

### 3. Domain-Specific Engineering

Software product engineering is an approach for building a singular evolving software product that will suit the needs of a particular customer (a one-of-a-kind product) or of a simple market (a one-size-fits-all product), The *Domain-specific Engineering (DsE) methodology* builds on and extends that approach to provide the means for rapidly building any of an envisioned set of similar software-based products.

The motivation for a focus on similar products is the opportunity it provides to leverage development effort involved in building multiple products and evolving them over time. Similarity takes the form of supporting capabilities that are similar in concept while differing in particular essential aspects. There may be differences in the capabilities a product will support, in the relative importance of quality factors and tradeoffs, in how a product is to be used, in the context in which a product will be used, in the types of changes the product needs to accommodate over time.

The focus of DsE is to avoid arbitrary and incidental differences among similar products while providing the means for each product to be flexibly customized to reflect essential differences. By formulating a set of products as addressing similar needs, the development of a given product can focus on resolving essential aspects that differ without spending undue effort on arbitrary differences or on aspects that are shared with other products. The leverage that DsE provides is in coalescing all essential differences into a coherent lattice of choices through which customers can delineate a product that best fits their needs.

The basis for a DsE effort is the recognition of a coherent market for similar products and the competence to build those products. The objective of DsE is to provide each customer with a product that is customized to fit their specific needs, both initially and as those needs change, with a fraction of the time and effort required to build a product conventionally.

The realization of a DsE effort is a *domain*, consisting of the knowledge (expressed in a product family) and expertise (expressed in a product manufacturing process) required

to build the type of products needed by customers comprising the targeted market. A *product family* is a unified representation of an envisioned set of similar products (generalizing the program family concept of Dijkstra and Parnas<sup>1</sup>). Membership of the set is limited to products that share particular functionalities and characteristics in common. Individual instances of that set are differentiated by choices (i.e., “decisions”) that customers are offered concerning the capabilities and properties of a product that will best fit their particular needs. The associated product manufacturing process is a specialized process (with associated methods, practices, and tools) for the decision-based derivation of customized instance products from the product family. Both the derived products and the domain itself can be evolved over time as market needs and enabling technology change.

Beyond encompassing all aspects of software product engineering, DsE addresses key elements of its context for a system-level scope: the totality of program management concerns, systems engineering, hardware engineering, and customer practices. This broadened scope enables a unified “whole-product” perspective for a disciplined approach to the coordinated development of a collection of similar and related software-based products. Furthermore, DsE targets building not only needed customized products but also the means by which those products are realized in hardware and software along with the associated materials and practices needed for the effective production and use of those products.

## The DsE Process

Following DsE, a product development program establishes its market focus, defines a product family that encompasses the products that customers need, and creates the process by which a project can build an instance of that family for a targeted customer. Engineering effort is leveraged across all past, current, and future products to achieve the manufacturing capability that streamlines the effort needed to build any individual product. Each product can be iteratively refined as development proceeds and the customer’s specific needs become better understood or change. Each deployed product

---

<sup>1</sup> D.L.Parnas, “On the Design and Development of Program Families”, IEEE Transactions on Software Engineering SE-2 (1), March 1976, 1-9. <<https://doi.org/10.1109/TSE.1976.233797>>

can then be progressively revised as the customer's needs change over time. Products can further be periodically improved as the domain itself evolves over time to offer improved capabilities needed by current and new customers.

The DsE methodology defines an iterative tripartite process (Figure 3.0-1):

- *Program Management*, the process by which an enterprise initiates and sustains a viable effort to engineer and manufacture products for an associated coherent market
- *Domain Engineering*, the program-defined process for developing a domain, as the means by which projects can build instance products
- *Product Manufacturing*, the domain-defined process by which a customized instance product is built for a designated customer

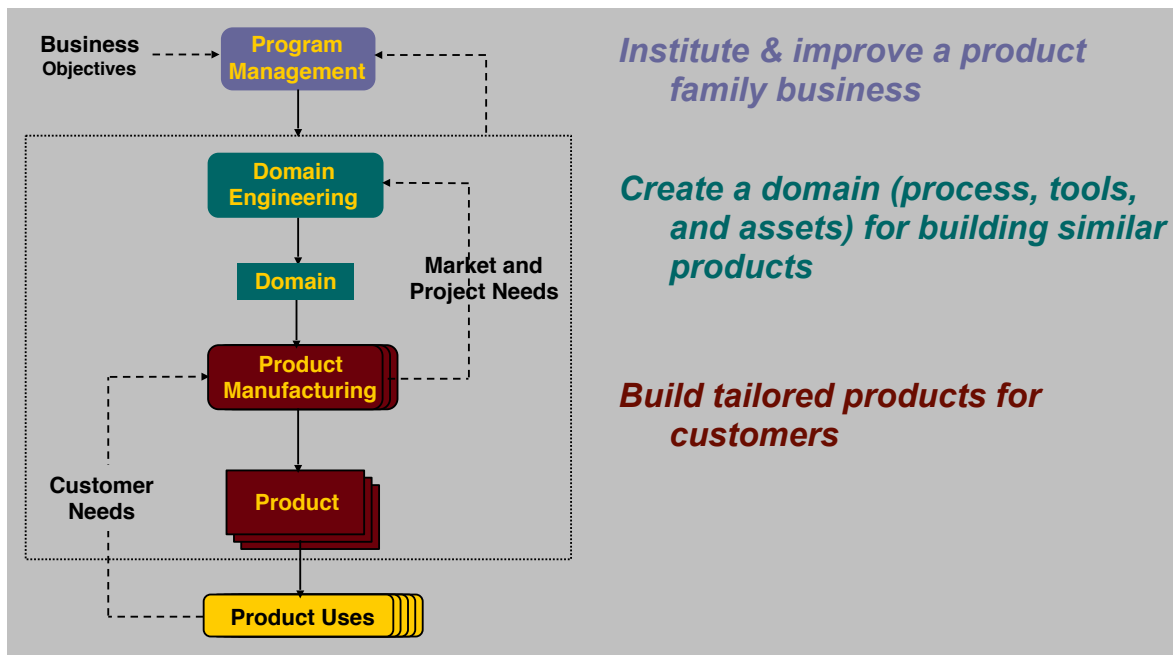


Figure 3.0-1. The DsE Tripartite Process

Each instance product is derived as a new or revised product for an individual customer or simple market. Multiple instance products can be derived concurrently or serially over time, for different customers or circumstances or to permit individual customers to explore alternatives and tradeoffs.

A product family represents a set of similar products; all differences in these products, corresponding to perceived differences in customers' needs or associated engineering tradeoffs, are indicated by an associated set of deferred *decisions*. Each instance of a family is uniquely characterized by its corresponding resolution of these decisions. Whether deriving a new product or a revision to an existing product, the associated effort is primarily focused on resolving uncertainty concerning actual needs and how to resolve deferred decisions so as to best characterize those needs to identify the derivable product that is the best fit. Because every product is derived as an instance of the product family, uncertainty and change as well as diversity among customers' needs are all inherently addressed in the DsE approach. Revisions to an existing product will normally require less effort due to there being less unresolved uncertainty concerning needed product capabilities to be explored.

<-----

Both market composition and associated customer needs change over time. Similarly, the means to manufacture products as represented by a domain also change over time. The premise underlying DsE is that similar problems are amenable to similar solutions. The more basic notion is that in building products for a coherent market, those products will be similar, with differences due to specific rationale in how customers' businesses differ. With a product family, products correspond to potential solutions to the similar needs of customers in a targeted market. The organizing medium for DsE is a *domain*, as the adjunct to that market and consisting of a product family and a manufacturing process that defines how to specify and derive instances of the product family.

The notion of a coherent market is that it corresponds to a set of potential customers that need similar but not identical products. Further, it is reasonable to determine what differences among customers' needs motivate differences in resulting products. Needs that are sufficiently equivalent correspond to the same product behavior; needs that differ sufficiently result in differing product behavior. DsE defines how to build products that are the same as other products in some respects while being different in other respects.

-----

## challenges

- managing co-evolution of program and domain with market
- managing evolution of family instances by means of decision model, as customer needs and domain changes
- determining how decision alternatives affect quality factors
- verifying family versus derived or example instances
- adjusting to unbuildable product model, mapping to alternative closest matching products

----->