Chapter 9
The Lean Product Development Process

This chapter describes the product development phases and activities, which are detailed in Chaps. 10–13. We consider four phases in the PDP: (1) the portfolio phase, which produces a general vision of the product, both aligned to the value pulled by the market/customers and consistent with the company's strategy and capacity; (2) the study phase, which includes the identification of the value pulled by both by external and internal stakeholders, the value proposition activities that outline the chief engineer's vision of the new product, and the value delivery planning for the next phases; (3) the execution phase, including the design, development, production/ramp up of the products and/or services that deliver the pulled value; and (4) the use phase when the resulting product/process is followed-up until its discontinuation. The Value Function Deployment (VFD) technique and the Product Development Visual Management Boards (PDVMB), which are also presented in this chapter, support the Lean Product Development Process execution.

9.1 Introduction

Womack and Jones [1] noted that the ideal process of designing a product should function congruently with single-piece flow in manufacturing. It suggests that this process should represent a continuous flow of value creation, from conception to production, without stops due to paperwork and no returns for error correction.

From their study of the Toyota Product Development System, Morgan and Liker [2] identified two main phases in the Toyota's lean product development process: (1) the study phase, *kentou*, and (2) the execution phase.

During *kentou* the PD teams can anticipate, study, and resolve problems, completing such tasks as fundamental design decisions, identifying failure modes, designing in countermeasures, and setting cross-functional objectives. *Kentou* results in far fewer engineering changes and creates process flow by allowing companies to focus on downstream task execution. It also provides a formal

M.V.P. Pessôa and L.G. Trabasso, *The Lean Product Design and Development Journey*, DOI 10.1007/978-3-319-46792-4_9

structure for cross-functional teams to "design in" solutions, which is far less expensive than solving problems or "fixing" designs later in the process. During the study phase, the product is conceived, a performance envelope is defined, and the solution space is explored in order to find a balanced (value/risk) design.

Once *kentou* is complete and the development strategy is set, the execution phase may begin. By the time it reaches this point, the LPDO has made a full commitment to the product, and has begun to invest significant sums of money in tooling and in its suppliers. Because of this investment, it is financially critical to have a high-velocity PD process with radically shortened lead times, by focusing on precise execution and smoothing product-to-market delivery [2]. The company's goal from this point forward is to optimize capital investment, match quick cycle-supporting or embedded technology lead times, make decisions closer to the customer and other relevant stakeholders, and react quickly to changes in the competitive environment. Creating flow by synchronizing product development activities is one of the most powerful ways to increase speed.

Rather than describing how Toyota works, our objective here is to help companies implement lean PD systems themselves. While the Value Function Deployment (VFD) technique, also described in this chapter, is the backbone of our implementation model, you can use other ways to achieve similar results provided you keep the same philosophy. Therefore, the book's proposal is to focus on the concepts of continuous improvement, value delivery, and waste reduction, as presented in Part II, while keeping in mind the cultural, organizational, and knowledge management aspects, as discussed in Part III.

9.2 The Process and Its Phases

The PDP model we use here aims to:

1. support the practical application of the concepts previously described in Parts II and III of this book; and
2. fit the VFD and the PD Visual Management Board (PDVMB), which are further described in this chapter.

Even though most tools and techniques can be used in the lean way experience shows that is very difficult for a person used to applying tools and techniques with the mindset bounded by a certain paradigm to do that in a different way. Unconsciously he or she turns back into the previous way. This is the reason we proposed the Value Function Deployment (VFD) technique [3, 4]. The VFD acts as a backbone of the Product Development Process, always reminding the practitioner about the lean directives while he/she can apply the tools and techniques he is accustomed to.

In the same way, the presented PD Visual Management Board (PDVMB) is a sample of simple *obeya*, which provides visual management. Our experience also shows that people struggle to initially define what is important to be included in

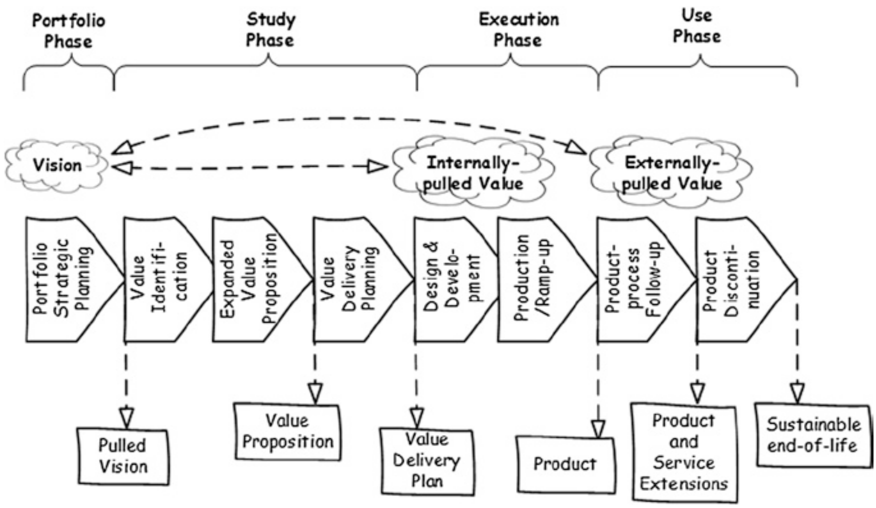

Fig. 9.1 Our PDP model

the *obeya*, and how to make the management by using it. The PDVMB is a starting point to defining your own *obeya*, and include what we believe is the minimum information to manage a LPDDP.

We divide the product lifecycle in four phases (Fig. 9.1), where the output from each phase is both aligned with (1) the value pulled from the final user/customer and other external stakeholders, and (2) the value pulled from the subsequent phases into the PDP (internal stakeholders). The phases are further detailed into groups of activities.

The phases are briefly described in sequence. The study and execution phases' activity groups are further detailed in Chaps. 10–13.

9.2.1 Portfolio Phase—Portfolio Management Activities

This phase includes all the portfolio management activities and ends by delivering a "product vision" which presents a general description of the expected development results and their market impact plus any constrains and assumptions initially bounding the product development conceptual work.

The result from this phase is a general vision of the product, both aligned to the value pulled by the market/customers and consistent with the company's strategy and capacity.

Good portfolio management is a key success factor to the LPDO. Portfolio management is about resource allocation (how your business spends its capital and human resources) and project selection (ensuring that you have a steady stream of big new product winners). Therefore, portfolio management has four goals [5, 6]:

1. Guarantee the strategic alignment where the final portfolio of projects is strategically sound and truly reflects the business's strategy.
2. Maximize of the return of the investment (both in terms of the company's objectives and, of course, the money).
3. Balance (long/short term and high/low risk) the development programs in the various markets the business is in.
4. Create a development cadence that balances value delivery through products/markets and the company's resources and capacity, thus reducing waste, unevenness, and overburden.

In fact, these goals act like valves defining which projects will enter and stay at the product development funnel, while regulating the flow of development projects (Fig. 9.2). These development projects can be either the development of complete new products or the improvement of existing ones.

The strategic alignment is guaranteed by taking into account that all development projects respond to both value pulled by the customers/market and the value pulled by the shareholders. In order to do that, the company needs to have a clear vision about itself, its products, and the related technologies it wants to master. Business, product, and technology governance play an important role at this

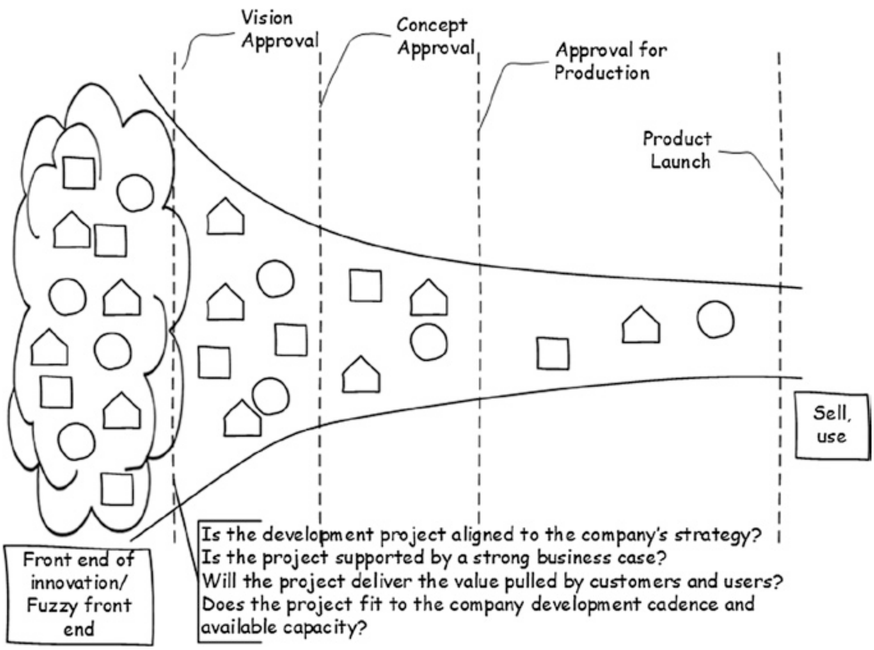

Fig. 9.2 The portfolio management goals through the funnel

moment since they can guarantee not only the individual development projects' alignment, but that the projects from the portfolio have a positive impact on each other.

As a rule of thumb, the LPDO should avoid developing new technology in individual development program critical paths. Therefore, technological innovation is strategically focused, often in response to a request from a chief engineer, and aims to create off-the-shelf proven technology. In the case technology development cannot be avoided, the incorporating of this new technology should be considered as one of the SBCE alternatives, as presented in sequence and detailed in Chap. 11.

Considering the PD funnel, the LPDO only triggers the concept development (normally done by the chief engineer and his/her staff) after a strong business case (a "value case") is achieved, and respects the cadence discipline (i.e. when the company, for a certain product, releases periodical updated product versions, like cars, cell phones, etc.). Not respecting the company's development capacity might lead to waste, unevenness, and overburden through the development portfolio. These ripple effects are one of the main causes of firefighting through and across projects.

9.2.2 Study Phase—Value Identification Activities

After receiving the Product Vision, the chief engineer or its equivalent starts the study phase's value identification activities which aim to provide deep understanding of the true value to be incorporated into the product (and/or service). All the related stakeholders through the value chain, both internal and external, must be considered and the value they pull understood. As a consequence, all the stakeholders from the use and execution phases should be listened to.

Different stakeholders have different importance; also, any pulled value item is associated with some risk (business, market, technical, etc.). Sometimes the identified value challenges the vision-related constraints, so if a trade-off solution is not achievable, the vision must be challenged (negotiated).

The objective of the value identification activities is to create a structured and unambiguous value items set, rooted in the stakeholders' pulled value, and which serves as reference for all the development team, therefore guiding all the development program activities.

This is the moment during the study phase when the value pulled by all key stakeholders is consolidated. In this process, explicit and implicit agreements are made in order to balance all stakeholders' needs, resolve conflicts, and include tangible and intangible values (protecting the environment, meeting the technical specifications, meeting the shareholders' expectations, providing an environment of rewarding work, etc.), or anything that has been forgotten.

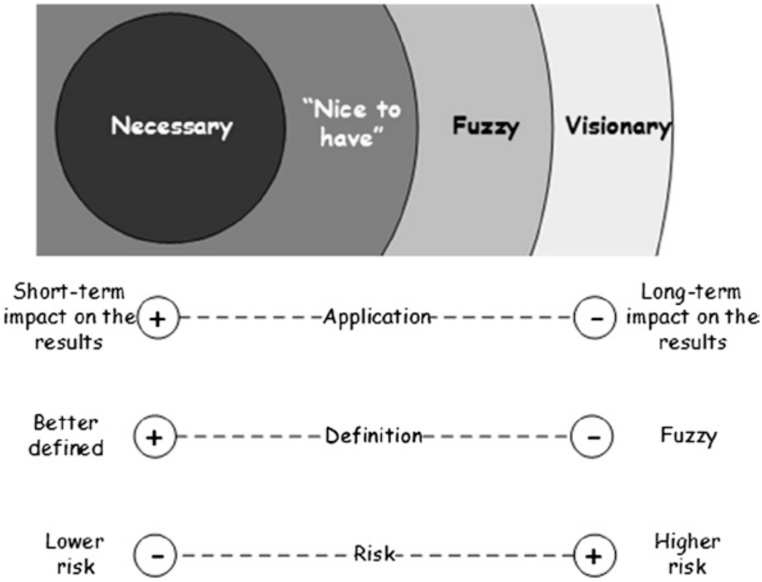

Fig. 9.3 Requirements classification

The value items set formalize the program goals, defines relationships between stakeholders, and sets the cost and time constrains. It creates the program "flight envelope" to deliver the desired value.

Even though the value, as perceived by the customer, is the primary pull force to the whole development, only by considering the value stream based on the needs from all key internal and external stakeholders can you ensure that all the people, groups, and businesses which can impact decisively in the development project will contribute with work and resources to ensure the project's success. Often, the efforts to make the value visible for the various stakeholders require the inclusion of additional development. These activities are though necessary, since anyone who does not perceive receiving any value will tend to stop contributing.

Each development project carries some risks. In fact, since several and sometimes conflicting pulled value might arise, the LPDO must discern the real value (the ones that will trigger the buying decision) from: (1) "nice to have" features/characteristics which consume resources and increase the development risk; (2) fuzzy requirements which still are not completely clear, maybe even in the customer's mind; and (3) visionary requirements which may require great architectural or technological changes and the related risks (Fig. 9.3). Any requirement out of the "real value list" could postpone the development and negatively affect the planned cadence. Considering or postponing (in the case where a new version of the product is released from time to time, i.e., annually) the other requirements should be balanced against the associated risks. In any case, the strategy chosen should allow dropping them at any time and with minimum impact on the development flow.

9.2.3 Study Phase—Value Proposition Activities

The Value Proposition outlines the chief engineer's vision of the new product; communicates customer-defined value and product-level performance objectives, and aligns the product-level performance goals of the entire program team; in summary, it communicates all the pulled value in a simple, unambiguous, and final written document [2].

The product development project aims to deliver the value proposition which might range from a particular product to a completely new or modified value chain.

Often during the execution of a development project, the development team might face conflicting pulled value issues. If the development project lacks a clear value proposition where priorities are set, the team might make decisions by having only partial knowledge.

In order to finalize the value proposition, the next challenge is to define which functional architecture is the preferable choice to deliver the product/service.

In most cases, PD is an open-ended problem, therefore accepting multiple possible solutions. As a consequence, each of the product's functions can be implemented in different ways. These alternatives, though, carry intrinsic risk, so the LPDO must carefully chose the path to follow, since this can be the difference between success and a huge failure.

In order to reduce the chances of iterative loop-backs or plan modification and resource changes, which might create ripple effects in the whole company's PD portfolio, the LPDO uses SBCE (see Chap. 6). As a consequence, the value proposition might include different product's subsystems alternatives, where the set have very low chances of causing rework loop-back due to failures of all alternatives. SBCE explores the solution space, supports the no-compromise attitude, allows emergent solution (combining) and creates knowledge.

Experience shows that the cost of applying SBCE is equivalent to applying the point-based approach, considering the average needed rework cycles. The great difference among them is that SBCE greatly reduces the risk of overtime, while generates more knowledge by understanding the several design alternatives of the product.

Although equivalent, SBCE requires, though, more resources to carry out simultaneously the different product design alternatives. These resources might not be available in all companies, therefore the need of prioritizing in which product's parts/modules/subsystems to apply the SBCE. We consider that the product's parts/modules/subsystems which deliver more value and/or are more risky as critical to applying SBCE (see Chap. 11 for details about the prioritizing strategy).

9.2.4 Study Phase—Value Delivery Planning Activities

During the value delivery planning activities, all the teams that will work on the project are defined and a set of pull events is determined. By having the teams and pull events, it is possible to create a plan that embeds real concurrent engineering and flow.

As we mentioned before, in the traditional PDP, the development plan is followed until it fails (point-based), for whatever reason, and then follows a series of iterative loop-backs, or plan modification and resource changes. As a consequence, the results from the work performed during the execution phase are pushed through the activities. A systemic view of the solution (and often only part of it) is only achieved in phase gates. These gates, besides damming information, often lead to unnecessary delays and inventories.

The lean principles state that no process along the value flow should produce an item, part, service, or information without direct request from the afterward processes. By pushing results through the PDP, the company is just accumulating a stock of information and items that no one wants yet and that might become obsolete before being used.

The best way to understand the logic and the challenge of pull production is to start with a real customer expressing a demand for an actual product and walk the other way, going through all the steps required to bring the product to the customer.

This promotes high flexibility, allowing all the activities along the process to produce exactly what the customer (either internal or external) wants and when he wants it. Moreover, the reduction in response time for fulfilling the consumer needs speeds up the return on investment and reduces inventory even in a complex production flow.

Applying the "real" pull system concept into product development is a challenge. Each development project is unique; therefore, there is no fully predetermined way of how to build a product. Subsequent processes cannot pull definite information from its predecessors, since they are neither aware of the outcome of the work they will perform, nor of the final product with all its specifications.

It is possible, however, to get a good feel for what to expect, since the activities follow a logical sequence and the history from previous similar projects and information gives a good idea of the necessary inputs and outputs to be generated.

In the case of a development project, the important thing is to let the customer pull the value of the performing team. To make this possible, the development activities must be connected in a simple way and help eliminate waste from them.

As a consequence, a pulled value delivery planning ensures progress and project quality. Instead of phase gates, which dam lots information and stop the flow, pull events based on tangible results such as models, prototype ready systems, etc. allow the flow [3, 4].

The pull events relate directly to the value items, i.e., the scope of an event is associated with valuable items and their effectiveness measures. Unlike phase gates, pull events are part of the development value stream, and cannot be eliminated.

As a result, pull events have four roles, namely they (1) determine that there has been progress in the effective delivery of value; (2) ensure that information on the project will be pulled, not pushed; (3) allow the combination and the strengthening of alternatives during the SBCE; and (4) are learning moments as they allow reflection (*hansei*) about the progress of the work and the results obtained through adopted strategy.

9.2.5 Execution Phase—Design and Development Activities

At this phase, all the module development teams will produce their deliverables in a fast and synchronized way according to the sequence of defined pull events.

The LPDO maximizes the return of the investment by guaranteeing that the product to be developed has been pulled by the customer and that the value chain is aligned both to the goal and within itself. As a consequence, after starting the execution, the LPDO uses decision analysis (e.g., cost/benefit) to find the best alternatives to keep going rather than for deciding whether it should continue or stop the project. Design strategies (i.e., DFX and DTX) aligned to the pulled value set also multiply the impact from the development effort and expedite the return of the investment.

Pull events foster concurrent engineering, are opportunities to reveal quality problems, and support knowledge creation. In this context, planning is decentralized, allowing different groups to realize their own plans to achieve the pull events. For example, narrowing the sets of points in SBCE are pull events.

As a consequence of continuous improvement, the PDP can be seen as a spiraling and iterative process through the PD funnel, each cycle corresponding to one PDCA round (Fig. 9.4), where the "C" corresponds to a particular pull event.

9.2.6 Execution Phase—Production/Ramp-up Activities

The activities on this phase will drastically change according to the kind of product and the related production expected rate. One-of-a-kind products, for instance, can be even the final prototype from the development phase.

During ramp-up the product production and service delivery begin. Energy supplied, manpower deployed or quantities produced are gradually increased. At this moment the production process is proven, and there might be change request to adapt either the product or the production process to support full power production.

Once these initial issues have been solved, production is adjusted to fulfil the market demand.

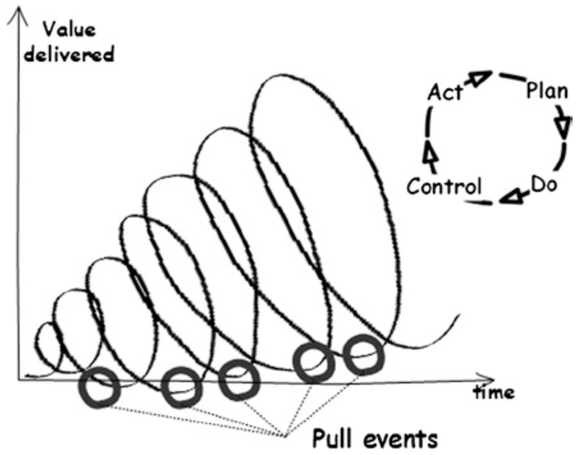

Fig. 9.4 Product development as a PDCA

At this phase the tools and techniques from the Toyota Production system are fully applied, and the development system tools and techniques are only necessary when a product/process change is requested.

9.2.7 Use Phase—Product/Process Follow-up and Product Process Discontinuation Activities

Even though this is the last phase to actually happen, all the PDS is based on it. The initial understanding of the use phase triggers the portfolio phase in order to consider this perceived need a candidate for a product development project. Even after a selection is made during the portfolio phase, the understanding of the use phase is further explored during the study phase in order to guarantee that the product to be delivered will match the pulled value. This phase includes the "Product/Process Follow-up Activities," and the "Product Discontinuation Activities," which comprise the product use, training, maintenance, evolution, and discontinuation.

9.3 The Value Function Deployment—VFD

The Value Function Deployment (VFD) [3, 4] technique described in this section applies the lean principles based on value creation and waste reduction to derive a project activity network that entails a sequenced set of confirmation events. These events pull only the necessary and sufficient information and materials from the product development team.

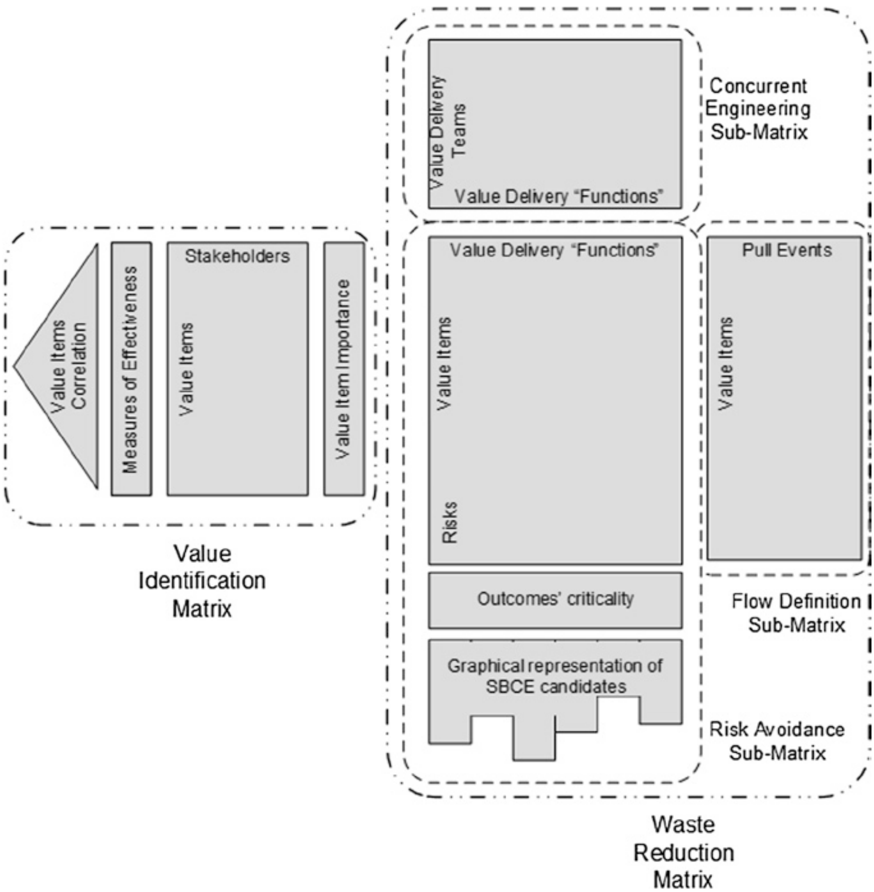

Fig. 9.5 The value function deployment matrices

The VFD is an adaptation of the Quality Function Deployment (QFD) technique and is composed of two interconnected matrices, the value identification matrix and the waste reduction matrix (Fig. 9.5). The former captures, prioritizes, and shows the correlation between all the value items expected by the project's stakeholders. The latter deploys the value items to the value delivery functions, calculates their criticality (rework avoidance sub-matrix), correlates the functions to the teams responsible to implement them (concurrent engineering sub-matrix), and defines the events that will pull this value from the teams (flow definition sub-matrix).

The VFD matrices' core elements are defined in sequence:

- **Stakeholders** are individuals or organizations that are actively involved during the development or whose interests may be affected by its execution or outcome.

- **Value** for a given stakeholder is the complete and balanced perception of the various benefits provided by the results from the development process. The value is stated in the stakeholders' terms and might not be free of ambiguity.
- **Value items** are the result of splitting the value into more specific and measurable elements (by asking "why you need that?" or "what do you mean by that?"). They can be functions, performance, level of acceptable risk, etc.
- **Measures of effectiveness** (MoE) are reference parameters used to analyze the conformity of the PDP results in relation to the stakeholders' expected value. They explain how you are going to perceive that the value item has indeed been incorporated into the product/service.
- **Value items correlation** indicate if two value items are conflicting, meaning that trade-offs will be needed.
- **Value delivery functions** are system level functions that encompass or relate to the value to be delivered. We considered the functions of the product/service to be developed. Each value delivery function must be traced to at least one value item from the set; the value items themselves must relate to at least one value delivery function.
- **Value delivery teams** are responsible for delivering value by performing the value delivery functions. We divided the teams in two groups: those which deliver value by developing the product/service itself, and those which deliver value by performing supporting processes through the value chain. Therefore, the part of the organization responsible for designing the specific subsystems of the product populates the teams that deliver value via *product*. Similarly, the part of the organization responsible for the designing of the processes that deliver the value through the value chain (such as marketing, supporting services, etc.) are the teams delivering value via *processes*; indeed, these processes are paramount for the stakeholders to perceive that they obtained the total pulled value of the obtained project's benefits.
- **Outcomes criticality** refers to the amount of value and the level of risk to deliver this value by each value delivery function. As a result, the functions which deliver more value and/or are at more risk are the most critical ones.
- **Pull events** typically are tied to physical evidence of progress (presentations of models, prototypes, initial production, etc.). We recommend using: (1) integration events that create "boundary objects" as built engineering projects, mock-ups, prototypes, etc.; (2) successful endings of checks and validations, which are moments of reducing uncertainty and risk in the program. The pull events set creates a "ladder," where each step gets closer to the development success.

Considering the presented PD lifecycle, Table 9.1 shows how the VFD is applied during its phases.

Table 9.1 VFD matrices and the PD lifecycle

VFD matrices	PD study phase—activities groups
Value identification matrix	Value identification
Rework avoidance sub-matrix	Value proposition
Concurrent engineering sub-matrix	Value delivery planning
Flow definition sub-matrix	Value delivery planning

9.3.1 Value Identification Matrix

The VFD is centered on the value pulled by the stakeholders. The Value identification matrix provides a straightforward visualization of all the value items pulled by the stakeholders, how each value item can be measured during the development, how the value items correlate to each other, and their relative importance for the development. The value identification and grouping is divided into five steps (Fig. 9.6):

1.1 **Identify the stakeholders**: All the stakeholders, both external and internal, must be considered. External stakeholders are those related to the use phase, while the internal are those related to the execution phase. Failing to recognize the external stakeholders may jeopardize the products' market success. Failing to recognize the internal stakeholders may compromise the concurrent engineering and smooth product development, production, and logistic flows.

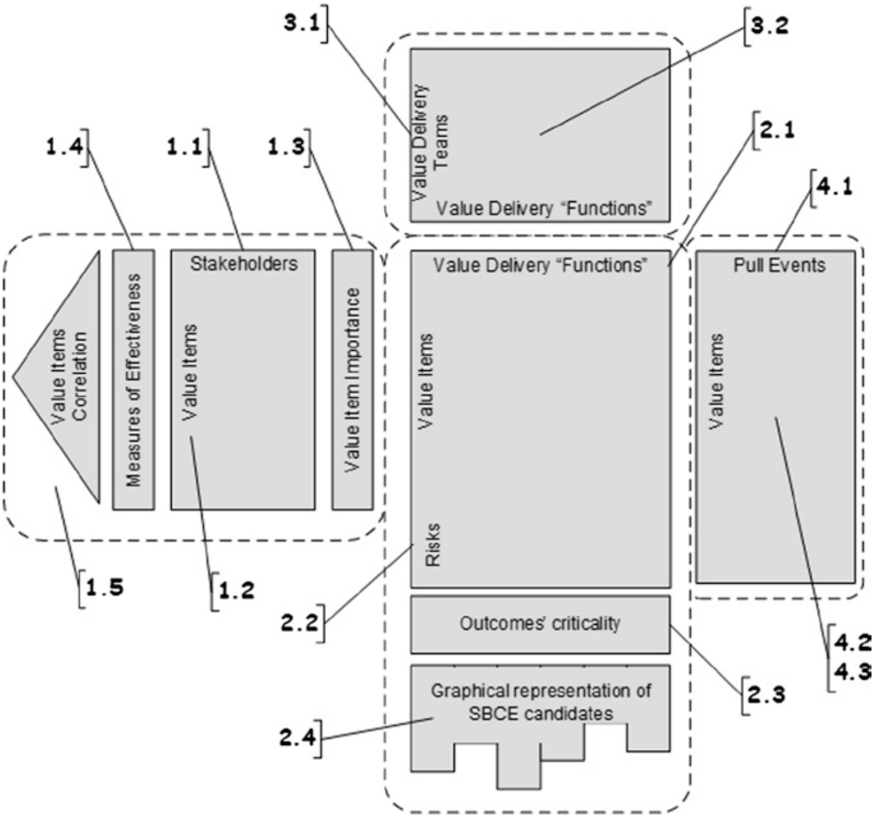

Fig. 9.6 Steps to fulfilling the matrices

1.2 **Analyze the value items**: This step includes understanding the stakeholders' needs and breaking them down into value items. The value items remove the ambiguity from the value set so the items can be addressed by the teams' deliverables and the progress on effectively delivering them can be measured. For example, a need presented as "be safe" can be broken down into items corresponding to the homologation tests defined by the product's regulatory agency.
1.3 **Prioritize the value items**: Each considered stakeholder has particular needs and thus rates the importance of the value items differently. The value items prioritization takes into account the combination of these ratings.
1.4 **Define measures of effectiveness (MoE)**: At least one measure of effectiveness must be defined for each value item. These measures allow the verification and validation that the value items were effectively incorporated into the project's results.
1.5 **Identify conflicting value items**: Conflicting value items are items that cannot be optimally delivered simultaneously (like having a car with high speed and low fuel consumption at the same time) if using the current company knowledge and capacity. The conflicting value items direct the creation of trade-off curves that, besides aiding the development team, are part of the company's knowledge assets. By challenging and improving the trade-off curves, a company becomes more competitive.

9.3.2 Waste Reduction Matrix

The objective of the waste reduction matrix is to support ways to reduce rework and guarantee the flow.

9.3.2.1 Rework Avoidance Sub-Matrix

The development of multiple alternatives prevents the early abandonment of promising solutions while giving room to the coexistence of preconceived alternatives. The SBCE helps guarantee the flow while reducing rework cycles: if one alternative on the set is proven to be inadequate, the others can still be used and no additional work is necessary. This process determines the most critical product functions or organizational value chain functions that will be developed through a set of alternatives, and is divided into three steps (Fig. 9.6 maps these steps on the VFD matrices):

2.1 **Define the value delivery functions**: This step determines the product's functions which deliver the complete value items set. Each function must contribute to delivering at least one value item and vice versa.

2.2 **Address risk response**: Identify the risks related to successfully delivering the development project results. The risks might relate to either incorporating the value into the functions themselves or issues that might arise during the development project management.

2.3 **Calculate the criticality of each value delivery function**: The functions' criticality is directly proportional to: (1) the amount and importance of value to be incorporated in these functions; and (2) the perceived risk to successfully deliver the expected value subset. The more valuable and the more risky, the more critical the functions are.

2.4 **Define the priority to parallel development**: The functions to be developed through a set of alternatives will be chosen by considering the restrictions imposed on the development project and the previously calculated criticality. The definition of the number of alternatives and the characteristics of each of the alternative will take place during the execution phase.

9.3.2.2 Concurrent Engineering Sub-Matrix

The strategy of using the functional architecture as the basis for determining the development team structure has great advantages for the application of SBCE. In this case, one team must determine the various alternatives, unlike functional organizations where this responsibility can be distributed among various groups, hindering the SBCE control.

The relationship between value functions and value delivery teams determines the need for concurrent engineering. This occurs because the effective delivery of a particular value item can depend on incorporating the results into different value delivery functions which are the responsibility of different teams.

This sub-matrix is divided into two steps (Fig. 9.6):

3.1 **Identify the value delivery teams:** This step determines which teams are responsible for the delivery of each function. These teams are either related to the product subsystems themselves or to organizational processes (such as marketing, quality, production, etc.).

3.2 **Define the contributing roles of each value delivery team**: This step maps the role of each team on delivering a particular function. After completely filled, this sub-matrix works like a Role & Responsibility Chart (RACI).

9.3.2.3 Flow Definition Sub-Matrix

No process along the value flow should produce an item, part, service or information without direct request from the afterward processes. The pull events are the backbone of the value flow and are important moments to knowledge capture; by pulling the value delivery, they allow the planning to reach execution. Every pull event is associated with physical progress evidences (i.e., models, prototypes, start of production, etc.). The pull event determination process is divided into three steps (Fig. 9.6):

4.1 **Define preliminary pull events:** To define a sequence of preliminary pull events, the development team can use the enterprise's standard process (if there is one), reuse historical information from previous projects, or consider best practices from the industry.
4.2 **Relate the pull events to the value items and risks**: A pull event scope is defined by the set of value items and risks it will check and how they will be checked (i.e. analysis, subsystem tests, integrated tests, etc.). A pull event must be related to at least one value item and/or risk, and each value item/risk must be checked by at least one pull event.
4.3 **Refine the pull event set**: The preliminary pull event set is refined until it meets the following criteria: (1) it must be capable of verifying the progress on the effective value incorporation and delivering during the project execution; (2) it must represent the value flow in order to guarantee the information pull, and not push; and (3) it must show the elimination of the risks that led to the development of multiple alternatives, allowing the combination and the reduction of the number of alternatives during the SBCE.

9.3.3 Systems Engineering and the VFD

During the definition and decomposition of the system to be developed system engineering design activities detail the system using a top-down approach, from conceptual design to detail design. The previous VFD description was made at the conceptual level, once it relates the pulled value to the value delivery functions. In order to guarantee value traceability and consider the SBCE risk reduction capabilities through the design phases, the VFD can and should be used at all design stages:

- Conceptual design (system-level): this is what we have already done in the study phase, when we checked which of the systems' functions were best candidates to SBCE, and looked for possible alternatives for supporting the subsystems that would perform these functions.
- Preliminary/layout design (subsystem-level): in the same way, the teams in charge of each subsystem can check which of its constituent modules are more critical. At this moment, a different VFD is built for each of the subsystem's alternatives. Depending on the system complexity, this breakdown has to be done in several steps, once the modules might be composed by submodules.
- Detail design (module level): Detail design goes until you reach parts definition. SBCE can also be applied to the most critical parts from each module, where alternatives might include chosen different parts, materials, or suppliers.

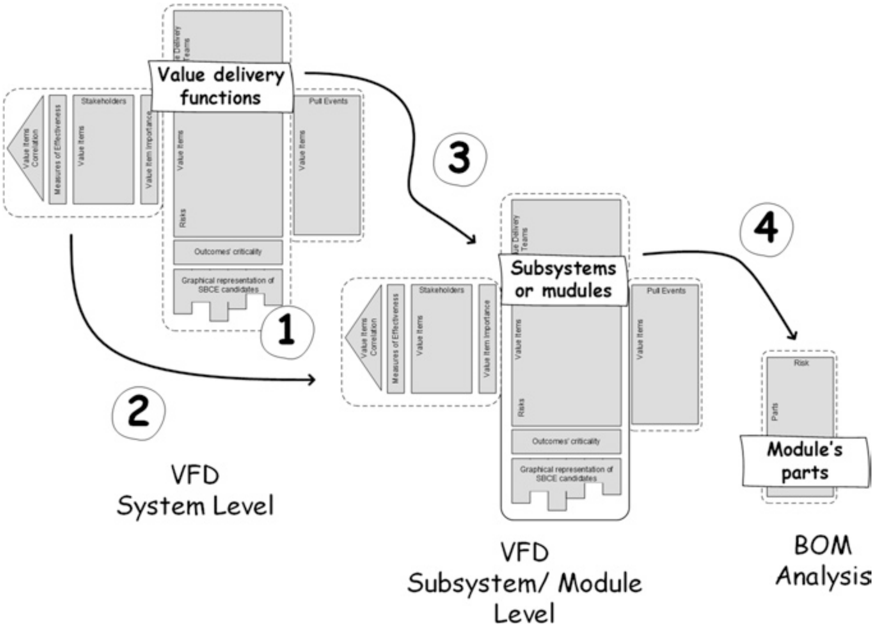

Fig. 9.7 Using the VFD during the design of the product/service

Figure 9.7 emphasizes some of the critical aspects when detailing the VFD into further detail levels:

1. Critical functions, which were chosen to SBCE, will lead to different subsystem alternatives; therefore one subsystem-level VFD has to be built for each of these alternatives.
2. Only the value that is related to the further levels of detail is carried out. Note that some internally pulled value items might be added, which is the case of including DFX directives. Externally pulled value items, though, can only be added at the system-level VFD, once they potentially impact the whole product/service.
3. The value delivery functions should be grouped into subsystems, which can be physical products or services, and these subsystems are further detailed during the design.
4. When analyzing the parts (Bill of Materials – BOM) there is no need to build a complete VFD, once the analysis is centered in the risk (different part numbers, materials and suppliers).

9.4 Product Development Visual Management Boards

As presented in Chap. 7, the *obeya* (big room) is a good example of visual communication. In order to support putting in practice the Lean PDP depicted herein, we developed *obeya* models to be used during the activities from the Study and Execution phases. These Product Development Visual Management Boards (PDVMB) function as continuously developing A3 charts.

The study phase PDVMB (Fig. 9.8) has the VFD filling as its focal point, the execution phase PDVMB (Fig. 9.9) has the product under development as its focal point, and the VFD keeps track of the development project progress and value alignment.

The study phase of the PDVMB supports the value proposition creation; communicates stakeholder-defined value, product-level performance objectives; and aligns the product-level performance goals of the entire program team.

The execution phase of the PDVMB keeps track of the product evolution during its design and developments and supports the concurrent engineering and change management.

Both the study and execution phase of the PDVMB include quality, time, and cost indicators. Quality is represented in the "compare current product/competitors/substitutes/new product" field, by showing the planned versus designed/developed product value delivery capacity. Time and cost can be tracked by creating an "S-curve" from the milestone chart.

The detailed PDVMB filling is explained in Chaps. 10–13.

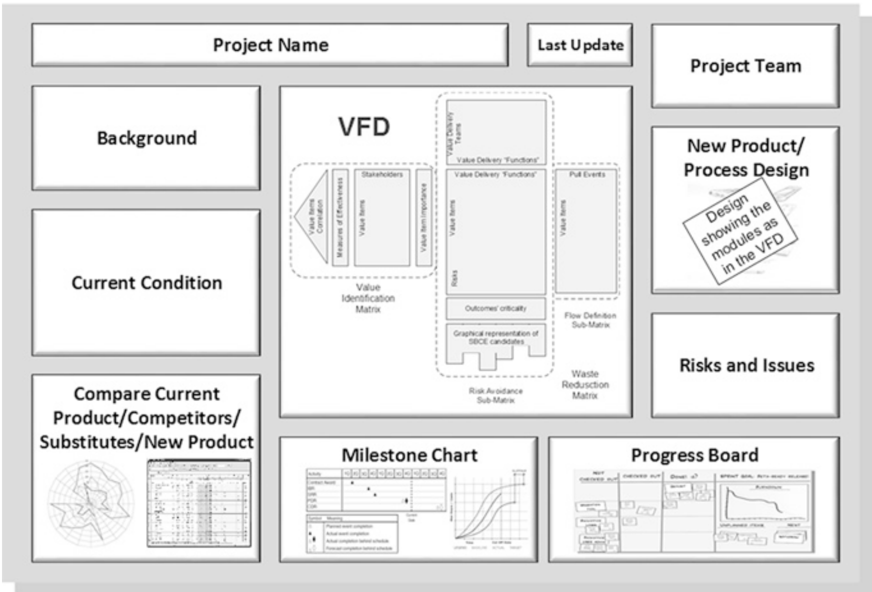

Fig. 9.8 Study phase visual management board

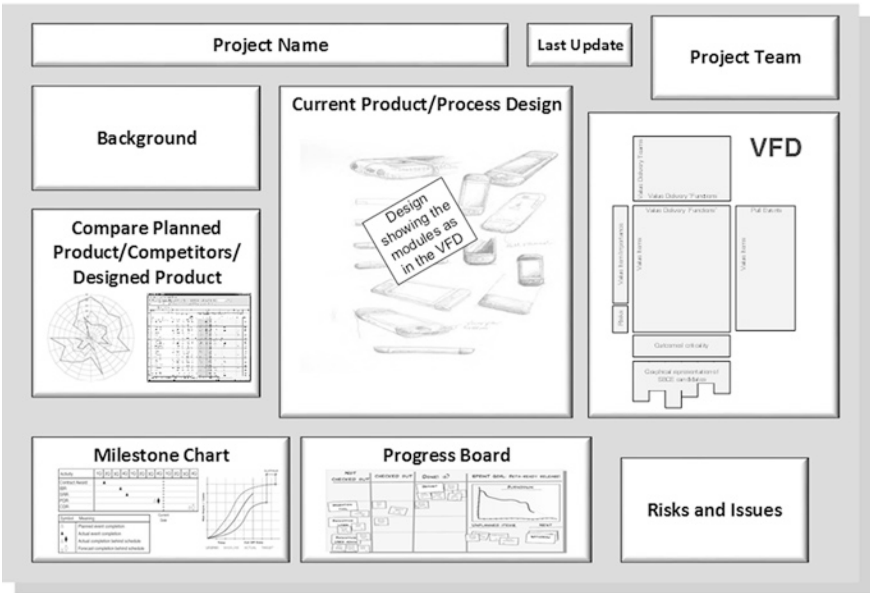

Fig. 9.9 Execution phase visual management board

9.5 A Practical View

At first sight, the VFD looks a bit complex and hard to fill. You must remember, though, that product development is itself a complex task and that the VFD filling is gradual. Consider, for instance, a project management plan with all the related process areas (time, cost, quality, risk, procurement, etc.). Looking at the VFD is like looking at most of those areas at the same time and they are integrated. Indeed, the VFD visually presents and supports answering some key development questions (Fig. 9.10), as presented in Chaps. 10–13:

1. What is the comparative importance of the value items among themselves, considering their relevance to the considered stakeholder set?
2. What value items conflict with each other, thus bringing the need of trade-offs?
3. How am I sure that the functional architecture (product and value chain) is capable of delivering all the pulled value?
4. To what functions should I give more attention once they are more critical (deliver more value and/or carry out more risk)?
5. How can I determine the need of concurrent engineering and who has to work together and when?
6. How can I define a balanced development execution strategy which covers the complete scope and considers all the risks?

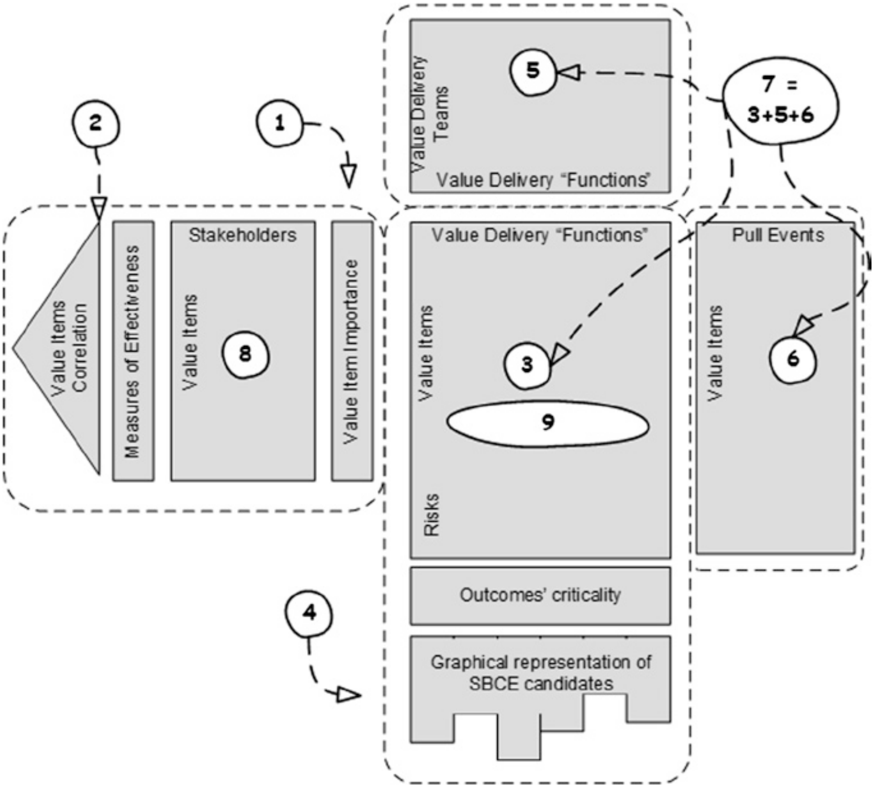

Fig. 9.10 Key answers supported by the VFD

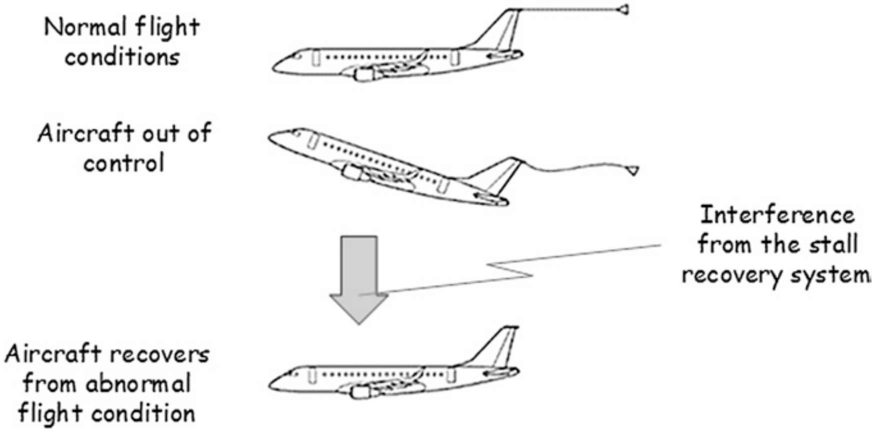

Fig. 9.11 The stall recovery system in action

7. How can I determine the “waste less” set of activities to be performed by all the teams?
8. How can I identify the need of DFX?
9. How can I identify the presence of integrative variables and the need of DTX?

In Chaps. 10–13 we use a product development example that illustrates the PDP being supported both by the VFD and the PDVMB. The data was collected from a finished and successful project which produced a stall recovery system to be used during flight tests and which had the objective of recovering the aircraft to normal flight conditions (Fig. 9.11) in case the pilots lose control of the aircraft while performing flight tests of a prototype aircraft.

References

1. Womack JP, Jones DT (2003) Lean thinking. Free Press, New York
2. Morgan JM, Liker JK (2006) The Toyota product development system. Productivity Press, New York
3. Pessôa MVP (2006) Proposta de um método para planejamento de desenvolvimento enxuto de produtos de engenharia (Doctorate Thesis) Instituto Tecnológico de Aeronáutica: São José dos Campos
4. Pessôa MVP, Loureiro G, Alves JM (2006) A value creation planning method to complex engineering product development. In: Proceedings of 13th ISPE international conference on concurrent engineering, Antibes. Leading the web in concurrent engineering, vol 143. IOS Press, Amsterdan, pp 871–881
5. Cooper GC, Edgett SJ, Kleinschmidt EJ (2001) Portfolio management for new products, 2nd edn. Basic Books, New York
6. Project Management Institute, PMI (2013) The standard for portfolio management, 3rd edn. Project Management Institute, Newton Square