3.1 The Context for DsE

This section characterizes essential aspects that provide context for understanding the DsE methodology: the nature of a domain-specific approach, the duality of diversity and change, key elements of software product engineering context subsumed by DsE (program management, hardware engineering, systems engineering, and enhanced customer collaboration), and an economic framework for managing a DsE program.

The Nature of a Domain-specific Approach

The objective of a development program is to build products for a targeted market. A program that pursues this objective by means of a domain is "domain-specific". With a domain-specific approach, efforts are leveraged by focusing on products that satisfy similar needs. A domain defines the products that can be built, both initially and as needs and circumstances change.

For a product to be viable to build and use, its capabilities must fit within the ability of developers to conceive and understand. The complexity of a product that is viable to build has increased over time as competence in building software has grown but it is not unbounded. The question then is how to decide what the bounds are for products to be viable to build.

The motivation for building a product is to enable capabilities needed to achieve the objectives of a customer endeavor. The nature and scope of that endeavor is subjectively decided. The purpose of a domain is to characterize products that will effectively address a market in which customers' endeavors and problems are similar enough to warrant similar solutions.

The Concept of a Coherent Market

DsE is conceived as a viable approach due to targeting products to a coherent market. A market is coherent if it represents customers that have similar needs. The definition of coherence is subjective but not arbitrary. The motivation is to identify an opportunity for leverage in the realization of products that are a good fit to each customer's needs.

The premise of DsE is that similar problems are amenable to similar solutions. A single product may suffice for all (or a subset of) customers if their needs are sufficiently similar but essential differences can require different products. The goal with a domain-specific approach is to build customized products as needed without having to duplicate most of the effort that different products would conventionally incur.

Even if some customers might be viewed as constituting a simple market, having needs that are similar enough to be met by a single product, DsE allows them to be addressed independently. No customer need receive a product that is constrained to fit other customers' differing needs. Products can be customized to each customer's needs and can converge or diverge relative to other products as customers' needs change.

Any essential differences in what customers need can be identified and individually addressed with only incremental additional effort and without repeated effort on common portions. By focusing efforts on the means and materials needed to build only similar products, effort need not be expended on redoing well-understood and previously solved portions shared with previously built similar products. Efforts can focus on exploring poorly understood and unprecedented capabilities that arise with diverse or changing market and customer needs and evolving technology.

In general, any problem will have many feasible solutions, even after eliminating solutions that differ superficially but produce equivalent behavior in actual use. When solutions differ, meaningful differences are traceable to ways in which customers' needs differ. Limiting the range of problems that can be solved to only those needed by a particular market reduces the solutions to be built to ones that are both more likely to be needed and more alike.

The Concept of a Domain

A ***domain*** is the "product" of an engineering effort to facilitate the manufacture of products as instances of a product family. The objective for a domain is to enable the realization of products that provide capabilities needed by customers in a targeted coherent market. With DsE, a domain includes both a concrete representation of a product family as a set of similar products—excluding products that do not satisfy the

similarity criteria for included products—and an associated means for mechanically deriving instances of that family.

For a provider, a domain is a capital investment in the means of production for a family of products suited to an evolving market. A domain institutionalizes the product development competence on which that investment depends. By anticipating the future capabilities needed by customers in a coherent market, a provider can reduce the time and effort needed to deliver products responsive to those needs.

Ultimate effectiveness of a DsE program is achieved through the coevolution of a domain and its targeted market. The domain changes as the composition of the market, the needs of customers in that market, and enabling technologies change. As the domain changes, its buildable capabilities influence customer and aggregate market perceptions of how their endeavors can be improved.

The Role of Developer Decisions in Product Manufacture

Development is a decision-making process for building a product that will best serve an intended purpose. With a product family as the expression of a set of similar products, differences that distinguish realizable instances of the family are formulated as deferred decisions for developer resolution (developers need consider only how included products are distinguished from each, not how they differ from excluded products).

A developer determines the needed product by resolving the deferred decisions, either intuitively based on past experience or systematically by exploring alternatives and associated tradeoffs, to correctly reflect the problem and its best solution. As deferred decisions are resolved, the candidate set of buildable products as expressed in the product family is reduced to a subset and ultimately to a single instance product. The developer modifies the resolutions of relevant deferred decisions if understanding of the actual problem or its solution changes.

With DsE, if behavior that is common to the product family or a subfamily needs to be changed, the effort to make the change is incurred by domain engineering. Making a change in the product family means that it will propagate uniformly into every product

to which it applies, simply by rederiving each product with no need for changes in its specification.

Ideally with a DsE approach, two developers expressing the same problem will produce differing but equivalent descriptions (e.g., using different problem nomenclatures) but those descriptions will result in products exhibiting equivalent behavior. The degree to which different descriptions reflect essential differences in the problem or preferred solution will primarily influence the degree to which the resulting products' behaviors differ.

In some cases, different resolutions of a decision seem equally good but impose different tradeoffs or no resolution is entirely satisfactory. For this, a domain can support multiple resolutions to realize different products to be comparatively evaluated for a best approximation to needs. This result can be augmented by determining whether the domain should be subsequently modified to enable building a product that is a better fit.

The Duality of Diversity and Change

Three causes of product change were discussed earlier (in section 1.3). A fourth cause of product "change" (or more broadly, the rationale for the existence of multiple similar products) is diversity of needs. Product change concerns needs changing over time whereas market diversity concerns the co-existence of similar but differing needs within a market. Product change and market diversity are both manifestations of *variability*.

Product variability represents the different realizations of a product that result from problem-solution uncertainty, deficiency, and change. Every customer is susceptible to the implications of product variability.

Market variability reflects essential differences in customers' needs, including as market composition changes over time. A simple market is one in which only product variability, not market variability, is a concern. The implication of variability in a more diverse coherent market is the need to provide multiple products to address the needs of the market as a whole.

Customers in a coherent market, by definition, will have similar needs but those needs may nevertheless differ sufficiently to preclude those customers being properly satisfied by a single product. The idea of diversity is that different products are needed either because they address different needs or because they address the same needs differently. Two products are considered to be "similar" to the degree that they exhibit similar behavior (with products that exhibit essentially identical behavior being considered "equivalent"). Such diversity may exist among customers in a market or even for a single customer facing diverse operational circumstances (e.g., differing legal or regulatory constraints, computational configurations, or ecosystem compositions).

Any given product realization may be the result of either product or market variability or both as the composition of simple markets within a coherent market changes due to the needs of different customers converging or diverging over time. (Whether a particular product realization is characterized as a "version" of a previously existing product versus as a different product is only a matter of whether it is addressing the changed needs of a single customer or the differing needs of different customers. Both cases exemplify variability.)

Resolving either product or market variability can be viewed as choosing among a set of alternative realizable products. The product that best fits a customer's current needs may differ from the product that best fits their needs at some future time. Similarly, the product that best fits one customer's needs may differ from the product that best fits another customer's needs. Both types of variability can be expressed in the form of a product family.

{domain vs product lifecycles and evolution}

Encompassing Elements of Context as DsE Concepts

With basic software product engineering, elements of context are taken as constraints imposed on how the product is to be realized. With DsE, these elements are reconceived as integral aspects of the overall effort.

Program Management

The objective of program management under DsE is to establish an appropriate alignment between the program and its targeted coherent market, conducive to a coordinated coevolution of the two. Changing market needs influence program technical efforts which in turn result in changing technical capabilities that influence the direction of further market change. The motivation is to create a sustained ability to build products that the market will need over time.

With basic software product engineering, program management initiates and oversees one or more development projects. Projects are independently managed but may be related in some way, such as building similar products for different customers or building different products for the same customer. With DsE, program management institutes a unified business, focusing on a coherent market of customers needing similar products. Through its domain engineering effort, the program establishes technical consistency across its product manufacturing projects.

DsE program management activities, including the efforts of domain engineering as its technical agent, are financed as capital investments in the domain business. There are several possible sources of investment funds, including:

- Enterprise direct funding for the initiation or sustainment of a chartered program
- A portion of domain engineering costs allocated to each manufacturing project based on its product revenues or profits
- Materials developed for another domain or by a product manufacturing project as custom product elements and provided in-kind as a basis for enhancing domain capabilities
- Direct customer funding provided with the objective of timely availability of needed domain capability enhancements
- Participation in targeted market or industry/government collaborations that provide funding or reduce otherwise incurred costs by means of shared resources or assets

Hardware Engineering

Software depends on hardware for interactions with the physical world. Within a DsE approach, all aspects of product behavior are first defined as software-based capabilities that can subsequently be realized or augmented in hardware form. A specification of envisioned hardware behavior is derived to guide the acquisition or development of a suitable hardware device. Access to all associated devices will be software-encapsulated to provide augmentation of and access to physical device capabilities. Some devices, specified in the product environment model as accessible but externally provided, will similarly be software encapsulated in the product for access to the services that the devices provide.

A limited degree of hardware expertise is needed in support of a program for selection of conventional, commercially available hardware components, such as in realizing a standardized product platform. When more specialized hardware elements are needed, hardware engineering is a proper aspect of software-based product development.

Specialized hardware elements, having been specified as entities in the product environment specification or as capabilities in the product requirements specification, are specified as hardware-encapsulating components in the product design specification. The expected behavior and physical realization of each hardware element is specified in the component specification in keeping with the practices in the relevant hardware engineering discipline. The component further specifies the interfaces through which the encapsulating software is able to interact with that hardware and interfaces through which other software components are able to access its services.

Systems Engineering

Systems engineering is integrated into domain engineering with a holistic software-first approach that defines the overall product organization. This includes specifying the platform and hardware components upon which the software-based product will depend. Hardware engineering efforts are chartered as a means to more efficiently or effectively realize software capabilities and quality objectives.

{Whole-Product Engineering and Manufacture; process customization in support of product customization; holistic -> uncertainty resolved at customer-system level}

Hardware-software codesign determines when custom devices are needed to enable or enhance software behavior. The initial premise is to define product behavior in software with general purpose hardware and then to conceive custom devices that enable or enhance more effective behavior:

- Build the product first in software, approximating and evaluating correct product behavior, using conventional or other existing hardware and emulated versions of inaccessible or specialized hardware;
- Derive hardware selection criteria and custom device specifications as results of the software-first realization (e.g., improve speed, precision, or predictability of computations; enable or expedite interactions with the physical environment);
- Select or build specified hardware as warranted for a product that will improve efficiency and effectiveness of the customer endeavor;
- Build and validate the deployable product, having modified hardware-encapsulating software to accommodate specified hardware.

Enhanced Customer Collaboration

A product is developed to provide capabilities that support a customer in performing their endeavors. The practices that an enterprise employs are influenced by the capabilities provided by such products. For best use of a product, there must be a proper fit between that product and customer practices. Provision of a new or modified product may create the need to modify customer practices, generally enabling the streamlining and reduction of effort associated with those practices. As a result, the impetus with DsE is not to view customer practices as a hard constraint but rather to approach the product development as an opportunity and means to improve those practices.

The Economic Basis and Metrics for Managing a DsE Program

DsE by focusing product efforts on a coherent market enables a program to standardize on the most effective solutions to a set of similar problems. This is based on an understanding of how customers' needs in that market are similar and how and why they differ. By rethinking the product development process—whether for a singular product or for each of a set of similar products, DsE enables significant improvements in productivity over that of a basic software product engineering approach.

The objective of a DsE program is to efficiently build products customized to each customer's specific needs at reduced time and cost. This objective is achieved by creating the means to mass produce customized products, following the industrial model of organizing development into customer-focused manufacturing efforts that leverage a market-focused engineering effort. Engineering subsumes routine development work into a shared infrastructure that reduces time, costs, uncertainties, and risks of manufacturing for each product as it is built and subsequently evolved over its useful life.

This benefits the program in several ways:

- Problem-solution competence is realized as a capital asset in the form of a domain;
- A standardized form and terminology is the basis for a shared understanding of customer and market needs;
- Improvements in the quality of multiple products can be achieved as a result of quality improvements in the product family;
- A standardized manufacturing process results in more predictable schedule and cost estimates;
- The time and effort to build each product is reduced due to a streamlined process based on leveraging existing assets in the form of a product family.

An initial concern for program management is to establish an economic rationale for determining the form and technical approach of a viable DsE-based program. A

continuing responsibility for all levels of management is having the means to measure and improve the (process and product) quality of its efforts. The purpose for this is to have an objective basis for setting achievable goals and knowing the degree to which those goals are being met. Goals are quantitatively characterized by metrics that identify the measures that indicate progress in meeting those goals. The purpose of metrics-based management is to make timely reality-based decisions regarding future plans.

Domain Investment and Technical Debt

Domain engineering subsumes product manufacturing concern for technical debt as a development issue. relation between domain evolution and technical debt: technical debt versus predictive investment in domain engineering: product family anticipation of potential change (effort on aspects of capability that may never be needed) versus an incomplete family (alternatives that have been envisioned as within the scope of a domain but are not yet supported or needed, giving only partial variability coverage and potential for only approximate fit to actual customer needs)

A DsE Metrics Strategy

The purpose of a metrics strategy is to guide management decision making with accurate and timely information. A metrics strategy for improving productivity builds on the roles of each of the three perspectives on process variation (defined in section 1.3) as context for management decision making:

- *Performance* – Monitoring work progress against a plan, to determine when and how to revise the plan (e.g., to defer less critical work or avoid future rework)
- *Maturity* – Comparing performance against potential productivity in order to characterize opportunities for improving performance (e.g., through additional training or mentoring or better definition of practices)
- *Capability* – Discovering opportunities for modifying the process to increase the potential for higher productivity or product quality (e.g., increased automation or investment in reusable assets)

Measurement Objectives

Just as the scope of management responsibility differs for the several levels of management for a DsE program, needed metrics differ accordingly. All levels are concerned with overall cost and schedule performance within their scope of responsibility and satisfaction of their customers with their results. Beyond these objectives, each level has more specific concerns:

- (Enterprise) Economic sustainability
- (Program) Enduring domain-market viability and alignment
- (Domain) Manufacturing productivity (product and process quality)
- (Product) Product fit for intended use over its useful life

Evaluating Economic Viability of DsE

An initial evaluation of the economic viability of a DsE-based approach should rely on an extrapolation from an organization's experience in building singular products. The objective is to determine whether the cost of developing and sustaining a domain and then deriving each of a set of similar products over a prescribed timeframe is sufficiently less than the aggregate cost of separately developing and sustaining the same set of products.

Such an analysis must be tailored to measures that reflect the organizational performance (actual or projected) of the enterprise in which a DsE-based program is being considered. The means for doing this is a notional model in which several variables must be adjusted to characterize a given program. Each of these is defined (and assigned an arbitrary value for purposes of illustration):

- (N_P) An estimate of the number of different products that will be built over the life of the program [= 5]
- (C_I) The average cost of building the initial release of a singular product conventionally [to be based on historic data; reduced for products 2..N_P to 0.75 * C_I of product 1 due to developer experience building product 1]

- (N_V) The projected average number of additional versions that will be needed over the useful life of each product [= 5]
- (C_V) The average cost, relative to C_I, allocated to develop a modified version of a released product [= 0.25 * C_I; to be based on historic data or a discretionary allocation]

Another set of variables are used to characterize the nature of a DsE program, each projected as a discretionary allocation that is a multiple of C_I. The program and domain engineering variables change inversely to the two product variables, each limited notionally to the ranges shown:

- (C_{DI}) The discretionary initial cost for establishing a DsE program in an enterprise [= 0.2 * C_I] {[0.1 -> 0.5]?}
- (C_{DE}) The discretionary cost of performing domain engineering over the life of a program [= 2.0 * C_I] {[0.5 -> 3.0]?}
- (C_{PI}) The cost of building the initial release of a product [= 0.25 * C_I] {[1.0 -> 0.1]?}
- (C_{PV}) The cost of building a modified version of a released product [= 0.25 * C_V (= 0.0625 * C_I)] {[1.0 -> 0.1]?}

With these two sets of variables, the costs for building a specified set of products with a DsE approach can be compared with the cost of conventionally developing those products. Based on the stated sample values for conventional development, total program cost for 5 products, each with 5 versions, would be

C_I * ((1 + 4 * 0.75) + (5 * 5 * 0.25)) = C_I * 10.25.

For the conventional development sample values and the DsE variables' hypothetical values, total program cost would be

C_I * (0.2 + 2.0 + (5 * 0.25) + (5 * 5 * 0.0625)) = C_I * 5.013.

The conclusions to be drawn from this comparison are anecdotal based on reasonable but arbitrarily chosen variable values. Based on these values for numbers of products/versions of 3 or more, the full lifecycle cost of a program, including domain engineering and product manufacturing, would be expected to be less than the cost of the corresponding cost of building the same set of products conventionally; as the number of products/versions increases, program lifecycle cost decreases to less than half relative to the cost of building the same products/versions conventionally.

An actual program would gain insight into its options by substituting its own representative values for conventional development variables and adjust the DsE variable values to reflect the amount of investment that the program chooses to make initially and over time. The latter portion of this model, for estimating the cost of a DsE approach, can also be used to compare levels of investment associated with alternative DsE technical approaches, by modifying the discretionary multipliers associated with each of those variables.

Observations

- For a program building a single product with no versions (a rare circumstance), a conventional development approach will suffice. However, if there are significant uncertainties (implying potential changes) involved in building the product, there is an argument for using DsE practices, in part to mitigate the need for late rework.
- The cost of developing a single software-based product conventionally is a lower bound on the cost of domain engineering. That cost reflects, in both the DsE and conventional approach, the capability of the development process being used and enterprise competence that determines current maturity in performance of that process. The upper bound is a function of the cohesiveness of the domain but should be expected to reflect reasonable diversity among a set of similar products.
- DsE process capability is greater than that of a conventional process (enables building and evolving multiple similar products, each at a fraction of singular

product cost including fractionally allocated cost of domain engineering). DsE leverages effort and expertise across similar products while being responsive to diversity among products and change over time in both customer and market needs.

- In weighing domain cost, during conventional development of a given product, problem-solution alternatives are also considered and explored but is viewed as a sunk cost, being discarded unless relevant to the current realization of that product.
- With a DsE process, once the cost of building a single product has been incurred, subsequent revisions of that product as well as the building of other similar products will incur an additional incremental cost. The DsE lifecycle cost of a single product (excluding its attributable domain engineering support costs) is expected to be significantly less than the conventional lifecycle cost for the same product. The actual cost for a particular product depends on the maturity of the domain and the degree to which the product fits within the scope of the domain.
- Similar well-understood portions of products can be built using a DsE approach even if more diverse or poorly understood portions are built using conventional practices within a DsE-defined framework.
- A DsE approach reduces the cost of product manufacture for each of 1-N similar products, vis-a-vis the cost of singular product development, by eliminating redundant efforts across projects and focusing projects on resolving essential differences among instances of the product family.
- The cut-off point at which a DsE approach would be favored is dependent on multiple factors including not only the number of products and versions of each but also the level of investment in domain engineering, the effect of this investment on reducing the cost of product manufacturing projects, and the degree to which needed products/versions are similar. Reducing the cost of manufacturing (e.g., due to process streamlining and automation) allows the

producer to choose among increased profit, reduced customer cost, or increased investment in domain engineering.

- Regarding DsE-reduced product cost, DsE-realized products can be priced based on benefit-imputed value to the customer (for greater profit or for investment in improved domain capabilities) or the actual cost (presumably reduced but possibly including an imputed allocation of domain infrastructure cost) of development (to gain market competitive advantage).
- Statistical quality assurance (see for example Deming or CMMI maturity level 4) is most practically applicable in the context of a DsE program. It includes as its necessary basis consistent (standardized) practices across projects and controls for process capability and performance variations traceable to product differences, leveraging assurance efforts across projects, with process streamlining reducing instances of special causes of variation.
- Increased investment in domain engineering is motivated either to reduce the cost of product manufacturing projects (a more streamlined process and/or increased automation), to improve realizable product quality or capabilities, or to increase market alignment or scope. This investment is discretionary for a program based on realistic expectations concerning future market needs. Investment can be increased or decreased over time based on achieved results and changing business circumstances.