Product Line Management 101

Efficiently Managing Product Families with Multiple Variants

TABLE OF CONTENTS

THE DILEMMA OF GROWING PRODUCT COMPLEXITY

“Complexity kills,” Ray Ozzie, American software industry entrepreneur once **famously wrote** back in 2005. “It sucks the life out of developers; it makes products difficult to plan, build, and test; it introduces security challenges, and it causes user and administrator frustration.”

If software development seemed complicated back then, imagine how much more so it is today. The leap from physical servers to virtual computers, containers, and microservices, from on-premise to distributed cloud environments, and from Waterfall to Agile and DevOps, mirrors the colossal leap in complexity when it comes to developing software.

On top of that, the consumerization of IT has caused the demand for consumer-grade browser-based and mobile applications to skyrocket. Customers expect top-quality, seamless user experiences which are connected, secure, and feature-rich by design.

Add to that key advancements in hardware, namely portable and handheld devices outshining traditional desktop set-ups with increased processing power, speed, and memory, and developers have even more complexity to contend with as they try to scale software and hardware capabilities in parallel. And as a backdrop to all this: consumers have come to expect personalized products, resulting in mass customization that adds to the complexity challenge.

Here are some of the key challenges that developers face that increase software complexity as a whole:

RELENTLESS DEMAND FOR PERSONALIZATION & CONFIGURATION OPTIONS

Nowadays, customers don't just want personalization – they demand it. Hardly any company creates just one version of a product, and consumers expect the option of variety and customization from the get-go. That's why product lines (a range of similar products or services sold by the same company) are common across industries, from automotive to healthcare, consumer electronics, software applications, and more.

But producing and maintaining product lines, like software, has also become incredibly complex. While offering multiple options gives companies a competitive advantage, juggling increasingly elaborate product lines is pushing organizations to their limit. On top of that, these traditional ways of producing them often struggle to keep up with the pace of today's market and customer demands:

- **Clone-And-Own (C&O)**
  In the past, product developers used the clone-and-own approach which involves the construction of new product variants based on existing ones. However, the efficiency of C&O decreases in proportion to the size of the product family, because it only lets developers benefit from reusing once, at the time of reusing the base product.

- **Product-centric development**
  In product-centric development, each individual product is developed independently from other products, or like in C&O begins as a clone of an existing product which is then modified to meet the needs of the new variant.

Nowadays, millions of dollars and thousands of costly man-hours are spent on mundane and redundant tasks, when they could be geared towards product innovation instead. As businesses everywhere face constant pressure to decrease costs, speed up time-to-market, remain competitive, and provide frequent updates, the need for a more efficient way to produce and maintain product lines becomes more pressing.

That's where Product Line Engineering comes in.

PLE GLOSSARY: KEY TERMS & THEIR MEANINGS

1. Product Line
A collection of closely-related products that are variants of each other. Products within a product line are built using a shared base or architecture. They can include any combination of software, systems in which software runs, or non-software systems associated with software-representable artifacts.

2. Assets
Assets are the building blocks that make up products in the product line. These can be any element associated with the engineering lifecycle of the products, such as requirements, design specs, test plans, documentation, and much more.

3. Variant
This is a version of an artifact or a product that has a different set of characteristics than other artifacts or products in the same product line.

4. Feature catalogue
As the name suggests, this is a catalogue of features that is available for each product variant. Using this catalogue, you can pick a feature to apply to a specific variant of a product. This helps teams speak the same language and remain on the same page when it comes to product variations.

5. Bill-of-features
The bill-of-features is created using the feature catalogue and specifies which features are selected for each product. This provides clarity to stakeholders about how the product line is built up.

6. PLE Factory Configurator
The PLE Factory Configurator is a software tool that takes product specifications as inputs and configures shared assets accordingly. By applying the bill-of-features to the shared assets, the Configurator is able to decide which features and content to include in which product variant.

7. Product asset instances
This is what the PLE factory produces as outputs. Product asset instances contain the complete set of shared artifacts or assets which relate to the product in question.

WHAT IS PRODUCT LINE ENGINEERING (PLE)?

Systems and Software Product Line Engineering, often shortened to Product Line Engineering or PLE, is a software engineering methodology borrowed from the manufacturing industry. PLE involves creating and maintaining a portfolio of related products using shared engineering assets and an efficient means of production, rather than creating each product individually. The key to successful PLE is taking advantage of replicable similarities between products, while also managing their differences efficiently.

In classical product-centric engineering, it's common for the same tasks to be repeated again and again for each product. This means that every time something is added, fixed, or re-written to one product variant, the same changes need to be applied to all the other variants one by one. Using resources to do this kind of work is not only a waste of money, but also keeps engineers focused on low-value routine work instead of high-value tasks, and fails to scale in the long run.

In a PLE approach, the product line is considered a single entity rather than a group of separate products that require individual management. In order to achieve this in practice, a common underlying hardware and software architecture is created to describe the base platform. This base platform describes what the products within the line have in common, as well as what the planned variations will be. In this way, shared product assets can be flexibly reused across the product line, while a subset of variations defines the differences between individual products within the portfolio.

According to the Software Engineering Institute, PLE brings the following benefits to product development organizations:

Benefits of PLE

1. Large-scale productivity improvement
2. Faster time to market
3. Higher product quality (and lower risk)
4. Increased organizational agility
5. Higher customer satisfaction
6. The ability to support mass customization
7. A more competitive market presence
8. More ability to encourage and maintain growth

IMPLEMENTING PLE IN PRACTICE

Changing technology is easy. Changing people? Not so much. Although the benefits of implementing product line engineering are clear, and increase as the number of product variants grows, implementing the framework in businesses can still be a challenge.

To get started, here is a three-stage process that multiple organizations have used to implement PLE incrementally:

Stage one:
To begin with, stage one concentrates on incorporating a PLE Factory approach into the organization. At this point, the business can start using PLE to develop a feature model and shared assets for the product line. The management of both shared and unique assets is made easier by specialized tools.

Stage two:
Next, the organization re-categorizes existing product assets into collections of shared assets that contain variation points. When getting started, you can decide to engineer variation points into some (but not all) shared assets, to implement stage two incrementally. This also means that work needs to be divided differently within the team, so new roles specific to the PLE framework will be defined at this point. These new roles will transition team members from product-specific tasks to asset-specific roles instead.

Stage three:
Finally, the organization can start to define products with new features or feature combinations, which will enable them to move towards strategic product line engineering as a whole.

Note: these steps do not need to be completed one after another. Organizations have found different sequences and combinations which work best for them, the most important thing being to proceed incrementally rather than rushing to do it all at once. Each incremental step brings the organization a little bit closer to receiving the full benefits that the product line engineering framework has to offer.

PLE INDUSTRY SUCCESS STORIES

Aegis Combat System

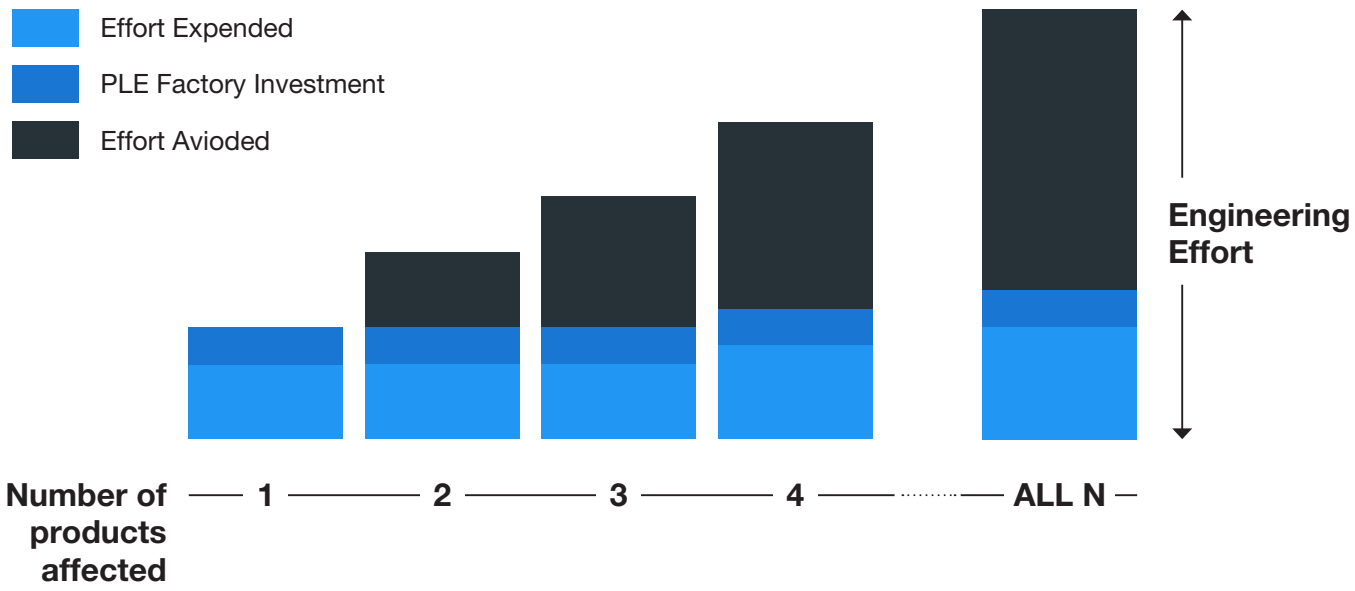
Testing can be an exhausting process for any system, let alone systems as huge and complicated as AEGIS, in which testing can account for up to half of the overall cost of system management. Using PLE to share verification and validation activities across as many product variants as possible, AEGIS Weapons System was able to avoid some **$47 million in costs** per year.

U.S. Army Live Training Transformation

The U.S. Army Live Training Transformation (LT2) has a product line of combat training systems. Prior to their PLE transformation, live training systems and devices were made up of individual products developed and implemented separately which made it extremely difficult and costly to achieve commonality and interoperability between systems. Using PLE methodology, **the U.S. Army** was able to improve their speed of migration from 10 years to 1 to 2 years, as well as generate over $300 million in cost avoidance across the development of live training systems.

General Motors

General Motors' product line of vehicles is often referred to as "mega-scale product line engineering" due to the extremely large product set, complex products, and complicated feature variation. By defining an internally consistent model for their vehicle product line using PLE shared assets and consistent feature configurations, subsystems, components and hardware allocations, they are already seeing a **huge return on their PLE implementation**.



THE RIGHT TOOLING REDUCES COSTS, DELAYS, AND TIME TO MARKET

In order for your development team to develop the right product variants for the right markets, while reducing rework, shortening release cycles, and cutting the costs of product verification, you need the right tooling in place.

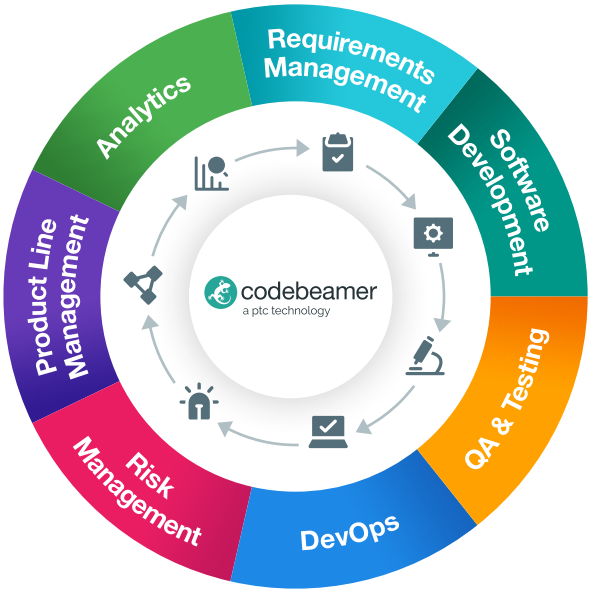
Using a mature, integrated, next-generation Application Lifecycle Management platform like Codebeamer allows you to:

1. Large-scale productivity improvement
2. Faster time to market
3. Higher product quality (and lower risk)
4. Increased organizational agility
5. Higher customer satisfaction

Try PTC's Codebeamer technology now

Start your free 30-day trial of Codebeamer in minutes:

Start Trial



ABOUT CODEBEAMER / SOURCES

About Codebeamer

Codebeamer is an Application Lifecycle Management (ALM) platform with unique configurability and product line configuration capabilities.

Codebeamer X is an integrated Engineering Lifecycle Management (ELM) platform for life sciences companies with regulatory process & compliance support.

To learn more, visit us at **https://intland.com/**. Follow **@intland** on **Linkedin** and **Facebook**.

About PTC (NASDAQ: PTC)

PTC® enables global manufacturers to realize double-digit impact with software solutions that enable them to accelerate product and service innovation, improve operational efficiency, and increase workforce productivity. In combination with an extensive partner network, PTC provides customers flexibility in how its technology can be deployed to drive digital transformation—on premises, in the cloud, or via its pure SaaS platform. At PTC, we don't just imagine a better world, we enable it.

PTC.com **@PTC** **Blogs**

Sources

'Complexity kills': Can Microsoft, and all of IT, simplify?
On the Dilemma of Software Complexity
Product-centric Development is Dead: Long Live Product Line Engineering
Product Line Engineering Success Stories
Product Line Engineering International Working Group
Software is Increasingly Complex. That Can Be Dangerous.
The theory behind product line engineering
What is Product Line Engineering?
What is product line engineering (PLE)?
INCOSE PLE Primer

DIGITAL TRANSFORMS PHYSICAL