James Cadle and Donald Yeates

fifth edition

Project Management for Information Systems

Pearson

Project Management for Information Systems

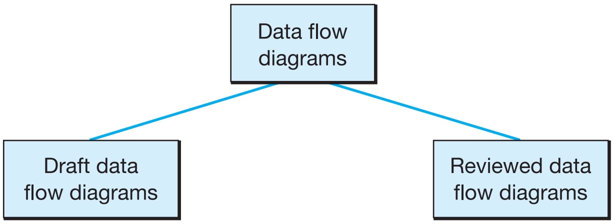

Figure 8.8 **Product breakdown structure: bottom level**

Finally, the products may be subdivided one more time as shown in Figure 8.8.

At the bottom level, we find an individual product for which we can write a **product description**. We discuss this in more detail in section 8.4.

Once we have completed our PBS we have a complete list of the products that the project will develop. Some of these will be final, deliverable products such as the feasibility study report. Others will be transitional products, created on the way to our final product but not deliverable. A good example of this might be the minutes of weekly team checkpoint meetings.

With our list of products, we can now consider the work we will need to do to create the products. PRINCE2® uses a technique known as a **product flow diagram** for this. The idea is simple enough: we look at the products in relation to each other and consider how one product is transformed into another. For example, let us say we have conducted our interviews for the stock control feasibility study. We have our interview notes and we need to transform them into entries in our requirements catalogue and into data flow diagrams (DFDs). We can represent this as a product flow, shown in Figure 8.9.

In this diagram we can see that our interview notes are transformed into requirements catalogue entries by the activity *Create requirements catalogue*. Similarly, the draft DFDs are transformed by the activity *Review DFDs* into a set of agreed DFDs.

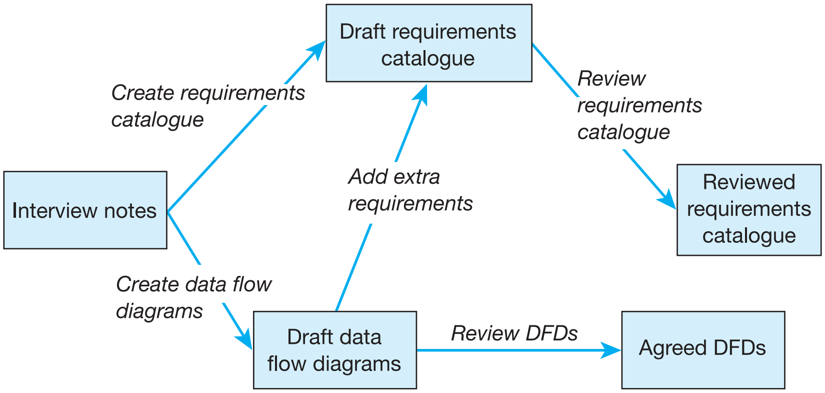
Draft requirements catalogue
Create requirements catalogue
Review requirements catalogue
Reviewed requirements catalogue
Interview notes
Add extra requirements
Create data flow diagrams
Draft data flow diagrams
Review DFDs
Agreed DFDs

Figure 8.9 **PRINCE2® product flow diagram**

The product flow diagram provides two important inputs to the planning process:

- A set of activities, for which we will need to estimate effort.
- An understanding of the dependencies between activities.

We discuss dependencies more fully in section 8.5. Before moving on, however, there are a couple more points to make in relation to PRINCE2® and products. The PRINCE2® manual and many of the textbooks on PRINCE2® provide outline PBSs for projects. These are quite detailed in respect of the management/quality products, less so for the specialist products – necessarily since all projects are different. However, when PRINCE2® is used in association with the structured systems analysis and design method (SSADM), then SSADM provides a more detailed PBS for its specialist products, as well as detailed product descriptions. Thus, a project manager using both PRINCE2® and SSADM will have available a very good general product breakdown structure from which to develop a project-specific structure. Even if you are not using SSADM, the SSADM PBS provides some guidance on the types of product that you are likely to develop in any IT project.

8.4 Product descriptions and work packages

8.4.1 Product descriptions

Once the bottom level of the PBS has been reached, we are in a position to create a description of each project. There are two main reasons for doing this:

- It encourages us to think through *exactly* what we want this product for, what form it should take and what quality and completion criteria should apply to it. Once we have a proper understanding of these things, we will be in a better position to develop accurate estimates for the development of the product.
- It provides, as it were, a detailed specification of work for the project team member who will ultimately be asked to develop it, leaving them in no doubt about what is required from them.

Each product description should contain, as a minimum, the following sections:

- *Purpose*. Why the product is required.
- *Composition*. What are the elements that make up the product. As we have seen, we develop our understanding of products through a process of decomposition but, at some stage, we reach the point where it is not sensible to decompose further as the product represents a coherent piece of work. However, even then, a product may have sub-components; a report, for instance, will contain a number of sections.
- *Derivation*. The product flow diagram, which we met in section 8.3, illustrated the principle that a project consists, in effect, of a number of transitions,

where one product is transformed into another. So, for each product, it is useful to record which previous products it will be based on or, perhaps, where the information needed to develop it will come from.

- *Quality/completion criteria*. This part of the product description is very important. The project manager (or whoever is planning the project) really needs to think through what is wanted, to what standard; and the person who is going to be asked to do the work needs to know exactly what is expected. Where a project is being undertaken within a defined set of procedures and standards, then often the product descriptions need only to make cross-references to these standards; otherwise the quality/completion criteria will have to be defined specifically for each product.

In addition to the above, it can be useful to define the following:

- *Format*. What format should the finished product take. Again, if there are standards or templates available, reference can be made to these.
- *Related products*. Often, there are products that need to be developed together, or at least kept in step with each other and, if so, it is worth recording the fact here. (We shall have more to say about this in the next section about work packages.)
- *Review methods*. It is a good policy to decide in advance what method/s will be used to review the finished product, for example management review or formal inspection. There is more information on possible review methods in Chapter 11.

Clearly, creating all the product descriptions is not a trivial task but, at the end of it, the project manager will have a much better understanding of what is involved in the project; and the project team members will have a clear view of what is expected of them.

If a project is being undertaken using PRINCE2®, then the PRINCE2® manual contains detailed product descriptions for the management/quality products and these definitions can either be copied into the relevant product descriptions, or cross-references can be made to the manual.

8.4.2 Work packages

At some point, the project manager has to assign specific pieces of work – **work packages** – to individuals. A very simple, and non-bureaucratic, way of doing this is to treat each product as a work package. All that then needs to be done is to add three sections to the product description:

- *Effort estimated*. How much effort has been estimated to produce the product.
- *Date required*. A date when the finished product is needed.
- *Allocated to*. The name of the person who has been tasked with developing the product.

However, sometimes it makes more sense to group a set of products together into a work package and this is illustrated in Figure 8.10.

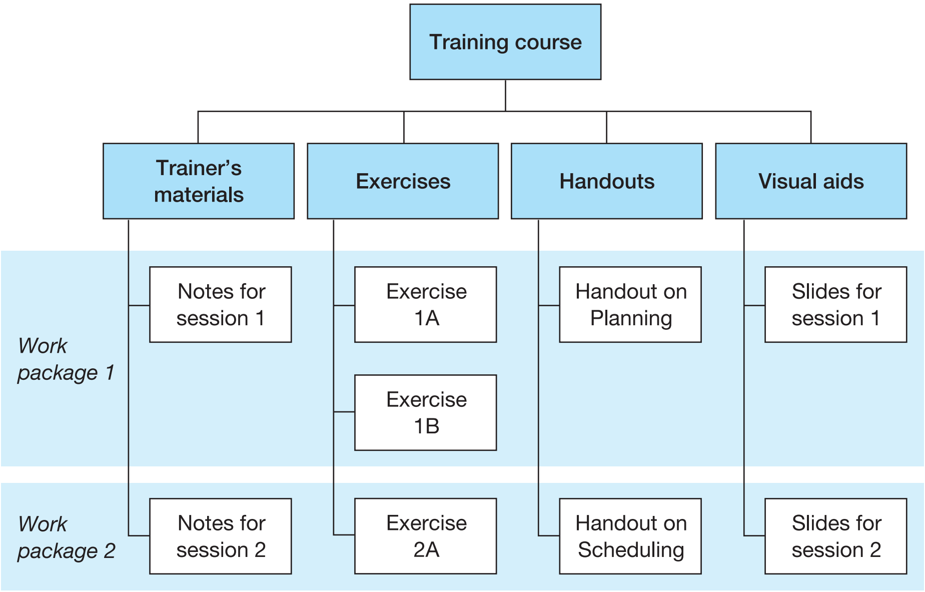

Figure 8.10 **Work packages for a training course**

In Figure 8.10, we show some of the products identified for developing a training course. We could treat each product as an independent work package but, in fact, it would be much better for the person developing each session of the course to take complete responsibility for all the products for that session; these include the trainer's notes, the exercise, any supporting handouts and the audio-visual materials as well. So, the products of each session have been grouped into work packages that can each be assigned to one person – with an effort estimate and a deadline for completion.

8.4.3 Work package assignment to roles or individuals

Once the work packages have been identified, the project manager can give some thought as to who is going to do what in respect of each: who will produce it, who review it, who approve it and so on. In the early stages of planning, assignments will probably have to be by role, as the individuals filling each role may not yet have been identified.

An excellent way of summarizing who does what is to use a 'linear responsibility chart' as illustrated in Figure 8.11.

A linear responsibility chart is a two-dimensional grid with the list of products or work packages on one axis, and the roles identified in the project organization chart on the other. Where each work package and role intersect, we record that role's responsibilities.

In Figure 8.11, we present two possible schemes for defining responsibilities. In the first, the role is said to be:

- Responsible: For actually creating the work package.
- Accountable: Managing creation of the work package.

R = Responsible
A = Accountable
C = Consultation
I = Information

Product/work package breakdown	Organization breakdown							
	Project sponsor	Project manager	Analysis team leader	Chief designer	Development manager	Test manager	Project support assistant	Senior user
Interview notes	I	A	R	I				C
Requirements catalogue	I	A	R	I				C
Use case diagram	I	A	R	I	I			C
Package review	I	A	R	I	I	I		I
Report text	I	A	R	I				I
Report illustrations	I	A	R					I
Report appendices	I	A	R	I				I

OR, could use
I = Initiation
E = Execution
A = Approval
C = Consultation
S = Supervision

Figure 8.11 **Linear responsibility chart**

- Consultation: Will provide information for creating the work package.
- Information: Will be kept informed about progress.

In the second scheme, we use five categories to define the responsibilities:

- Initiation: Starts the process off.
- Execution: Carries out the work.
- Approval: Reviews the results.
- Consultation: Is consulted during the work (and/or provides information needed to carry it out).
- Supervision: Manages the work.

The initial assignment of work packages to roles may have to be revisited later in the planning process, when the workloads on individuals, and their capabilities, are better understood.

8.5 Understanding dependencies

Dependencies are fundamental to planning a project and, later, in understanding the effects of any problems encountered. Yet many IS project managers do not conduct a proper analysis of project dependencies, arguing that for most IS projects they are obvious: one has to analyse a requirement before specifying a solution and write a program before testing it. Although this is true enough in this case, these arguments apply only to simple projects, with a few people involved in them. Where – as is often the case nowadays – there are several teams at work, each of them developing a specific part of a system, the need to understand the often complex dependencies becomes paramount.

In essence, understanding dependencies is simple. If activity B can begin only when activity A is complete, then we have a dependency. In our example project, we can begin to develop DFDs only once we have conducted our interviews. However, dependencies are often more complex than this. Do we, for example, need to have completed all of our interviews before we start any of our DFDs? Probably not. Quite possibly, we could produce a high-level DFD after we have interviewed the managing director and then develop it further as other interviews are completed. With multiple teams operating, dependencies become more complex still as we need to know exactly which components from one team are required before that, or another, team can start on something else.

We can analyse dependencies using a **network diagram**, also known as a dependency diagram or **PERT chart**. A network diagram for our example project is shown in Figure 8.12.

This diagram has been drawn using a format known as *activity-on-arrow*, which means that the lines represent project tasks and the circles the connections between tasks. From the diagram, we can read the following:

- Once the project starts, we have two activities – *Conduct interviews* and *Investigate other systems* – that can proceed in parallel.
- But the results of both activities have to be brought together before we can start the next three activities – *Analyse requirements*, *Investigate packages* and *Investigate hardware*. We show this bringing together by using a 'dummy' activity, one with zero duration, indicated by the dotted line.
- The three activities are then brought together – again using two dummy activities – before we can start the last task of our project, *Produce report*.

This very simple structure has already told us one important thing about our project: we can use more than one person on it if we wish, working independently until such time as their work must be brought together.

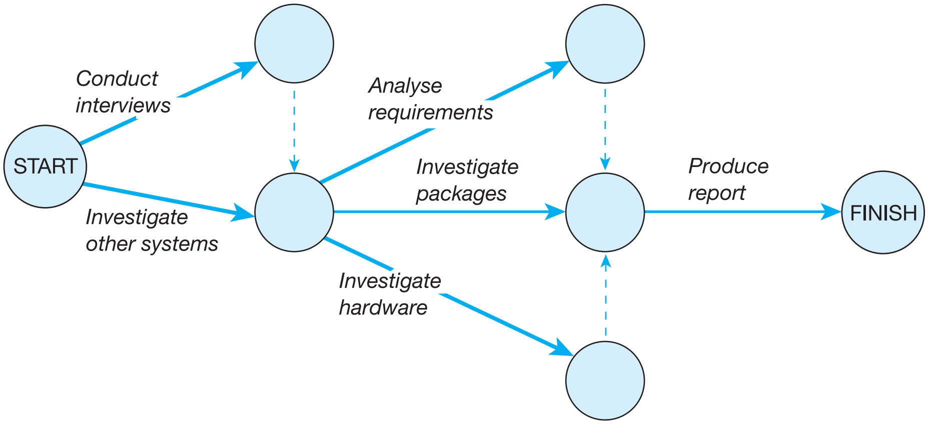

Figure 8.12 **Network diagram (activity-on-arrow format)**