

appendix A

Small Product Development and the SDF

Ulrich and Eppinger define a Product Development Process (PDP) that focuses on small product development. There are two significant differences between the PDP and the SDF. First, for simplicity, the PDP avoids a multitiered hierarchy. Second, the rework cycle is not explicitly integrated into the PDP. These are some of the key differences between small product development and complex system development. It is not necessary to encumber a small-scale development task with a rigorous hierarchy of sub-systems and components since it would take more energy to manage those tasks than to perform them. Also, since rework does not ripple through a complex hierarchy, its adverse impacts are not as great.

I. Mapping in the Logical Domain

Ulrich and Eppinger's Front-End Development Process is shown in [Figure A1](#). As [Figure A2](#) indicates below, its key activities map directly from the Logical Domain view of the SDF in the categories of Requirements Development, Synthesis, and Trades. The "Refine Specifications" activity is handled via feedback loops in the SDF. The "Plan Remaining Development Projects" activity would be considered a management activity in the SDF; elements of this task would also be covered in the Synthesis activity.

Logical Domain mapping of the PDP to the SDF is straightforward. The organizing concept, discussed in Chapter 3 above, readily applies to the PDP.

II. Mapping In the Time Domain

[Figure A3](#) depicts Ulrich and Eppinger's Product Development Process. It maps directly to the Time Domain view of the SDF, as shown in [Figure A4](#). Phases 1 to 4 map directly to the full life cycle view of the SDF illustrated earlier in Chapter 3.

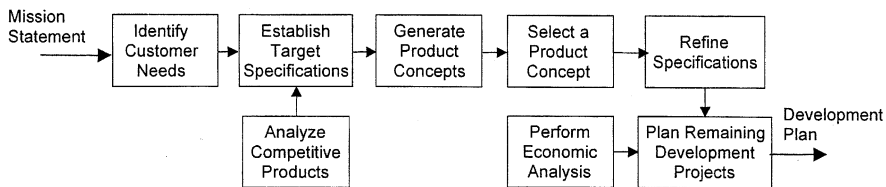


Figure A1 Ulrich and Eppinger's Front-End Process.⁶⁷ (Courtesy McGraw Hill, used with permission.)

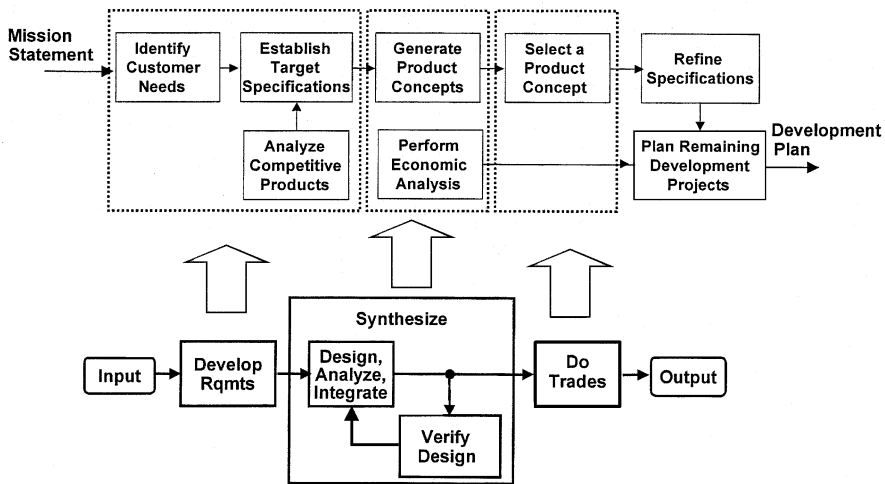


Figure A2 Mapping PDP to SDF in Logical Domain.

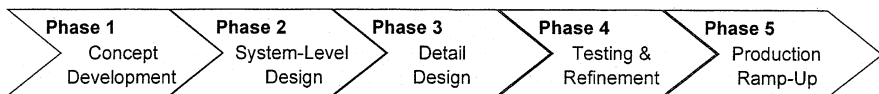
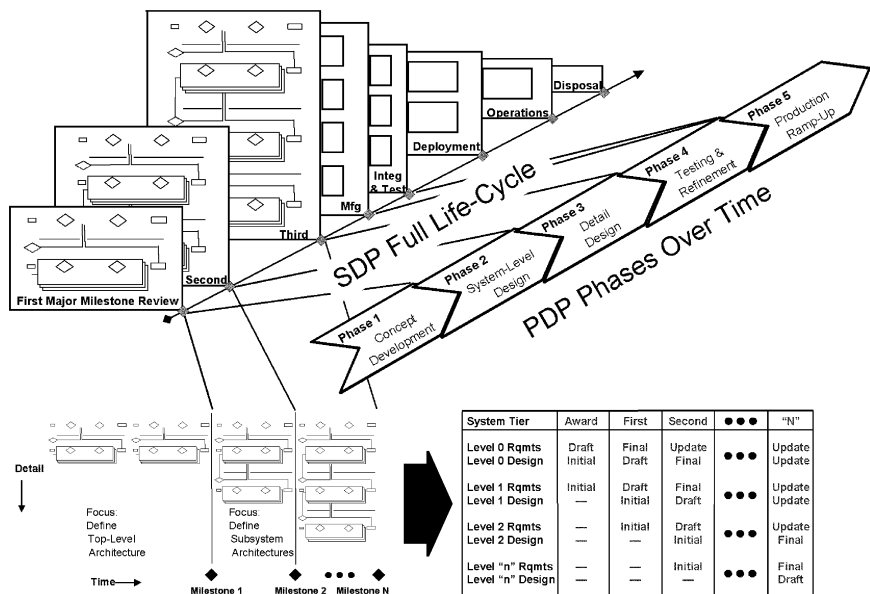


Figure A3 Ulrich and Eppinger's Product Development Process (PDP).⁶⁸ (Courtesy McGraw Hill, used with permission.)

Ulrich and Eppinger's small-scale development process represents a solid distillation of the SDF that focuses on the needs of simple systems. Multiple tiers are avoided and the rework cycle is not explicitly included. This brief discussion illustrates the utility of defining the Design Development Process in terms of both the Time and Logical Domains in order to preserve universality of application. This general rule applies to both simple and complex design development contexts.

⁶⁷ Adapted from Ulrich, Karl T. and Steven D. Eppinger, *Product Design and Development*, New York: McGraw-Hill, 1995, p. 18.

⁶⁸ Adapted from Ulrich and Eppinger (1995), p. 9.



Over time options focus down to unity while level of detail increases

I/O Evolution Over Time

Figure A4 Mapping PDP to SDF in Time Domain.