

DPC In short

Introduction

DPC, Dynamic Project Control, is a comprehensive method for the planning and monitoring of projects, large and small.

DPC is built on a sound theoretical basis. You can find the theoretical material on this site: www.jptollenboom.com.

Also visit this blog for regular updates and comments: www.jptollenboom.org.

In this short introduction we will highlight the 5 basic principles:

1. To be able to drive a project we need **feedback** information on its behaviour
2. To obtain feedback information we need to **monitor** - to track - the physical progress of the project
3. To be able to track a project we need a **planning**
4. The process of **constructing** a planning and things to avoid
5. The process of **monitoring** and the best practices

The Five Basic Principles

1. To be able to drive a project we need feedback information on its behaviour

We want to drive the project according to the well-known principles of feed-back control, as applied in process control theory.

Therefor, instead of simply giving instructions to those in charge of executing part of the project and / or specific activities (tasks), we will also measure the physical progress and report it in such a way that we can easily understand how the project is doing: whether it progresses according to plan or too slowly. With this information we will then make decisions about what to do to accelerate progress if needed.

Plan-Do-Check-Action: this is the fundamental *controlling and action* loop. It is schematically depicted in the next figure (Slide 2)

2. To obtain feedback information we need to monitor the physical progress of the project

The outcome of any project must be an **observable** object:

1. A book
2. A car
3. A plane
4. A machine

5. A small building
6. A huge building
7. A power plant
8. A movie
9. etc

When we want to track (to monitor) a project, we only need to measure the **growth** of the object at regular time intervals: it's like looking at the growth of a flower. This is called measuring the physical progress of the project.

The "growth-curve" as shown in slide 3, is usually called the "S-curve" because its shape looks like the character "S". This curve is also called the "progress curve".

Progress curves can have all kinds of shapes, not necessarily the "S" shape. Still, they are always referred to as the "S-curve".

When we monitor the progress we are interested in two things:

1. The present progress value - **how much** has the flower grown so far
2. The progress rate, or progress speed - **how fast** is the object growing

Both quantities are shown on slide 4.

Expressing the quantities

Progress is expressed in percentage complete, short %. It is a value that ranges from 0% (nothing is done) to 100% (the object is completed). Values larger than 100% never appear.

Progress rate is expressed in % per day; it is a velocity, a speed, a rate.

3. To be able to track a project we need a planning

The planning of a project is the complete list of all activities that are needed to complete the project. For every activity there is a start date and a finish date.

The planning is best constructed using a *planning software*. Examples are:

1. Smartsheet
2. ProjectLibre
3. Microsoft Project
4. Omniplan

In its most complete form, the planning will also have resources allocated to the activities. Activities or tasks, will also have *weight factors*. The weight factor of an activity is a measure of its importance in the total project. Very important tasks will have large weights.

The most used format to display a planning is the *Gantt chart* (named after its inventor). An example is shown in slide 5.

4. The process of constructing a planning and things to avoid

To construct the project's planning do this:

1. Define the complete list of activities.

Rule: the list must be exhaustive, what is not in the list does not exist.

2. For every activity define a duration in days.

Durations can be estimated or copied from previous experience. Sometimes it is possible to compute durations. If you really don't know, try using the formula shown on slide 6.

3. Allocate weights to the tasks
4. Group the task in groups of similar nature or per fase of the project.

Rule: The structure so created must be a **tree**. A tree can only have **one root**.

Best practice for beginners: limit the number of levels to 3, maximum 4.

See slide 7 for examples.

5. Now you have to define the dependancies: define which task must precede which other tasks. This is also called defining the **logics** of the project.

This is an important aspect of the construction of the planning. This will ultimately define the total time to completion, or the total duration of the project. Slide 8 shows the most common cases. More details will be giving during exercises in the class.

Best practice: avoid defining too many dependancies (links), try to keep tasks in sequence.

Best practice: if the total project duration is too long, try this:

- shorten some duration, when you have been too conservative.
- introduce parallelisation of tasks (full or partial).

5. The process of monitoring and the best practices

Now the planning has been constructed the only thing that remains to do is to run the project and to track its progress. Some rules:

1. If you failed to plan, you planned to fail.
2. Plan the projects, execute the plan.

To monitor the progress of the project, this is what you have to do:

1. Set the % complete values for all the active tasks
2. Compute aggregate % values. Aggregate values are the % complete values of the task groups as defined by the planning structure;
3. Generate progress curves for every group of tasks, every summary of the planning
4. Analyse the results, compare observed progress with scheduled progress
5. Act to correct the course of action if needed

This loop should be repeated on a regular basis, eg. every week, every two weeks, sometimes every day when the project moves very fast.

When you use the DPC engine, the report generation, scheduled S-curves and observed S-curves, is done completely automatically.

Slide 9 shows some typical DPC tools and situations.

slide 10 lists what to do in order to recover delays.

A New Project Culture

The reason to use DPC is not only to substantially increase the level of control over the projects, and thus the probability of success, but also to introduce a whole new project culture.

Slide 11 shows two styles of track:

1. A, the excellent track
2. B, the bad track, also known as the "Student syndrome".

A is excellent because it has these characteristics:

1. first part: start slow
2. second part, reasonable fast progress
3. last part: slow finish.

This track shows total success, because:

1. the project is completed in time
2. the last part is slow, we can be sure that the scope of work will be 100% and that the quality will be as requested.

Track B shows us, that although the project is completed in time, the last part was rushed to the end. This tells us that the scope of work may not be 100% fulfilled, and almost certainly that the quality of the end product will be bad, just because everything had to be rushed to the end.

This "student syndrome" type of project is a very bad, very costly habit, that systematically produces bad end results.

DPC can help in getting rid of this bad habit.

Rule: follow the "dream track" (slide 12)