Product Line Engineering for Global Development

Daniel J. Paulish
Siemens Corporate Research, Inc.
755 College Road East
Princeton, NJ 08540 USA
+1-609-734-6579
daniel.paulish@scr.siemens.com

Abstract: This paper describes how product line engineering practices can be used to better plan and manage global development projects. Software products are growing in complexity and the development organizations to implement new features are also growing in staff size. An approach is summarized to decompose large-scale requirements into a well-structured set of software components that can be developed in parallel among globally distributed development teams. The approach applies best practices of software requirements engineering including business object modeling coupled with product line architecture design. Agile development processes are exploited so that a collection of small, distributed application component development teams are controlled by a central organization. It is expected that the approach will result in substantial time-to-market and productivity improvements by application of modern industrial practices in the areas of requirements, design, and organization patterns.

Keywords: Project management, project planning, software architecture, high-level design, global development, requirements engineering, organization patterns, agile processes.

Background

Over the past several years, the LookOut Corporation has acquired a number of companies for the purpose of broadening its product offerings. These products have been developed at multiple sites located in the United States, Europe, and Australia, and they have been sold primarily into regional markets. Recently, the Chief Technical Officer of LookOut has conducted an analysis of current products, and he has identified eight existing products that perform similar management and control functions. If these current products could be integrated into a single product line with a common look-and-feel suitable for the global market, LookOut's management believes they could be both more competitive and minimize worldwide development costs.

The Chief Technical Officer has commissioned an architecture design team to analyze the existing products' functionality and propose a product line architecture that will live well into the future. Existing products will need to incrementally migrate to the new architecture based on their business need; i.e., a relative priority based on projected unit sales and the age of the current product. The design team has been challenged to initially define an optimal long-lived architecture as a vision of where the product line should be going, and then to analyze individual product migration approaches as a secondary step. In addition to the design work, the team has also been asked to estimate the development cost for implementing the new architecture and to propose a global development process to implement it.

A number of organizational and political barriers are facing the Chief Technical Officer in achieving the project goals. Divisions who are doing well with their existing products have less interest in the project than those with older less successful products. The existing products have been developed using different development processes and technologies. Each division's development manager is reluctant to give up key staff to work on the global product line, since it impacts support of the current products. Furthermore, everyone is concerned what types of staff reductions may result when the existing products are eliminated. To partially address these barriers, the design team has a representative from each of the major development sites and they are temporarily collocated within one facility. They are using a new product line oriented process to do their design and planning tasks, and it is assumed that each development site will eventually participate in the incremental product line development. Thus, a global development process is required to implement the product line for the global market.

Approach

The approach used for product line engineering for global development consists of improved practices in three key areas:

1. Requirements engineering: Model the functionality of Lookout's current products using current best practices and drive all aspects of the product line solution development from the model.
2. Software architecture: Drive the product line towards standard common data models and a component framework that will help enable integration.
3. Global development: Optimize the organization of product solution development using small distributed component development teams synchronized by a central organization.

Lookout's approach begins with a machine-readable description of the existing product requirements modeled in UML. An architecture framework is designed using multiple views [1,2], and it is designed to address the organizational, technological, and product factors unique to this product line to be sold into the global market [2,3]. The requirements model is decomposed such that functionality is mapped to software components that are defined within the architecture framework. Component development teams distributed among Lookout's development sites are commissioned to implement the components. A central product line management team controls the requirements model and architecture, and it provides the incremental product planning such that existing products are migrated or developed to fit within the new architecture and user interface. Component development is commissioned such that components making up an individual application or product within the product line are developed within one site.

Lookout's central product line management organization provides rules of thumb, centralized processes, interface specifications and development constraints to the distributed component development teams. The requirements model and architecture are designed such that component sizes are defined to be relatively small with a maximum specified size (in terms of lines of code (LOCs), function points, development time, development effort). The distributed development teams are constrained with respect to functionality, delivery schedule, effort, and schedule for developing their commissioned components, but they are free to use whatever local agile processes [4] they desire as long as they meet the constraints. Central product line management synchronizes the concurrent

development of the planned components and their functionality and is responsible for component integration.

Requirements Engineering

Lookout's Product Line Management organization contains a small team responsible for developing and controlling the requirements model described in UML [5] (Figure 1). This requirements engineering team is made up of subject matter experts who know each of the existing products. The model is defined broadly in the beginning, with lower levels of detail as the functionality is incrementally added. Since the model is machine readable, software tools are used to support the analysis of the model to identify errors and inconsistencies, provide requirements extraction for process artifacts, provide constraints on maximum component size, and generate the test cases that will be used to validate the implemented products.

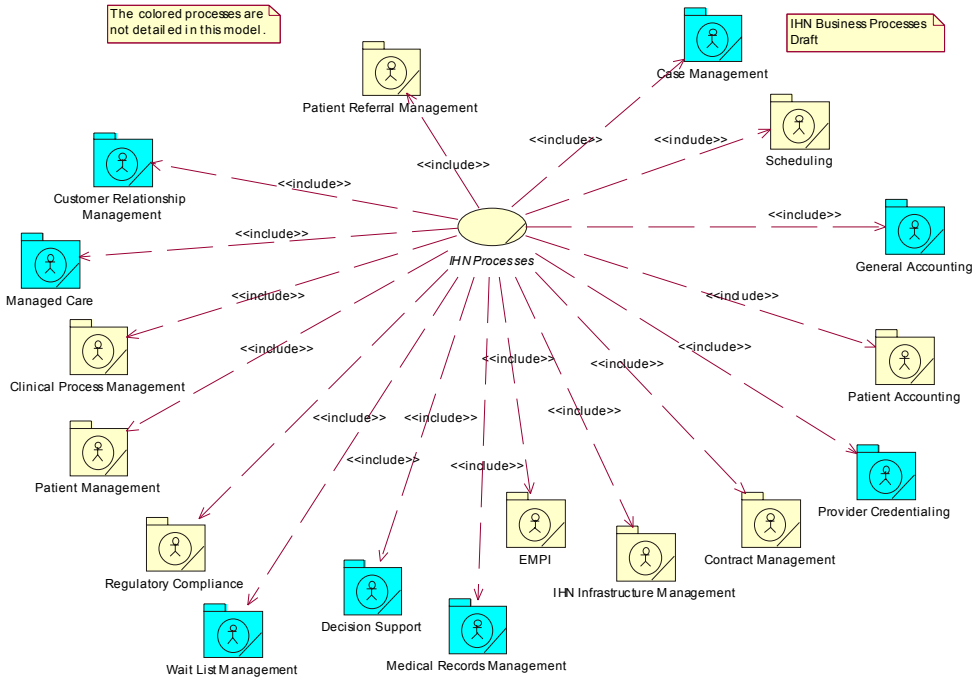

Figure 1. Example requirements model.

Software Architecture

The central organization also consists of a small architecture design team who develops the architecture framework that is made up of all the components to meet the product line's functionality (Figure 2). Within the framework, lower layered components will be mostly purchased, and the application divisions will develop higher layered components or existing application components will be restructured and wrapped to fit into the framework. The requirements model will be decomposed to allocate features to the designed application components. The architecture team will apply rules of thumb to limit the number of components in the design to around 85-150 [3]. Furthermore, the

architecture will be designed such that no individual component will be larger than about 100 KLOCs of functionality and can be developed within 12 months by a 10-person component development team. Using these guidelines, LookOut believes they can scope the product line to provide the desired functionality within a maximum size of about 15M LOCs.

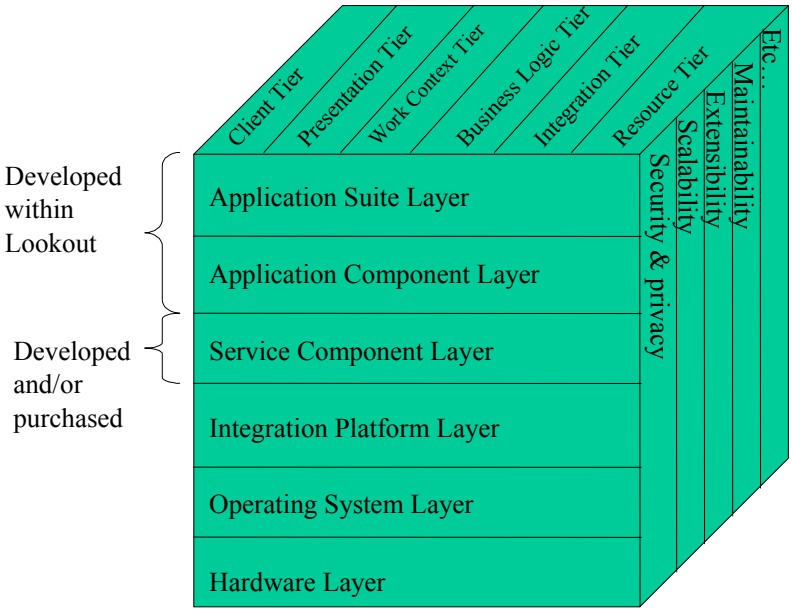

Figure 2. Example Architecture Framework (credit: Sun Microsystems [6]).

Organization

When all the components have been defined within the high-level design, the central product line management team will commission the distributed development sites to implement each application component for the first products to be released in the product line. The information that central product line management will provide to the distributed teams when they are commissioned includes:

- Part of the requirements model to implement
- Acceptance tests
- Incremental development plan and integration dates
- Component interface specifications
- Vertical slice implementation.

Thus, by commissioning the distributed application teams, the central team will control the functionality, schedule, and quality of each component that will make up the product line. The distributed teams will be free to apply whatever development process they wish as long as they meet the constraints imposed by the central team. Integration will be done incrementally, every 6-8 weeks [3], such that the development of the components will be planned, controlled, and synchronized by the central team. An example organization for implementing the product line is given in Figure 3.

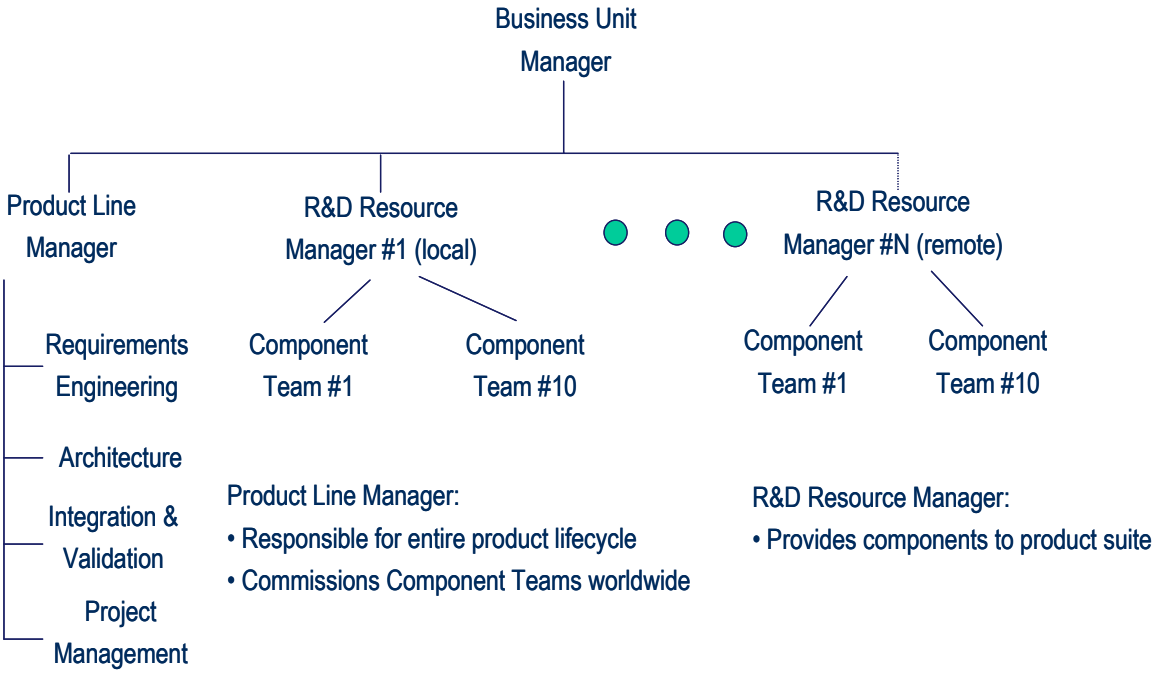

Figure 3. Example organization.

Success Factors

The following activities will help LookOut be successful with developing their global product line.

- Develop a machine-analyzable, visual requirements model that is centrally controlled.
- Centrally design and enforce an architecture framework.
- Decompose requirements and map them to software components within the architecture before commissioning the small distributed teams.
- Use agile processes within the small component development teams.
- Centralize incremental integration planning.
- Implement a vertical slice model.
- Synchronize communications among all development team leaders daily.
- Follow the rules of thumb concerning component size, development schedule, and team size.

Summary

In this position paper, we have provided some guidelines about how to develop a product line using a centralized product line management team and distributed component development teams. Using best practices for formal requirements engineering and architecture design provides the basis for commissioning the distributed teams to develop the software components that will make up the product line within a planned synchronized process. The distributed teams are kept small by restricting the functionality designed into each component. Thus, agile processes can be used within the component development teams and communications overhead among the teams is reduced.

Today's very large software systems take many years to develop using very large development teams. By using component-based architecture design approaches, restricting the size of components, and applying incremental development, the initial products within a product line can be installed in user sites within a couple years.

References

1. P. Clements et al, *Documenting Software Architectures: Views and Beyond*, 2003, Addison-Wesley, Boston, MA.
2. C. Hofmeister, R. Nord, and D. Soni, *Applied Software Architecture*, 2000, Addison-Wesley, Reading, MA.
3. D. Paulish, *Architecture-Centric Software Project Management: A Practical Guide,* 2002, Addison-Wesley, Boston, MA.
4. A. Cockburn, *Agile Software Development*, 2002, Addison-Wesley, Boston, MA.
5. S. Robertson and J. Robertson, *Mastering the Requirements Process,* 1999, Addison-Wesley, Reading, MA.
6. *Sun Journal*, "Take a 3-D Approach to Architecture Design", Volume 5, No. 1, http://www.sun.com/executives/sunjournal/v5n1/feature2.html.