Chapter 13
Execution Phase

By finishing the study phase, the development strategy is set and the LPDO is ready to start detailed engineering, prototyping, and production tooling. During the design and development activities (Fig. 13.1), all the module development teams should produce their deliverables in a fast and synchronized way, according to the sequence of defined pull events. This chapter uses the stall recovery system project example to present a stepwise execution of this phase's activities, where special emphasis is given on how to apply the SBCE and on how to identify and use the integrative design variables from the project's value set.

13.1 Introduction

During the execution phase the product envisioned during the study phase is physically developed and produced. Pull events foster concurrent engineering and sustain the sequence of rapid learning cycles which function as PDCA cycles; thus, making visible any quality problems and supporting knowledge creation. In this context, the execution activities' planning is decentralized, allowing different groups to realize their own plans to achieve the pull events.

We divided the execution phase in two sets of activities: (1) design and development activities; and (2) production/ramp-up activities. In the case of one of a kind and very personalized products production/ramp-up can be considered as part of the design and development activities, once only one product sample is going to be produced.

13.2 Using the Board to Guide the Design and Development Activities

In order to make the information visible and keep the development team synchronized, a PDVMB is also maintained during the Execution Phase. The greatest difference from the PDVMB used during the Study Phase is that the VFD is no

M.V.P. Pessôa and L.G. Trabasso, *The Lean Product Design and Development Journey*, DOI 10.1007/978-3-319-46792-4_13

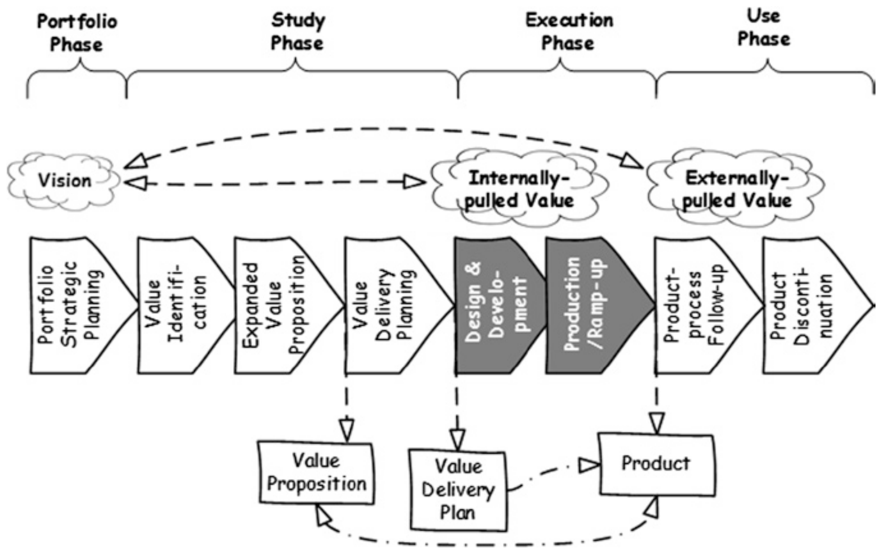

Fig. 13.1 Value proposition activities in the PDP

longer at the centre of the board; now the VFD is a reference of what has been planned, and the current state of the product being developed gets the focus.

The PDVMB filling sequence described in sequence (Fig. 13.2) will guide you during the Production Phase activities.

13.2.1 Setting Project Team

The project team directory must be updated in order to include the key people which will be responsible for actually executing the activities pulled by the several pull events.

In the case of small project groups, you can include the names of all the team members. On larger projects, you would rather include the point of contact from each participating group/area/supplier. Whoever you list here, this person must have the authority to decide and be accountable for his/her decisions.

13.2.2 Keeping the VFD Up-to-Date

Also in the VFD, the value delivered teams must be updated to match the actual project team. You do not need to put the team members' names here, but you must be able to map the value delivery team to the actual members, and vice versa. All the teams on the VFD shall have at least one person listed on the project team directory.

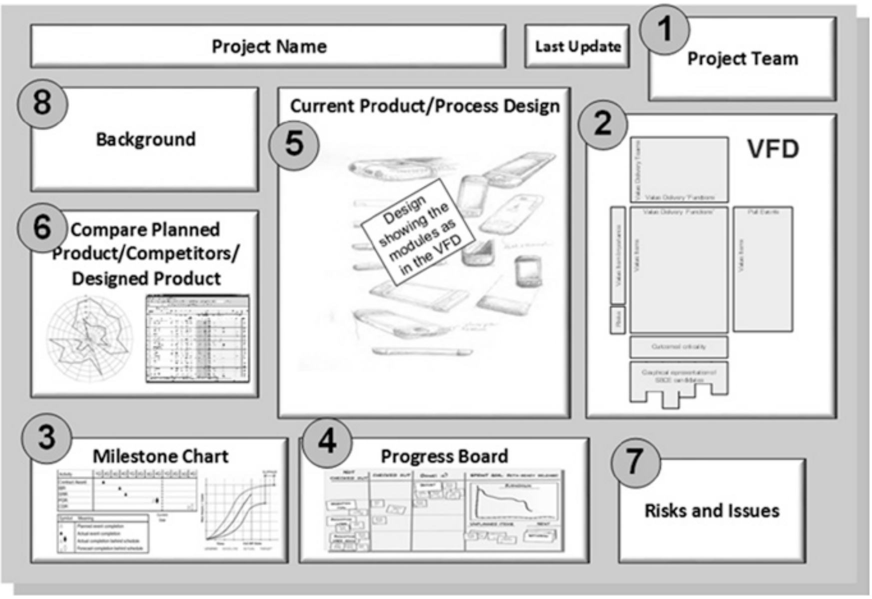

Fig. 13.2 PD visual management board filling sequence

As with any plan, we do not expect that everything will happen exactly as initially set in the VFD. During the execution, the team might find new risks, the value initially pulled might have some changes, new conflicts among value items might become apparent, you might decide to change some pull event's scope, and so forth. As a consequence, the team must always keep the VFD up-to-date.

13.2.3 Navigating Through Milestones

The milestone chart gives a program-level view of the design and development planned work, thus, supporting the meetings at the *obeya*. Each individual team shall have more detailed planning, even using bar charts, to support the execution and control of their work. We'd rather use a milestone chart at the program level in order to reduce wishful thinking which is not the case at the team level since they have much more knowledge to detail their activities and with low wishful thinking.

The minimum set of milestones should include the dates the team expects to have all the previously defined pull events executed. Other milestones, such as expected receiving dates from suppliers, should also be added here. This field is reviewed and updated during each team meeting.

13.2.4 Execution Kanbans

The activities which produce the necessary and sufficient information and materials are pulled from each development team by the pull events, and should be filled into the progress board.

Considering that the pull events network can be quite complex, particularly on large projects, we recommend keeping a separate progress board for each different path in the network (Fig. 13.3). Therefore, for each value item within the scope of a pull event, the pulled activities shall be included on the progress board.

When a value item is in the scope of a pull event, all the teams which help deliver this item have to provide the information and/or materials needed for the event. For example, if some functional value is going to be analyzed during an event, the teams should provide their designs showing how they incorporated the expected value; on the other hand, if the item will be tested, the teams should provide their prototypes for testing.

The activities are then controlled through the board until a new milestone is reached. If some of the activities from the previous milestone were not yet finished, they remain on the board, and the new activities are added.

The defining of the activities to be included in the progress board's backlog is done by answering questions such as: "What are the activities that each team related to the function Fx have to perform in order to deliver the value item Vy which is pulled for analysis by the event Ez?"

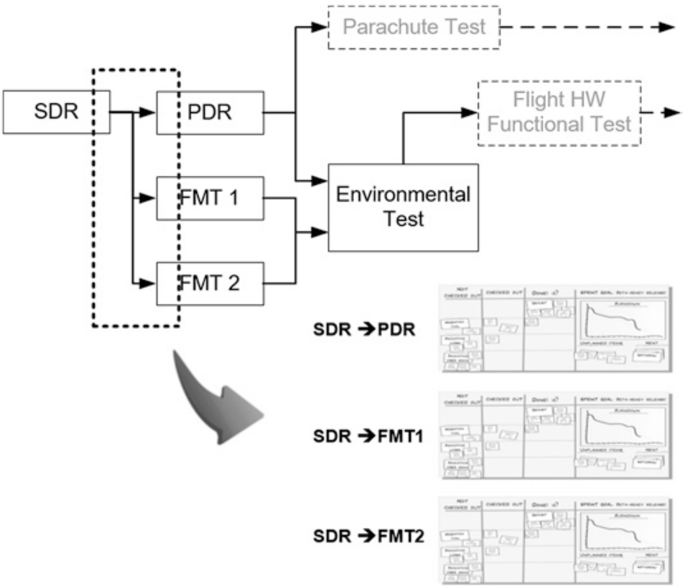

Fig. 13.3 Progress boards

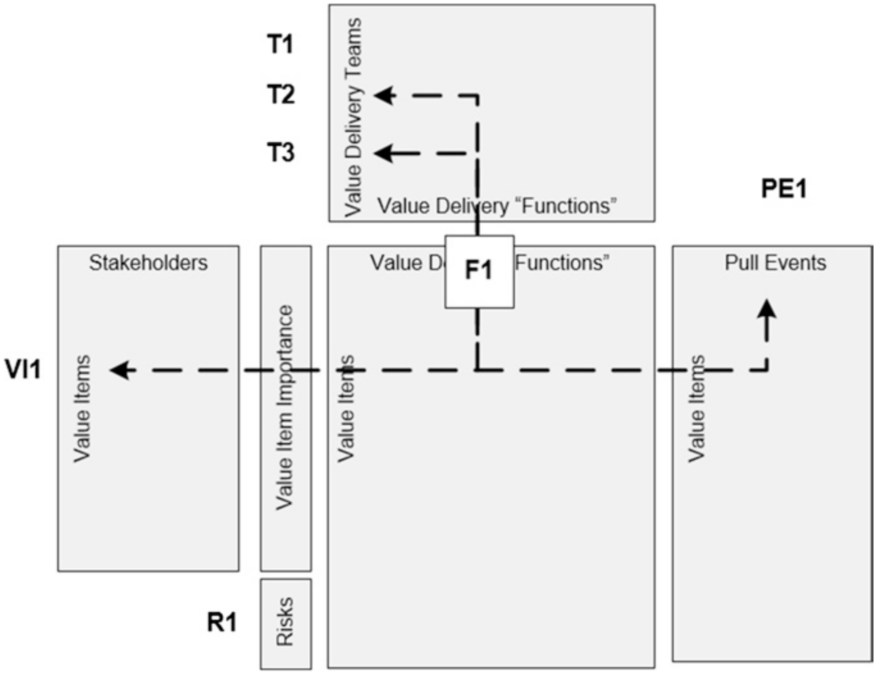

Fig. 13.4 Pulling activities from the teams

Figure 13.4 exemplifies the answering of this question where teams T1, T2, and T3, which are responsible for function F1, must perform the needed activities to deliver the necessary information or materials related to the value item VI1, according to pull event PE1's scope.

Considering the stall recovery system example, Table 13.1 lists the value items' subset that is related to the Parachute Launcher (PCL) development team and how they were included in the scope of the proposal pull event (the development activities are listed in parentheses in Table 13.2). Since this was the first development event, the only verification type used was analysis.

Table 13.2 lists the correspondent PCL development activities pulled by the proposal event. Whenever the method application suggested the use of concurrent engineering, the other participant teams are cited (when other teams are related to the same value item, the needed deliverables are pulled from all of them simultaneously). In the case of the development of multiple alternatives, the activities will be repeated for each alternative.

Table 13.1 Pulled activities example

Value	Value item	PCL	Proposal
1 Realign the aircraft	1.1 Trigger the system	a	
2 Safe and reliable operation	2.2 Have on the ground detection of system unavailability	a	A(1.1)
	2.2 Work when required (Reliability)	a	A(1.2)
	2.3 To not work when not required (Safety)	b	
	2.6 Useful life as…	b	A(1.3)
3 Work on aire rafts A and B	3.1 Mass no bigger than X	a	A(1.4)
	3.2 Interface mechanically with aircrafts A and B	a	A(1.5)
	3.3 Interface electrically with aircrafts A and B	b	A(1.6)
	3.5 Operate under the defined environmental conditions	a	A(1.7)
4 Quick and easy maintenance	4.1 Post deploy repair < X	a	A(1.8)
	4.2 Corrective maintenance time below T sec	a	
	4 3 Support Z years in stock	a	A(1.9)
	4.4 Must have technical documentation	m	
	4.5 Have traceability of the produced units	m	
5 The project must be viable	5.4 Comply with legal requirements	m	
	5.5 Stay within the budget	m	
	5.6 Stay within the deadline	a	
	5.7 Comply with the enterprise's rules	b	
7 Easy to manufacture and test	7.1 Adhere to the design for manufacturing and assembly guidelines	a	
	7.2 Have high rate of reuse of parts, processes, and technologies	m	
	7.3 Have complete and concise product, process and tests documentation	m	
	7.4 Have defined product and process acceptance criteria	a	
	7.6 Have more than one supplier to each procured item or raw material	m	

Table 13.2 Activities pulled from the PCL team

Proposal
(1.1) Determine alternatives "for the on-the-ground detection unavailability system" (TCJ, PCJ, TEQ)
(1.2) Include the PCL data in the system reliability estimate
(1.3) Include the PCL data in the useful life estimate
(1.4) Include the PCL data to the mortar mass estimate (PCH, TCJ, PCJ, LCK)
(1.5) Define the preliminary PCL mechanical interfaces
(1.6) Define the preliminary PCL electrical interfaces
(1.7) Estimate the PCL environmental condition limits (PCH, TCJ, PCJ, LCK)
(1.8) Estimate the time to post deploy repair
(1.9) Estimate maximum time to keep the system in stock

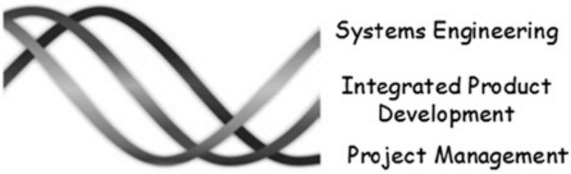

Fig. 13.5 Disciplines working together during the PD execution

13.2.5 Tracking the Current Product/Process Design

Here, the product evolution through the execution of the planned design and development activities is presented, where the product will evolve from functional design, to detailed design, and then prototype.

This is the moment when system engineering, integrated product development and project management are going to walk hand-to-hand (Fig. 13.5). The VFD and SBCE should be applied in each of the system's engineering phases as the design and development evolves (see Chap. 9).

Whenever a prototype has been built, pictures from it shall be included if it's not feasible to have the prototype itself at the meetings. Particular caution has to be taken when moving from value items to detailed requirements; the correct traceability among them will guarantee that value alignment is kept. We strongly recommend the use of and requirements management software to keep track of this detailing and allocation and requirements allocation into products subsystems/modules.

The technical discussions among the team members will have the current product/process design or prototype as the main point of reference to solve issues. For instance, on the one hand, a person from maintenance would suggest that symmetry should be chosen for a given part geometry. On the other hand, a person from manufacturing would argue that asymmetry ought to be chosen for the same part in order to ease the setup of the lathe. Indeed, this discussion might go on with other people and technical areas. This is the true sign of interaction and concurrent engineering. The final version of the product is then defined by the development project leader, and shall accommodate in the best and balanced possible way the requirements of the whole product lifecycle.

13.2.6 Mirror, Mirror on the Wall, Who Is the Fairest of Them All?

During the whole execution phase, value is embedded into the product, and keeping track of this evolution, particularly when SBCE's alternatives are canceled, is an important indication of the development project's evolution.

We suggest keeping a radar chart where you can compare both the planned and actual product as well as the actual version of your competitors. This will also give you a good measurement for the business case of your upcoming product.

13.2.7 Bumps on the Miles Ahead

It is not uncommon that development teams, once they have started the execution, forget to keep making risk management; they tend to focus on the present, solving actual problems, and not acting proactively on identifying new risks if mitigating activities are not defined in the initial plan.

Risk management is an ongoing activity, whenever a risk or issue is identified, mitigated, or solved, this field is revisited and updated. You should invest some time during the periodical team meetings to try identifying which new risks are present and/or which previously identified risks have been surpassed.

13.2.8 Background and Change Management

Keeping the background field in the *obeya* helps with change management decisions. Whenever a new idea is brought into discussion, you must first check if it is compatible with the original product vision and then how it fits into the value items set.

If a new idea/pulled value/requirement is not compatible with the vision you either have to discard it or negotiate the changing of the vision.

13.3 Production and Ramp-Up Activities

During execution, all lean, quality, and design tools and techniques are applied to guarantee a *jidoka*-like and just-in-time-like development project (Fig. 13.6) [1]. By making information visible, it is easier to identify problems and waste. By stopping the chain of development events and investing time to understand the problems/waste root causes, you simultaneously guarantee solid/robust project deliverables and products and promote learning. By using techniques like 5S and Kanban, you both limit the work in process (WIP) and reduce the chance of waste occurrence.

The 5S is a quality technique that supports the organization and discipline at the workspace and reduces waste, where each “S” means:

- **Seiketsu** (清潔): Standardizing the best practices in the work area and maintaining everything in order and according to its standard.
- **Seiri** (整理): Organizing, making work easier by eliminating obstacles.
- **Seiso** (清掃): Keeping the work place clean and pleasing to work in.
- **Seiton** (整頓): Straighten, tidying up, arranging all necessary items so they can be easily selected for use.
- **Shitsuke** (躾): Training, sustaining and keeping discipline.

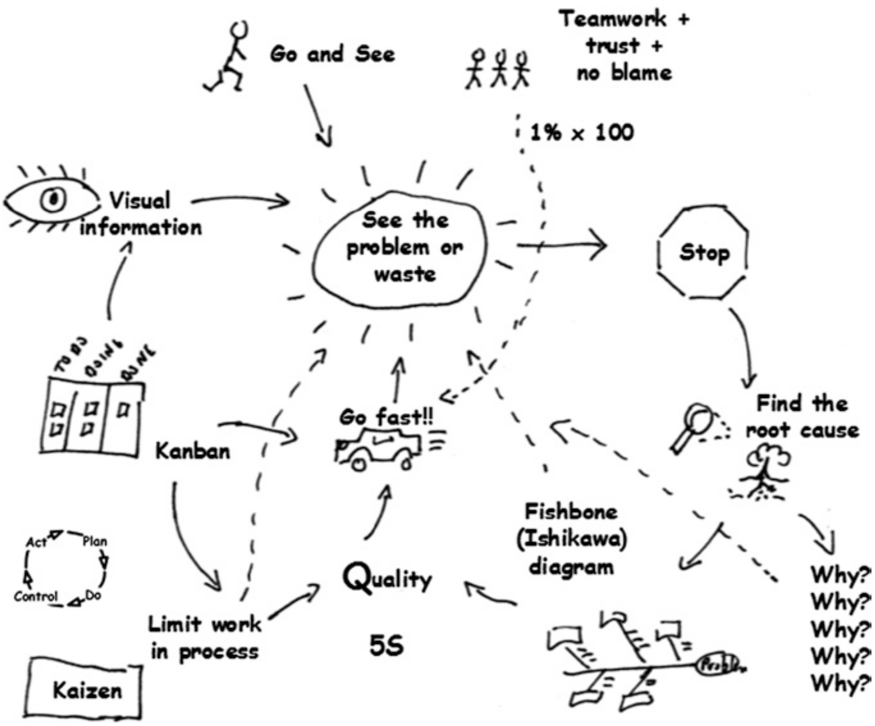

Fig. 13.6 Lean tools and techniques during production

13.4 A Practical View: Which DFX and/or DTX should I Use?

We suggest that you take advantage of the VFD in order to identify the need of DTX and/or DFX design tools. Since there is a fair amount of possible DTX and DFX techniques, it is not so easy to identify which of them would be useful during a particular project. In order to do that, we use the following rule of thumb.

For the DFX identification, look at the VFD's value identification matrix and ask, for each pulled value, if there is a related DFX. For DTX identification, look at the VFD's rework avoidance sub-matrix and check which value items are related to most, if not all, value delivery functions. These value items usually include an integrative design variable and are candidates to have an associated DTX.

Taking the stall recovery system example, Table 13.3 shows the high-level value pulled by the stakeholders and the related DFX. Similarly, Table 13.4 shows the value items associated to integrative design variables and their corresponding DTX tools.

Table 13.3 Stall recovery system's value items and DFX

Value	DFX
1 Realign the aircraft	The breadth of this value requires a deployment action prior to the DFX identification. See below.
2 Safe and reliable operation	Design for Safety
3 Work on aircrafts X and Y	Design for Modularity
4 Fast and easy maintenance	Design for Services Design for Modularity
5 Be viable	Design for Certification, Design to Cost
6 Provide learning	KBE—Knowledge Based Engineering
7 Be adequate to manufacturing and tests	Design for Manufacturing and Assembly and Design for Testing

Table 13.4 Stall recovery system's value items and DTX

Integrative Design Variable associated to Value items	DTX
1 Weight	Set a upper bound for the weight and use Design to Weight
2 Energy consumption	Set a upper bound for energy consumption and use Design to Net Power
3 Cost	Set a upper bound for cost and use Design to Cost

The value items deployed from the value [Realign the aircraft] are <Quick response to triggering>, <Return to normal flight attitudes>, <Eliminate aerodynamics effects on the aircraft after use> and < Eliminate electrical effects on the aircraft after use>. Each value item is associated to a system designed to deliver it. Find the system closet to a physical embodiment and apply the DFMA® tool to it. In this example, we chose the triggering mechanism to start with. The establishment of the minimum part count is always a good move to start the dialogue between the design and manufacturing technical areas.

Be aware of the design tradeoffs. Fear not! It's a rich experience for the product development team. The requirements for the several DFX techniques are—usually—in conflict to each other. The ideal product configuration for assembling might be the opposite for servicing the product. All people from the IPD design team should be committed to obtaining the best possible balanced results for the product, even if that means giving away some of his/her technical area expectations. The team coordinator has to assure that the final configuration of the product best balances all the lifecycle's requirements.

Still considering the stall recovery system example, the value items of having low weight, low energy consumption and low cost affect most of the value delivery functions, which indicates an opportunity to use DTX (Table 13.4); where, in this case, X means weight, net power and cost. Consequently, the functions should be balanced in these dimensions in order to optimize the product as a whole, which in this case can be achieved by the use of the Design to Weight (DTW), Design to Net Power (DTNP) and Design to Cost (DTC) directives.

In Chap. 2 we detailed the DTC technique, which can be used as template for DTW and DTNP. Note that your particular project may take benefit from using other integrative design variables, which were not mentioned in this book, but can be identified in the same way we presented here. You can then elaborate your own technical management cockpit (see Fig. 2.21) and place it at the Obeya. This cockpit will support you throughout your product design and development journey.

Reference

1. Blanchard BS, Fabrycky WJ (1988) Systems engineering and analysis, 3rd edn. Prentice Hall, New Jersey