Software Product Lines

Minsoo Ryu

Hanyang University

msryu@hanyang.ac.kr
Contents

- Basic Concept
- Core Asset Development
- Product Development
- Management
- Practice Areas
Software Product Line

- Software product lines refers to engineering techniques for creating a portfolio of similar software systems from a shared set of software assets using a common means of production

  - Manufacturers have long used analogous engineering techniques to create a product line of similar products using a common factory that assembles and configures parts designed to be reused across the varying products in the product line

  - For example, automotive manufacturers can now create tens of thousands of unique variations of one car model using a single pool of carefully architected parts and one factory specifically designed to configure and assemble those parts
Successful Reuse through SPL

- The idea of manufacturing software from reusable parts has been around for decades, but success has been elusive
  - Recent advances in the software product line field have demonstrated that narrow and strategic application of these concepts can yield order of magnitude improvements in time-to-market, quality, portfolio scalability and software engineering cost
  - The result is often a discontinuous jump in competitive business advantage, similar to that seen when manufacturers adopt mass production and mass customization paradigms
Predictive Reuse

- The characteristic that distinguishes software product lines from previous efforts is predictive versus opportunistic software reuse
  - Rather than put general software components into a library in hopes that opportunities for reuse will arise, software product lines only call for software artifacts to be created when reuse is predicted in one or more products in a well defined product line
Contents

- Basic Concept
- Core Asset Development
- Product Development
- Management
- Practice Areas
Three Essential Activities

- Core asset development and product development using the core assets, both under the aegis of technical and organizational management
  - New products can be built from core assets, or core assets can be extracted from existing products
  - Often, products and core assets are built in concert with each other
Core Asset Development

- The goal of the core asset development activity is to establish a production capability for products
  - This activity, like its counterparts, is iterative
  - Input: contextual factors
  - Output: product line scope, core asset base, production plan
Input: Four Contextual Factors

- **Product constraints**
  - Commonalities and variations of the same family of products
  - Behavioral features, relevant standards, performance, interfaces, quality requirements, …

- **Production constraints**
  - Time budget, engineers’ capability, process standards, …

- **Production strategy**
  - Overall strategy for realizing both the core assets and products
  - Proactive (starting with a set of core assets and spinning products off of them) or
  - Reactive (starting with a set of products and generalizing their components to produce the product line core assets), or using some combination of the two

- **Preexisting assets**
  - Legacy systems and existing products
  - Libraries, frameworks, algorithms, tools, components, and services mined from legacy systems
  - Externally available software
Output: Product Line Scope

☐ The product line scope is a description of the products that will constitute the product line or that the product line is capable of including
  ▪ In its simplest form, the scope may consist of an enumerated list of product names
  ▪ The scope of the product line must target the right products

☐ The product line scope must be defined carefully
  ▪ If the scope is too large and product members vary too widely, the core assets will be strained beyond their ability to accommodate the variation, economies of production will be lost
  ▪ If the scope is too small, the core assets might not be built in a generic enough fashion to accommodate future growth

☐ The scope definition of a product line is itself a core asset, evolved and maintained over the product line's lifetime
Output: Core Asset Base

- The core asset base includes all the core assets, which are the basis for the production of products in the product line
  - An architecture that the products in the product line will share
  - Software components that are developed for systematic reuse across the product line
  - Any real-time performance models or other architecture evaluation results associated with the product line architecture
  - Test plans, test cases, and all manner of design documentation
  - Requirements specifications
  - Domain models
  - Commercial off-the-shelf (COTS) software
  - Management artifacts such as schedules, budgets, and plans
  - Any production infrastructure such as domain-specific languages, tools, generators, and environments
  - Training specific to the product line
  - Technical management process definitions
  - Set of identified risks for building products in the product line
Output: Core Asset + Attached Process

- Each core asset should have an associated attached process that specifies how it will be used in the development of actual products.
A production plan prescribes how the products are produced from the core assets and fills two roles

- It includes the process to be used for building products
  - The set of attached processes, together with the necessary process "glue" to join them together into a coherent whole, define the production process for products
  - The attached processes and the (usually nontrivial) glue are designed to satisfy the production strategy and production constraints and reflect the chosen production method

- It lays out the project details to enable execution and management of the process and therefore includes such details as the schedule, bill of materials, and metrics
Output: Production Plan

- The production method, which is the overall implementation approach, specifies the models, processes, and tools to be used in the attached processes across core assets.
Contents

- Basic Concept
- Core Asset Development
- Product Development
- Management
- Practice Areas
Product Development

- The goal of product development activity is to produce particular products
  - Input: the three outputs of core asset development—the product line scope, the core assets, and the production plan—plus the product description for each individual product
  - Output: products, feedback, new assets, product constraints
Inputs

- **The product description for a particular product**
  - Often expressed as a delta or variation from some generic product description contained in the product line scope (such a generic description is, itself, a core asset)

- **The product line scope**
  - This indicates whether it's feasible to include the product under consideration in the product line

- **The core assets from which the product is built**

- **The production plan**
  - This details how the core assets are to be used to build the product
Product Builders

- Product builders use the core assets, in accordance with the production plan, to produce products that meet their respective requirements
  - Product builders also have an obligation to give feedback on any problems or deficiencies encountered with the core assets, so the core asset base remains healthy and viable
Contents

- Basic Concept
- Core Asset Development
- Product Development
- Management
- Practice Areas
Management

- Management plays a critical role in the successful fielding of a product line
  - Activities must be given resources and then be coordinated and supervised
  - Management at both the technical (or project) and organizational levels must be strongly committed to the software product line effort
Organization Management

Organizational management

- Identifies production constraints and ultimately determines the production strategy
- Creates an organizational structure that makes sense for the enterprise and makes sure that the organizational units receive the right resources (for example, well-trained personnel) in sufficient amounts
- Is responsible for the ultimate success or failure of the product line effort
- Determines a funding model that will ensure the evolution of the core assets and then provides the funds accordingly
- Orchestrates the technical activities in and iterations between the essential activities of core asset development and product development
- Ensures that these operations and the communication paths of the product line effort are documented in an operational concept
Technical Management

☐ Technical management

- Oversees the core asset development and product development activities by ensuring that those who build core assets and products are engaged in the required activities, follow the processes defined for the product line, and collect data sufficient to track progress

- Decides on the production method and provides the project management elements of the production plan
Finally, some individual or group should be designated to either fill the product line management role and act as a product line champion or find and empower one

- That champion must be a strong, visionary leader who can keep the organization squarely pointed toward the product line goals, especially when the going gets rough in the early stages
- Leadership is required for software product line success. Management and leadership are not always synonymous
All Three Together

- Each of the three activities (core asset development, product development, and management) is individually essential, and their careful blending is also essential—a blend of technology and business practices.
Chicken and Egg Issue

- Many organizations begin a software product line by developing the core assets first
  - These organizations take a proactive approach

- Other organizations begin with one or a small number of products they already have and then use them to generate the product line core assets and future products
  - This approach is reactive
Chicken and Egg Issue

Both of these approaches may be attacked incrementally

- For example, a proactive approach may begin with the production of only the most important core assets, rather than all of them
  - Early products use those core assets
  - Subsequent products are built using more core assets as they are added to the collection
  - Eventually, the full core asset base is fielded
- An incremental reactive approach works similarly; the core asset base is populated sparsely at first, using existing products as the source
  - More core assets are added as time and resources permit
Chicken and Egg Issue

- The proactive approach has obvious advantages—products come to market extremely quickly with a minimum of code writing.

- But there are also disadvantages:
  - It requires a significant up-front investment to produce the architecture and the components that are generic (and reliable) across the entire product space.
  - And it also requires copious up-front predictive knowledge—something that is not always available.
Chicken and Egg Issue

- The reactive approach has the advantage of a much lower cost of entry to software product lines because the core asset base is not built up front
  - However, for the product line to be successful, the architecture and other core assets must be robust, extensible, and appropriate to future product line needs
  - If the core assets are not built beyond the ability to satisfy the specific set of products already in the works, extending them for future products may prove too costly

- Whatever the approach, the process is rarely, if ever, linear
Contents

- Basic Concept
- Core Asset Development
- Product Development
- Management
- Practice Areas
Product Line Practice Areas

☐ To be able to carry out the three essential activities, you must master the practice areas relevant to each and apply them in a coordinated, focused fashion
  ▪ By "mastering," we mean an ability to achieve repeatable, not just one-time, success

☐ A practice area is a body of work or a collection of activities that an organization must master to successfully carry out the essential work of a product line
  ▪ Practice areas help to make the essential activities more achievable by defining activities that are smaller and more tractable than a broad imperative such as "develop core assets."
  ▪ Practice areas provide starting points from which organizations can make (and measure) progress in adopting a product line approach for software
Three practice areas

- Software engineering practice areas are necessary for applying the appropriate technology to create and evolve both core assets and products.
- Technical management practice areas are necessary for managing the creation and evolution of the core assets and the products.
- Organizational management practice areas are necessary for orchestrating the entire software product line effort.
Software Engineering Practice Areas

- Architecture Definition
- Architecture Evaluation
- Component Development
- Mining Existing Assets
- Requirements Engineering
- Software System Integration
- Testing
- Understanding Relevant Domains
- Using Externally Available Software
Technical Management Practice Areas

- Configuration Management
- Make/Buy/Mine/Commission Analysis
- Measurement and Tracking
- Process Discipline
- Scoping
- Technical Planning
- Technical Risk Management
- Tool Support
Organizational Management Practice Areas

- Building a Business Case
- Customer Interface Management
- Developing an Acquisition Strategy
- Funding
- Launching and Institutionalizing
- Market Analysis
- Operations
- Organizational Planning
- Organizational Risk Management
- Structuring the Organization
- Technology Forecasting
- Training