Overview of Product Development Processes

Version: 1.00
Updated: 2004-09-23

F.A. Salustri, Ryerson University, salustri@ryerson.ca
D. Proulx, University of Sherbrooke, dproulx@gme.usherb.ca

Preface

This module summarizes the product development process (PDP), which is the overall process by which products come into existence, and within which engineering design occurs. The component tasks of PDPs are briefly introduced, from the perspective of engineering design. It is intended to provide an overview of the PDP that might be useful to students at the beginning of an introductory course in design.

Keywords: product development, design engineering process.

The target audience of this module includes students taking *introductory courses in design*, and students interested in a summary overview of the PDP. It also provides a useful reference for module authors, in order to provide a degree of consistency across different modules.

The objectives of the module are (a) to provide a foundation for a common PDP to be used in all CDEN modules, and (b) to ensure a common information base for students.

The context of the module is that of introductory design engineering in years 1 and 2 of undergraduate study. Students have relatively little understanding of the engineering sciences at this point in their studies, so the PDP must be presented in a manner not requiring much analytic expertise.

Table of Contents

1. Background

1.1. Some General Considerations

A *product development process* (PDP) is the overall process that occurs in industry when a product is engineered and brought to market. It is not simply an academic method that is followed, as from a textbook, that guarantees an acceptable result. A PDP is the series of phases that are commonly executed each time a new product is developed or an existing product is modified. Engineering design is one aspect of a PDP; some other aspects are manufacturing, business, management, research & development, industrial design, quality assurance, and innovation. However, of all the aspects of PDPs, engineering design is the most fundamental. Design is the foundation of the technical side of product development. It also impacts directly upon all the other aspects of product development. No other aspect can boast such breadth of scope and depth of impact. (This is not to minimize the importance of the other aspects – a product developed without due diligence paid to *all* aspects of its developed will very likely fail either on technical or on other merits.)

A characteristic of engineered products is that they become parts of the environment and thus impact on the quality of life of a population in ways that engineers cannot generally predict. It is therefore incumbent on the designer of a product to account for the potential impacts a product can have on individual users of the product, on society, on humanity, and on the environment.

The users of a product are rarely the clients for which the product is being designed. When a client gives a designer a design task, an important question the designer must ask is: What does the *user* want and need this product to do? The client may have some sense of this, but the designer's expertise will likely direct him to explore other issues pertaining to function, form, aesthetics, characteristics such as size and weight, user interface, and cost. Furthermore, there are cultural issues to be considered especially if the product is to be marketed globally.

All these matters must be balanced against the needs of the client: will bringing the product to market be wise for the client? Will it make financial, marketing, legal, and corporate sense?

It is not necessary that every phase be performed for every product development project. There is no commitment here to a purely sequential execution of these phases; depending on context, at least some of these phases can overlap or even be carried out concurrently. Indeed, concurrent execution of as many phases of the PDP as possible is advised in the general case. However, the sequence represented here is typical of the general ordering of phases and depends primarily on the amount of knowledge available to designers about a particular project. Early phases are needed to develop knowledge typically needed in later phases. There is no single PDP; differences from one PDP to another depend on contextual information such as industrial sector, market structure, corporate culture, and regulatory restrictions. Mentioning *the* PDP is only intended to refer only to the particular PDP in this document, which is a reference document for CDEN. No commitment is made or should be implied regarding the universality of this PDP.

This is not intended to be a full and detailed description of the PDP. It is a general outline meant to ground further explication in other modules.

1.2. Defining "Designing"

Based on this foundational role of design in engineering, we introduce the following definition of the process of designing.

> *Engineering designing* is the synthesis of balanced, implementable artefacts that resolve poorly understood problems, such that their use in a given context will promote a preferred situation.

This definition is based on the CEAB's definition of design, the definition given in the announcement for the NSERC Design Engineering Chairs program, and aspects of various other definitions from the literature, but it does require some commentary.

Synthesis. Synthesis involves the bringing together of things to make something new and greater. Synthesis is the "creative" part of designing, but it is not just creativity. Creativity is unconstrained and extremely unpredictable. Synthesis is somewhat more structured and a little more predictable. That is, synthesis can be thought of a *creatively rational* thinking, or conversely, as *tempered creativity*. There are many methods to stimulate tempered creativity in design engineers.

Balance. A balanced design is one that finds the ideal trade-off between all the major drivers of the design. Balanced designs are relatively easy to identify in hindsight: the original Apple Macintosh, the Studebaker, the original VW Beetle, the Boeing 747, the Aeron Chair, the Apple iPod. None of these products were of particularly high quality, or low cost, or especially light or functional; but all of them found a balance that was ideally suited to the context, times, and environment into which they were introduced.

Implementable. A design is in many ways a plan for the manufacture or, more generally, an implementation. If a design cannot be realised, then what is the point in designing it? It is important to note, however, that design engineers often lack the expertise to ensure that their designs are implementable. It is important, therefore, to keep manufacturing experts and other implementation specialists intimately involved in the design process.

Artefact. Literally, something "made by hand"; something man-made (as opposed to something that occurs naturally).

Poorly understood problem. While it is usually easy to specify a design problem superficially (e.g. a new commuter train is needed between Union Station and Pearson International Airport), it can be extremely difficult to gain an understanding of the problem that is deep and broad enough, that designers can design a solution easily. What's more, customers/clients/users rarely understand the problem themselves. It is through consultation with clients and users that designers finally gain a sufficient understanding of the problem, to start designing a solution.

Context. The context of a design is more than just the environment into which the designed product will be introduced. It includes the companies involved in the product's development, the people who will work on the project, government agencies and regulatory bodies, economic factors, legal and ethical factors, and other circumstances. It is virtually impossible to track every possible contextual factor in a design problem, but there is tremendous corporate knowledge available, depending on the product class and industry, about which factors are the most crucial.

Promote. There are few guarantees in design engineering, because design is partly about predicting the future. Companies and managers who insist on success are only setting themselves up for failure and will build companies that are *brittle* – that cannot handle product or engineering failures. Instead, successful companies are adaptable to failure; they realise the best they can do is work generally towards what they hope are better situations.

Preferred Situation. We can never reach the 100% perfect solution, but it is important to have such ideals. With ideals, one knows "where to aim" to achieve success. So long as new products move an environment towards the ideal we have improved the general quality of life, and that is the best we can expect.

1.3. The Systems Perspective

In this and other related modules, a *systems perspective* will usually be adopted. We explain this perspective here.

> A *system* is a set of interacting elements that provide identifiable functions, and that is crisply distinguishable from its environment.

So an automobile is a system, but so is its engine. People are systems too. A paperclip is not a system (because it is not made up of distinct parts), but a paperclip holding two sheets of paper is a system.

Systems (products) are *transformers* – they convert inputs to outputs. These inputs and outputs come from and go to the operating environment of the system. This is shown in Figure 1.

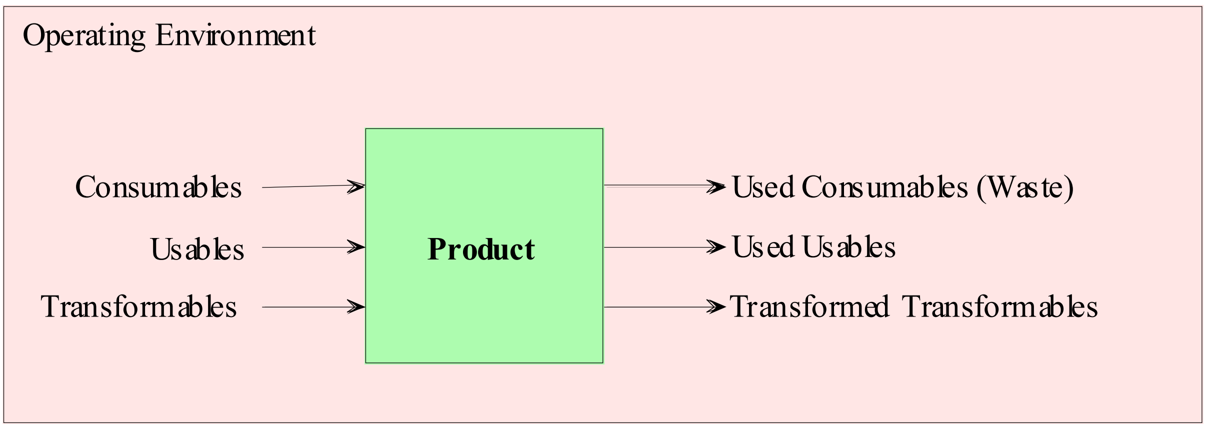

Figure 1: The systems perspective.

There are generally three kinds of inputs to systems. *Transformables* are the inputs that are the object of the principal functional transformation of the system. People are the transformables of automobiles, elevators, and banks; a "blank" workpiece is a transformable in a manufacturing system. *Consumables* are inputs that are destroyed during the transformation process. Gasoline and air are consumables in an automobile; electricity is a consumable of a blender. *Usables* are inputs that are used to effect a transformation, but are not destroyed – that is, they can be reused. Tools and machines are usables on a factory floor. Your pen is a usable when taking notes in class (though the ink and paper are consumables).

The product/system changes things in the environment leading to the *preferred situation* mentioned in Section 1.2. An automobile's engine transforms fuel and air into mechanical power. The car itself transforms passengers from one location to another. A blender transforms fruit and ice cream into a smoothie.

Every product, however, produces undesirable outputs as well as the desired ones. The designer must evaluate how bad the effects of the undesirable outputs are, and make sure that the product follows health and safety regulations as well as environmental concerns. Most importantly, the introduction of a product changes the environment – and might make the environment unsuitable for the product, limiting the product's life.

Furthermore, there may be *undesired inputs* to a product/system. When designing a personal digital assistant, undesirable inputs include liquids spilled on the PDA, software viruses downloaded into the PDA, the energy due to impacts when the PDA is dropped, and many others. The designer must think of as many possible undesirable inputs as possible, assess which must be withstood by the product and design a suitable solution.

Not only does the systems perspective apply in thinking about products; it also applies to manufacturing of products. In this case, the "factory" is the system in an environment; raw material is one system input, and the manufactured product is one of the outputs. This emphasises the impact of products on the environment, and how a product that was considered "good" when it was first introduced can literally make itself obsolete. Taking a systems perspective helps designers keep all these factors in mind.

2. Product Development Processes

2.1. Stages and Gates

A general PDP is shown in Figure 2.

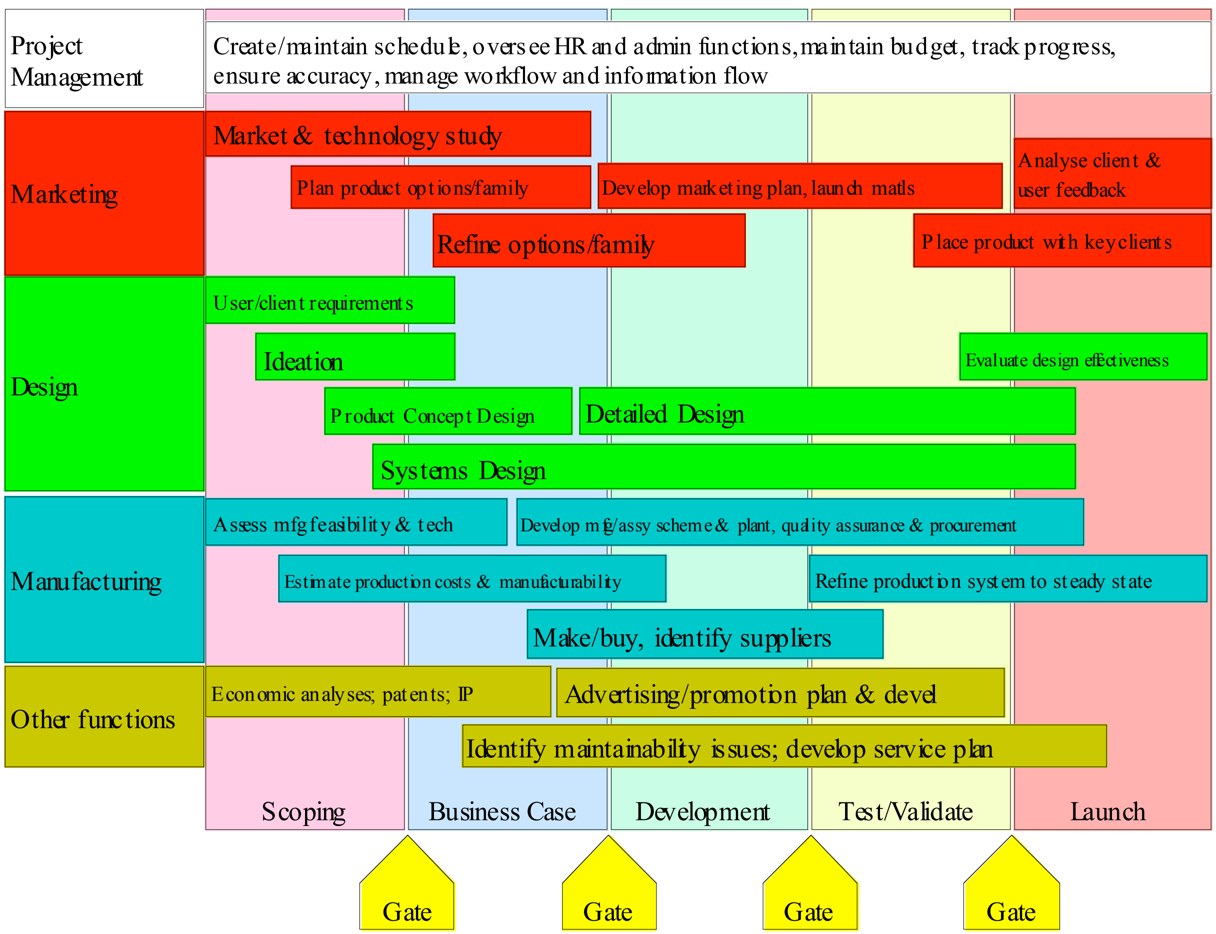

Figure 2: A generic product development process showing the five major stages and the roles played by different disciplines in each stage.

The stages and gates indicated on the horizontal axis are taken from the STAGE-GATE method (trademark in Canada by The Product Development Institute, Inc. – http://www.prod-dev.com). We note here that while we refer to the stages and gates as defined by STAGE-GATE, we will **not** discuss the details of STAGE-GATE. Instead, we will consider the interaction between stages and gates in general, and the phases of engineering a product. The interested reader is referred to

Horizontally, the figure identifies five *stages* of product development. (One should imagine time on the horizontal axis.) Each stage is separated from the next by a *gate*. Gates are checkpoints at which the status of a product is reviewed carefully; only projects that meet the requirements of a gate are allowed to proceed to the next stage.

Vertically, down the left side of the figure, the principal *sectors* of product development are listed. Each sector covers an area of expertise, and is involved with a variety of tasks throughout the PDP. Some of the typical tasks undertaken by each sector are given, running across the five stages.

The boundaries between the tasks of product development are not crisp. It is sometimes very difficult to tell why one task ends *here* and not *there*. There is no simple answer. Factors influencing where boundaries are drawn include corporate culture, design methods used, and technological limitations. Furthermore, the boundaries themselves are quite fuzzy, shifting even from one project to the next within a single company, in response to the experiences of the design team, technological limitations, contractual obligations, regulatory limits, and other factors. All this notwithstanding, Figure 2 remains a reasonable approximation of how many engineering companies work.

The five stages are briefly described below.

Scoping. Scoping is typically a stage carried out by management and marketing specialists, with significant input from engineering. In this stage, opportunities are sought to develop improved versions of existing products or to introduce new products. Sometimes, clients will announce their need for new products (e.g. NASA announces requests for proposals for a transport system to replace the Space Shuttle). Other times, marketing divisions may identify new customer needs internally (e.g. Ford decides to redesign the 2005 Mustang based largely on the 40-year-old original). The scoping stage involves considering various potential projects, with respect to the expected revenue and expenditures, technical needs (e.g. does the company have to hire new specialists?), scheduling with respect to other ongoing projects, etc. The design engineer is responsible for providing technical expertise to help evaluate the likelihood that the company can compete successfully, should they choose to pursue the project. Sometimes, the scoping exercise is also used to let design engineers establish a baseline of competing products against which they must compete, and to brainstorm some preliminary design ideas – giving them a jump start on other design activities that might occur later in the process. Management them decides, on the merit of each proposal, whether the company should pursue it.

Business Case. Having established sometimes several potential projects, the second stage treats them from the business point of view. In this case, the financial needs and expectations, including requirements for new machinery, human resources, etc, are analysed. Again, design engineers play an important role in identifying the kinds of expertise and equipment, and other resources that would likely be needed. To do this, product concepts must be fleshed out further. Sales personnel estimate the rate at which the new product might be sold, which defines production requirements. From this, manufacturing engineers can estimate the costs associated with actually fabricating the product. Business and management specialists, accountants, and costing advisors then estimate in greater detail the costs and expected revenues of pursuing the project. If a sound business case can be made, the project passes the second gate.

Development. In this stage, most of the product development is actually done. This includes system and subsystem design, and detail design of parts. As the actual form of the product is established, the concept may change, sometimes significantly. These changes, however, are always driven by the need to satisfy requirements established during the first two stages. At the same time, manufacturing engineers refine their plans for fabricating the product and getting production up to full speed. As the design matures, other corporate groups, such as Human Resource departments, begin adjusting the employee base to get ready for production. New facilities are constructed and equipment purchased. As the product design matures, some computer-based tests can be started. Since the design is still immature, the results of such tests are relatively unreliable. However, computer-based testing costs very little compared to the costs associated with having to correct design flaws later in the development process. Computer-based tests require no special test rigs, specimens, or other expensive items that are necessarily used only once per test.

Test/Validate. Once the design is mature enough that actual part dimensions, tolerances, and material selections are available, a more detailed validation of the design can be carried out through more detailed (and time-consuming) computer-based tests, as well as building prototypes and actually testing the product physically. For example, in the aircraft business, it is typical that as many as three whole aircraft are built only to be tested until they break – for the sake of proving that the design will behave as the engineers have predicted. As test results become available, the engineers refine the design to improve its performance, quality, and cost wherever possible. At the same time, refinements are made to the manufacturing facilities to account for any problems encountered during the construction of the prototypes.

In some industries (such as software engineering and electronics engineering), prototypes are referred to as *alpha versions*. They are given (or sold at reduced prices) to trusted customers and users for further field testing – when they are used in real settings, which are always different from laboratory settings. Based on the results of these early field tests, the design (and associated production processes and facilities) may be further refined.

It is unlikely that a product development project will be terminated at this stage, because the company has already invested a great deal of money in the project, and the development team has had many opportunities to fix and problems that were identified. However, other factors may come into play. Market conditions may change; new government regulations may take effect; users may not respond as expected to the product. A classic case of this last item was the Apple *Lisa* computer, which failed in the market because it was so powerful and different that users just did not know what to do with it. Apple then "dumbed it down" to create the *Macintosh*, which almost immediately set a new gold standard in computing.

If a product passes the gate at the end of this stage, it is finally ready for the last stage.

Launch. The final stage of a PDP is the product launch. The product is formally introduced on the market; advertising plays a key role here in bringing it to the attention of potential customers (unless it was specifically commissioned, as is typically done, for example, in space exploration and development). In this stage the manufacturing facilities are slowly brought up to full capacity. The facilities are carefully monitored during ramp-up, to ensure that production at the required rate is maintained without adversely affecting product quality. Any problems in the manufacturing and assembly of the product must be addressed quickly, or long-term product quality can suffer, leading to customer dissatisfaction and possible safety risks.

2.2. The Role of the Disciplines in a PDP

Referring back to Figure 2, we note that there are five disciplinary groups that constitute rows in the Figure: project management, marketing, design, manufacturing, and "other functions". Our focus is on design, but it is very important to note that expertise from each discipline is needed in virtually every stage of a PDP. It follows, then, that a company must not compartmentalise its expertise according to the stages, but rather the company should have experts from all the disciplines *present in every stage*. It makes sense to keep the same experts involved in the same project for as long as possible – preferably the whole project – because these people will come to know more about the project than other personnel who have equivalent technical skills but simply lack the historical understanding of the project. This is one of the main reasons why product development – and design engineering – are inherently team-based activities.

2.3. Some Variations on the Generic PDP

No two PDPs are the same. A major characteristic that distinguishes different PDPs is a company's *basic business driver*. Each kind of driver has requirements that force changes in the PDP. Some common variations on the generic PDP in Figure 2 are shown in Figure 3, below.

	Market Driven	**Technology Driven**	**Platform Driven**	**Process Driven**	**Customer Driven**
Description	Begin with market opportunity, the find appropriate technology	Begin with a new technology, then find appropriate market	Assume new product built around existing product's technologies	Product characteristics highly constrained by production processes	New products are slight variations of existing configurations.
Distinguishing Features	None – this is the generic PDP	Additional initial activity matching technology to market	Concept development assumes a technology platform	Both product & process developed together from outset, or an existing process must be specified at the outset	Similarity of projects allows for highly structured development process
Examples	Sporting goods, furniture, tools	Gore-Tex rainwear, Tyvek envelopes	Consumer electronics, computers, printers	Snack foods, cereal, chemicals, semiconductors	Switches, motors, batteries, containers

Figure 3: Some common variations on a generic product development process (based on [2]).

We see from Figure 3 that the generic PDP in Figure 2 is a *market driven* process. The variations shown subsequently in Figure 3 are all based on differences in the context of the PDP, i.e. the facts, assumptions, and constraints pulled into the PDP from external sources. For example, a *technology-driven* PDP constrains the process to work with a specific technology – provided from "beyond" the PDP – whereas the *market-driven* PDP is constrained to treat specifically one market opportunity. Although these variations require the actual PDP to be adapted, there are still vast similarities.

One common problem that afflicts companies seeking to broaden their scope is the need to adapt their PDP. When a company attempts to enter a new market area that is driven by different factors than the one to which they are accustomed, their PDP must be adjusted to account for the new market area's drivers. Many companies do not even realize that the new market has a different business driver. They try to apply their own PDP without change, but the PDP is not suited to the new driver. The result, invariably, include products unable to compete in the market, internal strife, loss of morale, and occasionally severe financial difficulties.

The way products are developed (the PDP) is strongly coupled to the structure and culture of a company. Both the PDP and the organization must be able to adapt to external forces while remaining internally compatible.

2.4. Major Aspects of PDPs

2.4.1. Concurrent Engineering

Originally, products were developed in a strictly sequential. As products became more complex, and the expected develop time shortened and the expected cost of products dropped in a highly competitive global market, this sequential approach became problematic. It incurred long lead-times, poor quality, and unmanageable cost and complexity (compared to expectations).

Concurrent engineering is an approach that has been successfully implemented in every industrial sector and for every kind of product, leading to more cost-effective products of higher quality that are developed much more rapidly than can be done with other known approaches. Concurrent engineering is reflected in the generic PDP.

Concurrent engineering is based on the observation that there is no reason to access engineering expertise sequentially, even if the stages of product development occur in a (roughly) sequential manner. That is, manufacturing expertise should play an important role in *all* the phases of product development, not just in the manufacturing phase. The same can be said of all the other kinds of expertise needed in a PDP.

2.4.2. Teamwork and Collaboration

As is made evident in Sections 2.2 and 2.3 many *stakeholders* are involved in any given product development project. These stakeholders all have a *stake* in the project (i.e. they stand to gain or lose by the product's success or failure), and so must work together to complete the project successfully. However, it can be extremely difficult for these partners in product development to work together.

A typical structure of a design team is presented in Figure 4. This is the sort of team one might expect to need for the design and manufacture of a product of modest complexity contracted by an external client.

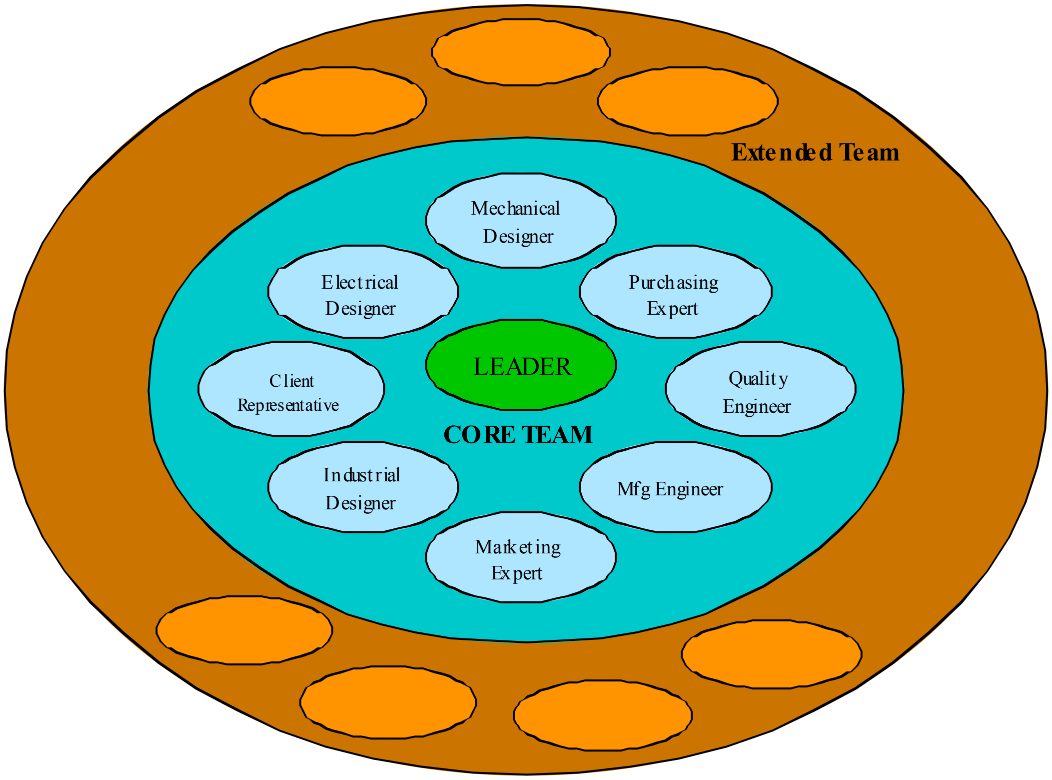

Figure 4: A generic team composition structure for electromechanical products of moderate complexity (based on [2]).

Individuals with specific kinds of expertise are in the inner circle; this is the Core Team. The Core Team should follow the project from beginning to end. The Extended Team includes those people who (a) might come and go from a project over its lifetime, and (b) are usually working for or with members of the Core Team. The light brown ovals in the Extended Team circle denote other specialties that are needed, but only occasionally, by the Core Team. Sales, Finance, Standards, and Legal experts are usually the kinds of people that occupy positions on the Extended Team.

NOTE: the *client*, and the *user* (of the product being designed) are not necessarily the same. It is important to remember that the client may not actually know as much about a product's users as the client may think. While an engineer has an obligation to work diligently and ethically for one's client, the engineer must also work for the *public good*, which means that design engineers must balance the needs of clients against the needs of the users of the products.

For very small design projects, there might not be need for a Team at all. If one person can be found with the experience needed to design the (small) product, then the Core Team collapses to a single person. The Extended Team, however, usually remains in existence, as a group of consultants to the designer.

For very large design projects, added rings may be added. In this case, the Core Team consists of systems specialists, each charged with a particular major system or component/assembly of the product. For example, in aerospace design of large aircraft, the core team would typically include a Project Leader, plus a Group Leader for each of the wing, fuselage, tail, cockpit, engines, etc. Each of these leaders would lead a sub-team structured as in Figure 4.

2.4.3. Project Management

Project management is a fundamental aspect of product development. It is where engineering and business overlap, and where the expertise of the engineer is exactly as valuable as that of the business executive. Product development occurs in a business setting, and must satisfy certain business needs as well as consumer/societal needs. Developing products costs money, which must be recouped via sales if a company is to remain in business. In sufficiently large enterprises, there may be multiple levels of project managers ranging from team leaders to senior project directors responsible for all new product development.

A number of factors impact how project managers must operate:

Scheduling and Work Efficiency. Project managers must manage the teams, ensuring that work schedules are reasonable and met by workers. They must be able to predict the achievement of milestones and completion of project stages, while simultaneously helping engineers overcome technical hurdles by securing appropriate resources (computers, supplies, expertise, documentation, etc).

Maintaining Product Development History. Traceability of engineering work is essential for (a) justifying the quality of the *design* (not necessarily the quality of the product), and (b) providing feedback to upper management as well as when design flaws are identified. Project managers must ensure documentation is up to date and accurate. They also must ensure that any necessary standards (e.g. ISO 9000) are being followed.

Project Championship. The project manager is often in charge of *defending* a particular project to upper management, especially at the early stages of development; this is often called *championing* a project. This means the project manager must continually justify the project as worth pursuing, in the face of other potential projects that a company might undertake. The project manager needs accurate, reliable information from his engineering team to champion a project successfully. This also means that the project manager accepts the *responsibility* for the project's success. If the project fails, upper management will blame the project manager.

Implementing Corporate Strategies. Upper management will set goals and identify strategies – for example, adopting *lean manufacturing*. It will fall upon the project manager to implement those strategies. This will require convincing workers that these strategies are worthwhile, and supporting the activities needed to implement new strategies while letting workers continue with their regular work.

Facilitating Teamwork and Collaboration. Sometimes, teams have members who do not work particularly well together. The project manager is responsible for managing these situations, identifying interpersonal problems, and resolving them. Rarely is changing the team's membership an option – instead, the project manager may have to exercise a variety of "soft skills" and techniques to resolve dysfunction in teams.

2.4.4. Technical Communications

In any environment with multiple stakeholders, it is vital that all the stakeholders have all the information they need, and *only* the information they need, exactly when they need it, in a form they can understand easily. This means that the stakeholders must be able to communicate effectively and efficiently. In order to do this, design agents need to have the *skills* to communicate orally and in writing so that all other stakeholders can understand. They also need the *tools* to make sure their design is properly documented and that communications channels are efficient and timely.

Also, it is impossible to *prove* the validity of a design in the conventional, mathematical sense. One must then justify a design based on rational, logical arguments. These arguments are presented in reports and presentations.

In written reports, spelling and grammar can be improved significantly through the use of appropriate software typically built into word processing software (e.g. the "technical writing" option in Microsoft Word's grammar checker).

Composition, however, is an element of technical writing for which there is no computerised support. Good composition means a report is clearly laid out, that its structure promotes straightforward comprehension, that the choice of words leaves little room for misinterpretation. A well-composed report is such that the reader feels like he is being taken on a well-organised tour, where report chapters, sections, etc. fit "naturally" together, and where the report's conclusion seems unavoidable given the arguments in its support.

Oral presentations are significantly different from written reports for three main reasons. First is the use of *voice*; a speaker can use inflection and emphasis in different ways than a writer. Proper inflection and emphasis – including the judicious use of hand gestures and facial expressions can augment the actual words spoken. The result is a shorter and crisper (and thus more compelling) presentation. Second, presentations are *two-way communications* venues. One's audience may ask questions and make comments that can lead the speaker to emphasise various points. A speaker should be prepared for questions by having considered how the audience might interpret the presentation. Also, the audience's *body language* can tell a speaker much of what the audience is really thinking. With practice, one can become sensitive to these inputs, and alter one's presentation to engage the audience more fully. Third is the use of *presentation materials*. Whether done with props, powerpoint presentations, or overheads, visual materials can both *summarise* and *emphasise* key points in a presentation. This frees the speaker from doing so vocally and frees more presentation time to make compelling arguments. Presentation materials can also be used as *prompts* for a speaker, liberating one from the use of scripts and cue cards, and from rote memorisation of presentations.

2.4.5. Usability and User-Centred Design

The design of a product should always be centred on the needs and abilities of the product's intended *user* community, even if one's client is in fact not a user.

Some people think that products can be divided into two major groups: products used by humans (e.g. cars, dishwashers, VCRs – so-called *consumer goods*), and products used only by other products (e.g. engines and motors, structural frames of buildings, pipelines). This is a false distinction, however. Sooner or later, humans use everything. Virtually every product of the second sort has to be installed, maintained (and repaired when maintenance is not enough), and eventually de-installed and disposed of. Human intervention is required for all of these tasks. So even for the most "industrial" products, usability is fundamental – even if it is just from the point of view of installation, maintenance, repair, or disposal.

Consider the automobile. Its driver is clearly a user of the *car*. However, is the driver also a user of the *engine* of the car? In fact, the driver is *not* a (direct) user of the engine; the driver uses the instruments and controls in the "cockpit", but he does not use the engine itself. A *mechanic*, on the other hand, may be thought of as the direct user of the engine. While a mechanic might not be a user in the conventional sense, he must still interact with the engine to attain a goal and satisfy his customer. The more difficult it is for a mechanic to work on an engine, the more money will be charged to the car's owner for the work, and the lower the reliability of that work will be. A mechanic has a different purpose and level of expertise in "using" an engine, of course, and so designing an engine to be used by a mechanic is different than designing it to be used by a driver. It is up to the engine designers to recognise the crucial role of a mechanic in providing a reliable product, and to design engines that mechanics can use.

One would usually see this kind of issue referred to as a "design for maintainability" issue, but this perspective can ignore issues pertaining to the capacity of a human to actually do the maintenance. It is therefore valuable to consider *usability* issues any time a human interacts with a product, whether the human is the intended user (like the driver of the car) or not (like the mechanic).

2.4.6. Product Lifecycle Considerations

Any product will go through a lifecycle – a set of stages during which things will happen to it, and to which it will react in certain ways. Research has shown that a product that is very good in some of its lifecycle stages but bad in other stages will tend to do poorly in the long-term. Governments and society are also starting to expect more from products (e.g. environmental friendliness *throughout* its lifecycle), so product developers now consider all stages of a product's life when they are designing it.

Consider Figure 4, below.

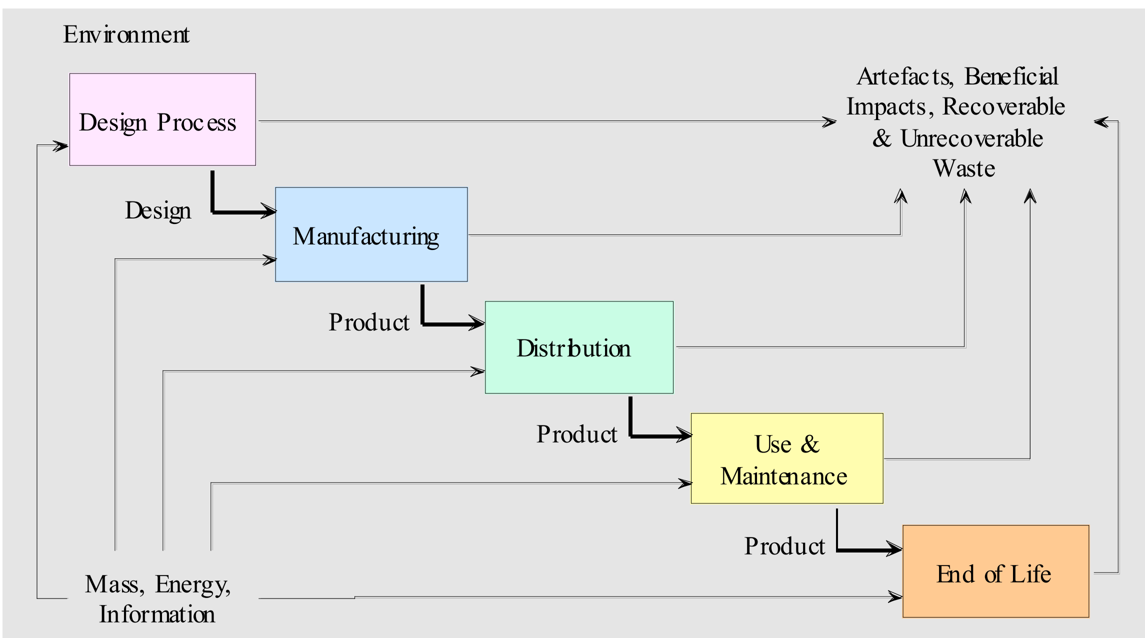

Figure 4: Basic relationships between lifecycle stages.

To develop a product, one must properly address all the stages of the product's life. A well-balanced product must trade-off a variety of issues that all impact the environment. Product developers must design products that address all the issues.

At every stage of a product's life cycle, material, energy, and information is consumed, generating (some) waste. Both consumption and waste adversely affect the environment (not only ecologically). Lowering the amount of consumed inputs and waste outputs is important to achieve product balance, thereby increasing the efficiency of the product as a part of the environment. This makes good financial, business, social, and technological sense.

2.4.7. End of Product Life

Another goal of engineering is to plan what will happen to a product at the end of its useful life, either by obsolescence or by its failing to meet new or different user needs. Planning for *end of product life* requires substantial effort in the initial design to ensure that the effect of the product's end of life is not detrimental. There are two common concerns when dealing with the end of a product's life: environmental impact, and cost. The difficulty is that these two concerns act against one other; lessening the environmental impact of a product usually involves an increase in cost. Lately there has been increased societal pressure to develop environmentally safe products. There are five basic ways to address the end of product life. These are introduced below.

Resource Reduction. By using less material to make a product, there is less physical waste at the end of its life. This principle applies not only to the product itself but also to many other aspects of its development, fabrication, and use. There are, of course, trade-offs. One might find a way to use less material to make a product, but only by producing more waste during manufacture – each case must be treated individually to assess the *total* impact on the environment.

Product Reuse. Reusing existing products rather than replacing them leads to savings of energy and resources needed to make new products. Preparing a used product for reuse requires only minor maintenance and cleaning. Reuse is particularly useful when a product reaches its end of life due to a lack of need by its users, rather than a degradation of its functionality (i.e. the product breaks). Thus, to improve a product's reuse, it should be designed to outlast its usefulness in its original target market. This has obvious implications on the cost of the product.

Product Remanufacture. If a product cannot be reused, then it could be remanufactured. Remanufacturing requires greater effort than reuse. Remanufacturing requires a product to be disassembled and its parts cleaned, repaired, or replaced, and then reassembled. While remanufacture costs more than reuse, it can cost less than manufacturing a new product. The viability of remanufacture as an end of life treatment for a particular product depends on the amount and cost of the needed resources. This can only be determined after inspection of the candidate products – inspections that can themselves incur substantial costs. These costs can be offset partly by the opportunity to upgrade the product during remanufacture, thus adding further value to the product than it had before remanufacture.

Product Recycling. Recycling is the most "extreme" form of treating a product at the end of its life. Recycling involves treating products as if they were just raw materials, to be processed into new products. Materials that are typically recycled include paper, glass, aluminum, steel, and plastic. Products to be recycled are generally destroyed completely by the recycling process, and the materials that compose them are then reused to manufacture new products. For example, paper recycling involves turning the paper back into pulp from which impurities (i.e. ink) are removed. The pulp is then formed into new paper stock. Recycling is also a viable process for the waste of manufacturing processes (such as the metal "chips" that result from many manufacturing processes like lathing, milling, etc).

Disposal. If no other means of using a product past its end of life can be justified on environmental and economic grounds, then the product must be disposed of. Waste disposal is a severe problem facing society today. Waste is generally divided into the following categories: liquid, solid, and radioactive. There are some very clever engineering solutions that have been implemented to handle waste, and then there are some that are not as clever but meet the needs of the user and of society. Sometimes the materials that are considered to be indestructible waste can also be handled in very useful ways. Different products can be disposed of in different ways, and engineering ingenuity can also be applied to solve this problem. An important area in modern design practice is finding ways to design products so that they do *not* have to be disposed of at their end of life.

2.4.8. Health and Safety

Health and Safety (HS) aspects of industrial products have become an important consideration for governments, courts, professional organisations, and the population in general. Most specialists in this field agree that the design engineers have a duty to assure that their designs are safe for all intended *and non-intended* users. However, designers need specific tools in order to integrate HS considerations efficiently into their design processes.

In a separate CDEN module, a detailed presentation is given of *why* and *how* HS considerations must be integrated into the PDP. A brief historical review of HS considerations in design engineering will familiarise the reader with the general concepts. An overview of laws, regulations, normalisation, and best practices will be used to lead to three important conclusions.

- Engineers have a professional obligation to deliver a product that is safe.
- Though there is much *theory* of how to handle HS, it is not easy to practice.
- If HS are considered constraints on a design, then there are *competitive advantages* in integrating them during the *early* stages of a PDP.

The Risk Analysis and Control Procedure (RACP) will introduce the reader to the five typical steps of risk management: define the scope of the analysis; identify hazards; estimate and evaluate the associated risks; find solutions to mitigate the risks; and finally evaluate the effectiveness of the solutions. For each step, guidelines, tools, or methods will be presented.

Finally, one must be able to assess *how* and *when* RACP should be included in a particular PDP. As it happens, the RACP can and should be applied at *every* phase of a PDP, giving more specific solutions at each phase.

An example of the application of the RACP to the design of a three-axis CNC moulding machine is also presented to show how the RACP actually works in a realistic setting.

2.5. Designing as Part of Product Development

Design, within the scope of a PDP, covers the technical aspects of product development only; that is, the methods and tools needed by designers to overcome the technical – not the management, business, financial, or societal – hurdles posed by a design problem. Every good design engineer should be very concerned with *all* the aspects of product development; but the kind of *engineering* expertise needed to develop a product is different from the kinds of expertise needed to handle the other aspects of product development. This section focuses on only that aspect of product development that deals with its purely technical aspects.

While designers should be present at every stage of product development – because their expertise is valuable at every stage – it is the early stages of a PDP where designers are most intensively put to work. Just as we coarsely outlined a general PDP process in Section 2, we can now describe a general design process. The reader should refer to Figure 5.

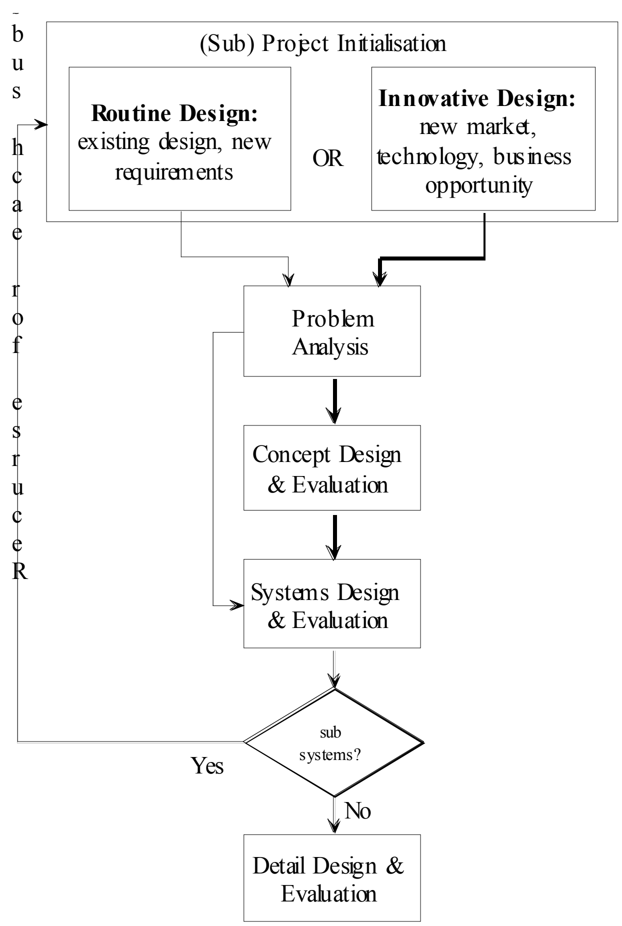

Figure 5: A generic design process (*not* a product development process).

The design process is *recursive* in that it starts by treating the product as a whole, then breaks it down to finer levels of detail on each loop. Designing begins by considering the product as a whole, and coming up with overall ideas/concepts for the product as a whole. One of the results of this step of designing is a list of subsystems that must be in the product. We can then repeat the very same process for each subsystem to determine sub-subsystems. This *recursive* process continues until the design is broken down into elements so "small" and simple that they can be designed directly (or listed out from a catalogue of parts that can be purchased).

2.5.1. Initialisation

At this task, a team is formed, the basic "problem statement" is discussed, some background research may be required, and a general strategy for the product is established.

A key strategic decision in this task is the degree of *innovation* required for a suitable design solution.

There are basically two alternatives here. First and most typically, is to base a new design on an existing design, and to only refine or optimise the design. In this case, the degree of innovation expected for the product *as a whole* is relatively low, although certain elements (subsystems, parts, etc) may still be highly innovative. This kind of designing, so-called *routine* or *variant* design, accounts for more than 80% of design projects. Innovation requires taking risks, which is dangerous from a business point of view in modern settings. So innovative design is something done relatively rarely.

In routine design problems, the process follows the *thin arrows* in Figure 5, and requires that extensive information about past designs be available to the design team. In routine design one proceeds from the *problem analysis* stage directly to the *systems design* stage, which involves designing the major functional units of a product. Since the *design concept* (i.e. the general idea of the kind of product to be designed) is given in the form of the existent design, there is no real need to do the *concept design* stage.

Alternatively, there may be a strategic decision to design a highly innovative product. In this case, the thick arrows in Figure 5 represent a typical design process. If one has decided to pursue a highly innovative design, then there is no design concept at the outset, and the *concept design* stage must be done before *systems design*.

Once one has established specifications for the main systems in a design, one must decide if there will be *sub*systems. A subsystem is just a component of a system that is itself a system. For example, an automobile is a system and so too is the automobile's engine.

If there are subsystems, then each subsystem can now be treated as another design problem. At the end of the systems design stage, subsystems will be identified as "black boxes" with certain requirements arising directly from the systems design and problem analysis stages. These requirements are fed back into the first stage of the process and one can now begin designing each subsystem, using *exactly* the same process as before. The basic steps of a typical design process are laid out in the following sections.

2.5.2. Problem Analysis

The first phase, **problem analysis**, involves the research that will form the basis from which the majority of design and development ideas originate. This involves the task of determining the market for which the product is being produced, what features and functions the market is looking for and what existing products exist if any. From this information, specific technologies, manufacturing processes, testing and quality procedures, and economic plans are generated specific to the product market.

Ideally, the problem is stated only *functionally*; that is, with respect to what a product must *do*. A badly stated problem will make commitments to particular shapes, parts, materials, etc. This is bad because this usually limits the designer's activities unnecessarily. Design is about discovering the shape and structure of a design. Giving requirements that include elements of shape and structure is like giving part of the design solution. However, the solution has not yet been designed – so how can we know those shapes and structures are best?

It is also typical that the customer or client bringing you the design problem does not really understand the problem. It will be your job to work with the customer/client/user to discover what the problem really is. For example, as a result of focus groups, it was determined that users of hospitals wished the entire hospital would come to them, rather than them having to go to the hospital. Designers evolved this obviously impossible user need into the much more manageable strategy of minimising patient movement within the hospital once the patient is admitted.

As a design proceeds, more and more requirements are identified, but these are usually derived from the initial requirements, and are under the control of the development team. The initial set of requirements, however, comes from *outside* the development team – the team has relatively little control over them. The initial requirements also typically form the basis of the *contract* between the development organisation and the clients or users. There are obviously many important ethical and legal implications arising from this. So, during problem analysis, it is important to have legal and upper management representatives present, as well as the technical experts.

The result of the problem analysis phase is a requirement specification – also called a *product design specification*. This document formalises the expectations of the clients/users in a form that the development team can work with to design and manufacture an appropriate product. As the design evolves, the product design specification will become the touchstone for evaluating the design. A design that meets or exceeds the requirements in the specification is acceptable; all other designs are unacceptable. A design that does things that are not called for in the specification (an *over*designed product) can be just as bad as one that does not meet the requirements.

2.5.3. Ideation

The second phase, **ideation**, involves the development of ideas of what the product might be like from the point of view of only some of the key requirements identified during problem analysis. This allows each team member to assume a particular role in which he or she is most comfortable and has useful expertise. For example, one team member might consider only the issues of cost and safety, and disregard issues of usability and manufacturability. Other team members will look at other subgroups of the requirements, and come up various with design ideas that only treat those specific requirements.

For example, a few years ago, a segment aired on the news program Nightline with Ted Koppel, in which the prestigious design firm IDEO redesigned a shopping cart from scratch in only five days. One team worked *only* on the issue of carrying small children in the cart safely. One of their ideas was to use "velcro pants" that would firmly hold a child in place without harming the child. While this was seen as a relatively "whacky" idea, it started a thinking process. The velcro restraint solution was inappropriate, but *why*? The IDEO designers reasoned that while it achieved the goal of helping to control the children while they rode in the cart, it was actually very unpleasant for the child (although the parent might not mind). With this insight, they realised that the child was as much a user of the cart as the parent, and they eventually designed a small play surface integral to the shopping cart's handle. The combined play area and cart handle also acted as a physical restraint to keep the child properly seated in the cart. The play area would keep the child distracted, letting the parent shop.

Ideation is useful because the human brain finds it difficult to deal with more than a few units of information at a time. By treating only a few requirements at a time, a designer can come up with partial solutions that can be woven together in the next phase of the process.

Ideation can be ostensibly the most creative phase of designing. One may use any one of many different "creativity methods" here. However, none of these methods can guarantee creativity. Instead, they promote an environment where (a) new ideas are more easily had, and (b) designers are more likely to recognise a good idea when they have it.

There are two key factors that impact positively on the ability to ideate: personal experience and the ability to cognitively "associate" experiences (as in the psychological method of "free association" – indeed, free association is a foundation of many creativity methods). Personal experience can come from anywhere – travelling, reading, summer and part-time jobs, community service, etc. The experience need not – indeed, *should* not – be in your chosen field. The broader the experience, the larger the base from which you will be able to develop new ideas. The ability to associate lets you connect different experiences in different ways and "see" new ways of doing things. There are various methods one can practice individually or in groups to help form associations.

2.5.4. Conceptual Design and Evaluation

The third phase, **concept design and evaluation**, combines different ideas together into total design concepts for a product. Here, different ideas (one, say, for cost and safety, and another, say, for usability and manufacturability) are blended together and then evaluated with respect to the requirements determined during problem analysis. The result of this phase is a "winning" concept that will be detailed in subsequent phases.

Recall that each idea addresses only some of the requirements of a design problem. A concept, on the other hand, must address *all* the requirements. Given a list of ideas, one may randomly choose any set of ideas that between them cover all the requirements, and seek to combine them in various ways. Say the requirements can be grouped under the headings: cost, quality, manufacturability, and safety. Next, say we had developed one idea that treated cost, quality, and safety, and another idea that treated cost and manufacturability. These two ideas can be combined because they cover all four base requirements between them.

One can, using this method, go through a list of all the ideas that were generated, and come up with all the possible concepts that arise from the combination of the ideas. This provides a base set of concepts.

The next step is to evaluate the concepts – to put them in a *qualitative* rank ordering from best to worst. The ranking must be qualitative, because we simply have not fleshed out the concepts enough to do detailed quantitative analyses. It is vital to appreciate this point: there is no need to have hard "scientific" information to evaluate designs early in the design process. One can, with a little practice, become extremely proficient at judging designs *conceptually*. The importance of this is that the early decisions in a design process are the most critical – they are, in fact, far more important than the downstream decisions that must be made later in the process.

There are many techniques used in industry and written about in books and articles about. In a separate module, a few of the most popular ones are discussed in detail, including design reviews, evaluation matrices, and kano models. The point of all these techniques is to use the qualitative information that is available during concept design to assess the viability of designs. These techniques all work best when done in group settings rather than individually, so that the sum of all the experience and expertise of the whole team can be brought to bear on the evaluation process.

2.5.5. Systems Design

The fourth phase, **systems design**, involves the definition and design of the major *components* of the product. Whereas concept development focused on the primary function of the product as a whole, the systems design phase involves the development of the product's internal arrangement, and the interfaces between them. As products become more complex, controlling the interactions and interfaces between their components and sub-systems becomes a crucial aspect of ensuring a successful design.

Systems are designed from a functional perspective – we do not (yet) associate a system with a particular assembly of parts. Parts and assemblies are *how* a system implements its function. Since we are designing in a top-down direction (starting with the product as a whole, and moving *down* towards the details), we must start with the function and move towards the specification of assemblies and parts. But we're not there yet.

Identifying the major systems in a product is a process of synthesis (creativity tempered by rational thinking). Nonetheless, one can generally use the product requirements derived from *problem analysis* (Section 2.5.2) as a guide.

For example, if we were designing an elevator, one requirement might be that *the elevator lift and lower people and cargo*. This immediately suggests (a) a *containment/carrying system* to hold people/cargo while lifting/lowering them, (b) a *lifting system* that actually does the lifting and lowering, (c) a *structural system* that holds the lifting and carrying systems in place, (d) a *control system* to let users control the operation of the elevator, and (e) a *power system* that manages and conditions the power needed for the other systems to work. We've defined five functional systems here, without making any commitments to shape, form, material, or specific methods of operation. This is good: remaining in the functional domain for as long as possible is one way to promote innovative ideas.

More than just identifying systems, though, is the need to "wire them together" to specify how they should interact in the final product. This can be much trickier than just identifying the systems. The easiest way to do this is with a "block diagram". Draw each system you've identified as a block, and connect the blocks together with arrows showing "flows" between them.

In general, there are only three kinds of flows between systems: flows of mass, of energy, and of information. Mass flows are often shown as thick arrows, energy flows as thin arrows, and information flows as dashed arrows. For example, in the elevator, the movement of people and cargo from outside the system into the elevator and back out again could be represented by a thick arrow coming from outside the elevator system and connecting to the *carrying system* mentioned above. Also, an energy flow would connect the *power system* to the *carrying system*, the *lifting system*, and *control system*. Since we have not yet made specific decisions about how the elevator will work, we say energy and not "electricity" because we may choose to run the elevator on other kinds of power. (Can you think of one?)

The result of systems design is a roadmap or *architecture* showing the principal functional elements of the product and how those elements have to interact to provide the overall product requirements. The product architecture is essential as a guide for subsequent phases of designing. As the design evolves, it will have to be validated as a way of implementing the architecture. Anything the product does that is not specified in the architecture is either undesirable or an unneeded complexity.

2.5.6. Recursion to Subsystems

Consider the elevator design problem discussed above. It should be obvious that systems such as the *carrying system* are themselves complex things requiring further designing. For such systems, we can simply begin the design process over again. The architecture that results from systems design identifies how inputs provided to the product as a whole are transformed into outputs. Systems design also "breaks" the product into subsystems like the carrying system. The architecture also shows which inputs to the product must pass to its constituent subsystems, and which outputs from the subsystems eventually become outputs of the product itself.

These subsystem inputs and outputs seed the requirements engineering stage of the subsystems themselves. That is, we can now treat each subsystem as a separate design problem for which all the steps of the design process can be repeated.

When we repeat the process for each subsystem, we will end up with a number of sub-subsystems. We can continue this recursive design process until we reach a point where system elements are so simple that single parts (rather than systems) will be sufficient. When we reach this point we will have developed:

a) a hierarchy of systems connected by the requirements that every element of the product must meet;

b) a map of how all the systems and system elements interact;

c) a guide for developing systems integration procedures (i.e. putting all the parts back together again to validate that the product actually does what it is supposed to do); and

d) a set of requirements for every individual part of the product.

All this information "falls out" of a properly conducted design process. Armed with this information, we can now begin to design individual parts. This is the realm of *detailed design*.

2.5.7. Detail Design

The fifth phase, **detail design**, involves all the development of all the details of the product. This includes the final geometry and tolerances for the product, as well as the material selection. Manufacturing process and suppliers are finalized and all required tooling is designed. Building up a manufacturing facility and process for the product can occur in earnest only once most of the product design is finalised. At this stage *alpha prototypes* are produced using parts which are usually not made by the actual manufacturing process (since it is still under development), but are made with very high attention to detail and tolerances. Alpha prototypes are used to evaluate the product as a whole, as well as all component interfaces. Alpha prototypes are usually evaluated internally and by specially identified customers/clients.

By the end of this phase, the product as been fully designed and a great deal of time, effort and money have been invested in the project. Any errors in the design that escape notice here will likely remain in the product to be sold. Depending on the product, this could be quite dangerous. This is why it is absolutely vital that all those involved in the design process have regular communication between disciplines and regular reviews of the work completed.

By now, virtually all the product design work has been finished. What little design remains to be done will be in response to (hopefully) minor changes and improvements especially pertaining to the manufacturability of the product, and further product tests.

2.5.8. Other design phases

As product development continues toward full-scale production, the amount of design work decreases. However, there is still opportunity and need for some design input.

During **testing and refinement**, the production of *beta prototypes* tests product performance, reliability and endurance. Beta prototypes are produced using the *actual* manufacturing process that will be used in full production. Beta prototypes are used to evaluate both the product *and* manufacturing process. Sometimes, design problems only become evident in "real" manufacture. The beta prototypes are intended to let engineers catch those kinds of problems and address them before full-scale production starts. Design expertise is important here, because the schedule for production ramp-up will have been set and cannot be changed without causing a great deal of difficulty for the company. Even if their role is attenuated here, designers must react quickly and with great accuracy to any problems that require either redesign of the product or of the manufacturing system.

Finally, **production ramp-up** is the first full run of the entire production process. Usually used to train those who will be involved in the production process, products produced in this stage usually undergo a very detailed scrutinizing by quality control and then either kept as production references or supplied to preferred customers. Any other glitches or hiccups in the production process are usually worked out in this phase. Design engineers are important here, to diagnose problems and come up with solutions very quickly.

References

[1] K.T. Ulrich and S.D. Eppinger. **Product design and development.** McGraw-Hill, 1995.

Bibliography

[1] C.L. Dym and P. Little. 2000. **Engineering Design: a Project-Based Approach**. Wiley & Sons, New York.

[2] G. Voland. 2004. **Engineering by Design** (2/e). Pearson/Prentice-Hall, New Jersey.

[3] K. Vicente. 2003. **The Human Factor**. Knopf Canada.

[4] D.A. Norman. 1988. **The Design of Everyday Things**. Basic Books.

[5] S. Pugh. 1991. **Total Design: Integrated Methods for Successful Product Engineering**. Addison-Wesley, Wokington England.