The emerging practice of software product line development

By Charles W. Krueger, PhD

A new class of software development methods, tools, and techniques is emerging that greatly simplifies the engineering of product line portfolios. The improvements are so large that they impact the fundamentals of how companies do business.

In today's customer-driven environment, most companies target the needs of their prospective customers by creating a product line – a portfolio of closely related products with variations in features and functions – rather than just a single product. For example, a company that develops a radar countermeasures system will not be content selling that technology to just one client on one platform, but rather will see great economic benefit to engineering many variations of that product. Product variations could encompass different requirements for separate branches of the military, cockpit manufacturers, export rules and localizations for various countries, and a diversity of feature options and price points to support a wide range of anticipated and unanticipated customer requirements.

For companies that utilize embedded software in their products, this product diversity poses a serious problem. Tools and techniques for software development tend to focus on individual products. As a result, developing software for a *product line* is extremely complex due to multiple intertwined products, features, and production deadlines – all aimed at moving target markets. These tactical software development challenges are big enough to impede the realization of business-critical goals and strategies.

A new class of software development methods, tools, and techniques – collectively referred to as *software product line development* – is emerging to address this problem, offering improvements in development time to market, cost, quality, and portfolio scale and scope. What is most interesting is the *magnitude* of tactical and strategic improvements that are possible, not measured in percentage points but more commonly in factors of 2 to 10. These improvements are so large that they impact the fundamentals of how companies compete.

The genesis of software product line development methods

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 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.

> **"The idea of manufacturing software from reusable parts has been around for decades, but success has been elusive."**

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 orders-of-magnitude improvements in software engineering capability. The result is often a discontinuous jump in competitive business advantage, similar to that seen when manufacturers adopt mass production and mass customization paradigms.

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.

The concepts

Software product lines can be described in terms of four simple concepts, as illustrated in Figure 1:

- ***Software asset inputs:*** A collection of software assets – such as requirements, source code components, test cases, architecture, and documentation – that can be configured and composed in different ways to create all of the products in a product line. Each of the assets has a well-defined role within a common architecture for the product line. To accommodate variation among the products, some of the assets may be optional and some of the assets may have internal variation points that can be configured in different ways to provide different behavior.
- ***Decision model and product decisions:*** The decision model describes optional and variable features for the products in the product line. Each product in the product line is uniquely defined by its product decisions – choices for each of the optional and variable features in the decision model.
- ***Production mechanism and process:*** The means for composing and configuring products from the software asset inputs. Product decisions are used during production to determine which software asset inputs to use and how to configure the variation points within those assets.
- ***Software product outputs:*** The collection of all products that can be produced for the product line. The scope of the product line is determined by the set of software product outputs that can be produced from the software assets and decision model.

Software product line development approaches provide a shift in perspective so that development organizations can engineer their entire portfolio as though it were a single system rather than a multitude of products.

The benefits

The benefits of the software product line approach come in the form of tactical improvements in software engineering – deploying software products faster, cheaper, and better. However, what is most interesting is that these tactical improvements are often large enough to have an impact well beyond the borders of the engineering department, offering strategic competitive benefits.

Time to market

If new products in a software product line are engineered and brought to market faster, strategic business benefits result. Companies with software product line success stories have reported decreasing their time to market for new products by factors of 2 to 50 when compared to the conventional techniques they were using.

Quality

Quality benefits of software product lines can be measured in two ways. The first is how well each product matches the needs of each customer. The mass customization capabilities of software product lines directly address this measure of quality. The second is the rate of defects found in each of the products in the product line, which can also be significantly improved due to software reuse. Companies have reported reductions in defect rates as high as 96 percent.

Engineering cost

Software product line techniques can significantly increase the productivity of software engineers, seen as a reduction in the effort and cost required to develop, deploy, and maintain a portfolio. Typical productivity improvements reported in case studies range between a factor of 2 to 3, though higher factors are not uncommon.

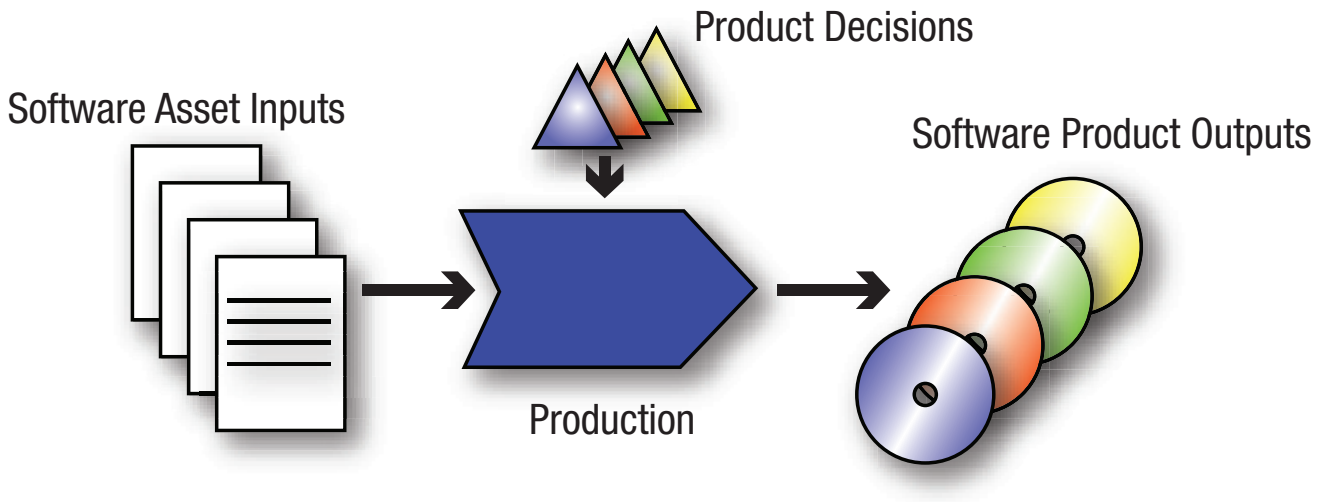

Figure 1

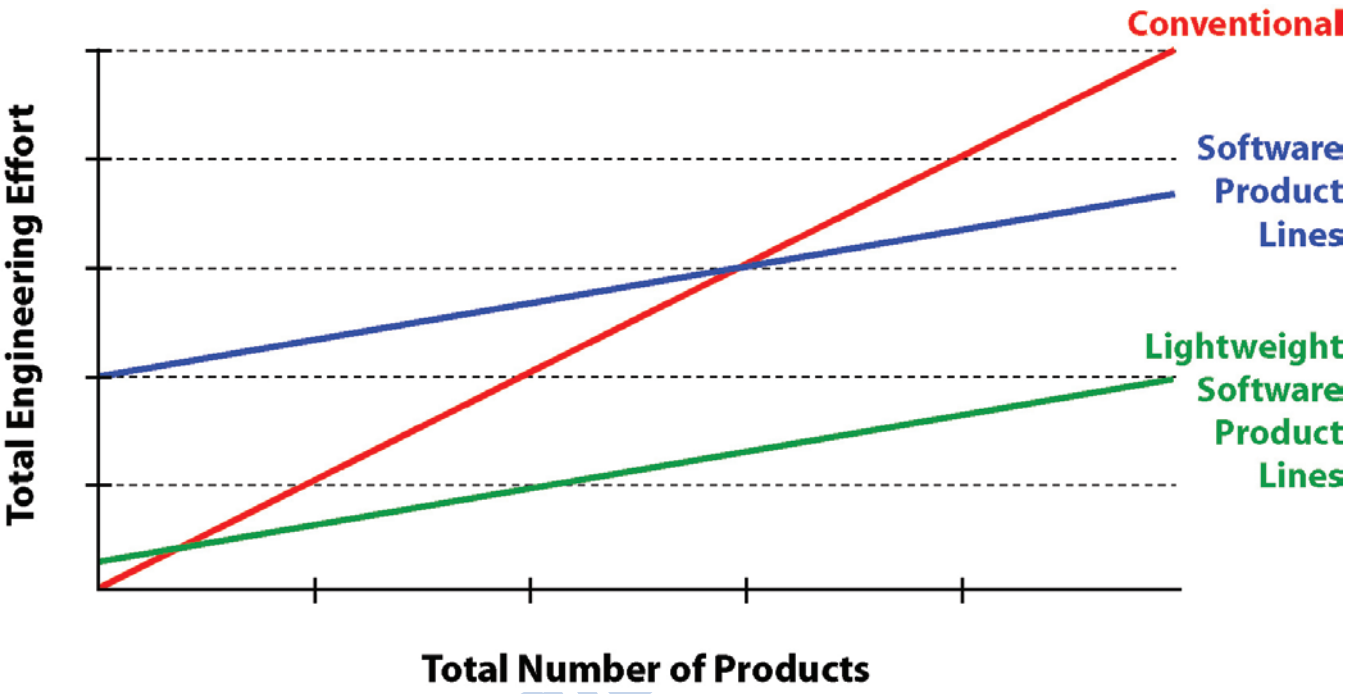

Figure 2

Figure 2 illustrates the effort – and thus cost – required to develop, deploy, and maintain a portfolio. The red line represents a conventional productivity line. The blue line represents a productivity line of the pioneering efforts in software product lines, where a significant up-front effort was typically required (seen as the high Y-axis intercept) to launch a software product line. The green line represents the new generation of software product line methods that can achieve higher productivity gains with much less up-front effort.

Scalability

A company that takes a software product line approach needs to scale, without constraints, to whatever number of products is optimal for the business. The benefit of software product line approaches is that they can often scale to orders-of-magnitude more products in a portfolio than conventional software engineering techniques.

Advantages of software product line development

The emerging practice of software product line development offers significant tactical software engineering improvements as well as strategic business advantages. In much the same way that manufacturers advanced from manual labor to mass production to mass customization, software product line development allows software development organizations to advance from labor-intensive to highly efficient and automated portfolio development methods. To learn more about these methods and successes stories, see the community website www.SoftwareProductLines.com.

Charles W. Krueger, *PhD, is founder and CEO of BigLever Software and has 20 years of experience in software product line development. He has helped companies establish software product line practices, including* Software Product Line Hall of Fame *inductees Salion and LSI Logic. He is the author of more than 25 articles, including ACM* Software Reuse. *He is a frequent organizer and speaker for the International Software Product Line Conferences and moderates the community website SoftwareProductLines.com. He received his PhD in Computer Science from Carnegie Mellon University.*

To learn more, contact Charles at:

BigLever Software, Inc.
10500 Laurel Hill Cove
Austin, TX 78730
Tel: 512-426-2227
E-mail: ckrueger@biglever.com
Website: www.biglever.com