

3.4.1 Domain Management

The domain management model specifies the direction, organizing, resourcing, planning, and coordination of a domain engineering project in accordance with market-targeting business objectives specified by program management. Program management establishes a domain as its technical agent and, through its program performance element, performs continuous improvement in the quality of its work.

Domain management is a variant of singular-product project management (as described in section 2.2), differing in three aspects: (1) its “customer” is a set of projects that build customized products for customers in the targeted coherent market, (2) its “product” is a domain (i.e., a product family and associated means for deriving products), and (3) the product family encompasses complete software-based products as appropriate (i.e., based on competence to perform all aspects of integrated systems, software, and hardware engineering and manufacture).

The overarching objective of domain management is to direct the building of a domain that provides associated manufacturing projects with the capabilities and materials needed to build products that meet the needs of customers in the targeted market. The elements of domain management are governance, planning, and increment performance of an effective and efficient domain engineering process.

Domain Governance

Domain governance specifies the relationship of the domain to the program and related domains (if any). A domain is initiated by a program to create and coordinate technical capabilities that enables projects to build similar customized products for their customers.

Program management specifies the objectives, market, and initial resources for the domain. Domain governance coordinates domain engineering work with projects and reports on progress to program management. Domain objectives focus on the current and future needs of the targeted coherent market as a whole. Program management provides direction on conformance to enterprise- and program-level policies and

procedures and use of associated services, including marketing, financial, facilities, personnel, and technology capabilities that support domain efforts.

Domain governance specifies the process for performance of domain engineering and quality criteria by which its productivity and domain quality are measured and improved. (The introduction to this chapter describes an assumed process for domain engineering in overview. Its activities are described in further detail in this and subsequent sections of this chapter.) Quality criteria for domain engineering productivity is an elaboration of the four categories of developmental quality as defined for software engineering (in section 2.2): feasibility, sustainability, conformability, and verifiability. Domain quality is evaluated in terms of the productivity and product quality that can be achieved by projects in building products.

Domain Planning

Domain planning specifies a domain plan having two aspects: the nature of the relationship between the domain and dependent projects and a master plan for realizing the domain over its expected useful life.

The program performance element of project management specifies the relationship of the domain to manufacturing projects: as a shared technical resource, as a collaborative collective within which projects operate, or as the technical authority over projects.

The domain master plan specifies building the capabilities of the envisioned domain as an evolving series of increments.

Increment Performance

Increment performance specifies the performance, in accordance with the domain engineering process, of a domain increment as specified in the domain master plan.

Increment performance entails elaborating, resourcing, and directing performance of the assigned increment plan to produce a consistent domain realization that satisfies specified quality criteria. This includes tasking to improve and extend the domain as projects build products and identify changing and emerging customer and market needs.

DRAFT