development and enactment: Concepts and definitions. Software Engineering Institute, CMU/SEI-92-TR-04, Pittsburgh, PA, Sept. 1992

[HoEtAl99] Christine Hofmeister, Robert Nord, Dilip Soni: Applied Software Architecture.

[J2EE00] Java 2 Enterprise Edition Architecture Blueprints. SUN Microsystems, 2000. http://java.sun.com/j2ee/blueprints/

[Kru95] Kruchten: The 4+1 View Model of Architecture, IEEE Software (12) 6, 1995.

[OrHa98] Robert Orfali and Dan Harkey: Client/server programming with Java and CORBA. Wiley, 1998.

[SuEtAl97] Giancarlo Succi, Luigi Benedicenti, Paolo Predonzani, Tullio Vernazza: Standardizing the reuse of software processes. StandardView (5) 2, 1997.

[SuEtAl00] Giancarlo Succi, Tullio Vernazza: Strategic Software Production With Domain-Oriented Reuse (erscheint im September 2000)