**School of Electronic &
Electrical Engineering**
FACULTY OF ENGINEERING

UNIVERSITY OF LEEDS

Project Management: Scheduling

Time: A Limited Resource

There are essentially two types of project resource: Human resources… and all other resources! Human resources are primarily concerned with availability of people and their *time*; other resources include financial resources and any 'assumed' resources (equipment, space, services, etc). Of course, in commercial projects, human resources and financial resources are linked (people need to be available *and* paid).

As a project manager, you need to estimate the resources (money and time) required to fulfil your project objectives. If you are working in a duration-driven, resource-limited situation, this will directly affect the scope of your project. (With fixed duration and limited resource, you can only achieve a certain amount of output.) In a duration-driven, resource-limited project, it is necessary to review your scope, *after* initial estimates of the resources needed (especially time) are made.

> ***Thought:*** Don't be pressurised into committing to an unrealistic project scope, and don't be tempted to promise something which is not achievable – this is *not* good project management. Your Supervisor may make sweeping statements about what he/she thinks can be achieved – be willing to enter into a discussion if you disagree – it's *your* project!

Project Scheduling

Project Scheduling is the most difficult phase of project planning because it requires estimates of the time required to perform tasks. Some of these tasks you may not have done before, so you will have little means to estimate the time!

Scheduling is about looking at the way time and resources are applied in your project. It is about estimating the time individual tasks or phases will take and determining when each one should start

and finish. For each task, it takes into consideration what the task *dependencies* are: that is, what other tasks (or resources) does each task depend on before it can begin?

Since you are largely working on your own in your project, the process is much simpler, often just sequential, and relatively few tasks will be concurrent. Nevertheless, it is worth thinking very carefully about the time allocated to each task, and what needs to happen before a particular task can start. This is particularly important when the only time resource is your own! Tasks may not depend on each other, but *you* cannot work 100% of the time on two tasks simultaneously. Sometimes you are forced to introduce a dependency on previous tasks simply to form a sequence of work that you can do on your own.

> ***Thought:*** Student projects sometimes fail to achieve very much. When asked the reasons, it is often apparent that there was lots of 'dead time' in the project – time when nothing happened at all. The reason? 'I was waiting for 'x' to be delivered or 'I was waiting for the workshop to machine 'y' for me. This is inexcusable and very poor project management. You cannot afford for this to happen!

The first step in project scheduling is to de-compose the work required (to fulfil the project requirements) into manageable and *identifiable* tasks.

This process is called a 'Work-Breakdown-Structure'.

Work-Breakdown-Structure (WBS)

All work that is to be done as part of your project should be included in your WBS. If work is not included in your WBS, consider it *outside the remit of your project!* Each task will have an associated time estimate, required resources and may - or may not - depend on other tasks before it can be started.

There are two issues in creating a WBS. First, how detailed should it be? Second, how can you estimate the time required for each task?

The level of detail should be *appropriate* to how often you will *monitor* your progress (see *Schedule Monitoring* below). If you intend to monitor progress fortnightly (for example) then your WBS should be based on tasks which take typically 4-5 days (i.e. *less* time than your monitoring time frame).

Once you have a list of tasks, the next step is to estimate the time for each one. This is by far the most difficult step. Do you *really* know how long it will take to design circuit 'x'? Have you done it before?

Have you done any preliminary design work to investigate how to do it? But isn't that part of the project itself?! These questions need careful thought.

Estimates of time are usually too low – so be careful when saying 'it will take a week to design this circuit and get it working'. Will it? And how many hours in the week will you *really* spend working on it?

Some reasons why estimates are often wrong:

- it takes longer to *understand* a new topic than you imagined
- it takes longer to get your circuit design/program code/simulation/experiment working…
- you discover something extra needs to be done that you had not thought about before

There's nothing wrong with asking someone with experience (e.g. your Supervisor!) how long they think something may take. In large technical projects, a good Project Manager will always consult with team members - people with experience and expertise.

> ***Thought:*** Accept that some estimates may be way off – so be cautious with *all* the estimates; that way, your overall time may be about right! Make realistic estimates. *You* are the one doing the work. The question is not how long should it take an experienced engineer – but how will it take *you?* Of course, this is not an excuse to achieve little (unless you want a *little* mark!).

Tasks may be grouped into larger 'work-packages' to make it easy to take an overview of your WBS. For example, if you have a project involving some circuit design, consider identifying this as a 'package' and then sub-dividing the design work into functional blocks for each part. Using a hierarchical task numbering system will help to keep track of the tasks involved in a project.

Milestones can be defined as additional 'tasks' with no work associated with them. This makes them easy to spot in the sequence of tasks.

Finally, you need to identify the sequence of tasks. This will involve identifying any dependencies (tasks which must be preceded by other tasks) and any fixed dates in your schedule (e.g. report deadlines!).

What tasks *could* be done concurrently? (This does not necessarily mean they *should* be done concurrently, particularly if *you* are the only human resource that will work on the project.) Which tasks depend on other tasks – or on external events - before they can be started?

Getting the WBS right is the key to good project scheduling. The advantages of a good WBS are that risks are identified more effectively, better estimates of total project duration are made, and there will be better overall control and monitoring of the project 'pieces'.

Critical Path

A completed WBS can be analysed to show the critical path between start and finish. The critical path shows the shortest time (taking all the *dependencies* into account) that can elapse from project start to project finish. If this time is close to an overall fixed duration (for duration-driven projects) you may have too much risk of not completing the work in the given time frame. In this case, you may need to consider changing some of the scope or requirements in your project. Tasks which may run concurrently will not contribute to the critical path, but linked dependent tasks will. Scheduling software such as MS Project is useful for showing this type of information.

Schedule Monitoring

Many projects end up with disappointing results simply because of a lack of *pro-active* monitoring of time and progress. It is good practice to monitor and reflect on your progress at regular intervals. A good Project Manager will adopt a 'Standard Reporting Period' and update the progress of tasks at these regular intervals.

It is therefore useful to have your tasks broken down in the WBS on a timescale that is *shorter* than the Standard Reporting Period. This will give the reporting period more significance, making it less driven by individual tasks, and showing a more overall picture of progress. So, if you decide to adopt a weekly standard reporting period (probably a good choice for your project) then your individual tasks should have typical associated times of not more than (say) 2-3 days. This means

you would update your task progress fortnightly, and your WBS would contain tasks of typically 2-3 days duration (though specified in hours, rather than days). This is not a rigid rule, but you should consider whether it is appropriate for your project.

Thought: Saying that a particular task is 90% finished doesn't mean that it is actually near to completion. Instead, report on what is *actually* 100% finished. Projects have been known to have lots of tasks that are 90% finished, creating the false impression that the whole project is nearly done. Your Supervisor will rightly be sceptical about progress reports, unless particular tasks are 100% finished.

MS Project

MS Project is a software application available in the School. It is designed specifically to help with project scheduling, monitoring and progress reporting. It is *not* 'project management' software. MS Project – or any other similar software tool – will not *manage* anything. It is *you* that manages the project! And as we discussed, there is much more to project management than scheduling.

MS Project is 'narrow focus' software, meaning that it is designed to do a very specific task (unlike Excel, for example, which is deliberately designed to be very flexible). It is worth noting before you even begin with MS Project, that this software is far more comprehensive than you are ever likely to need. Like a powerful CAD package, that does not mean that you can't use it to solve relatively simple problems, but be aware that MS Project does much more than you need. Don't get too stuck in the details of what the software does! If you find it appropriate, use it to do the basic scheduling for your project needs.

The following notes are not a 'manual' for MS Project (there is plenty of on-line help/'wizards' etc). What follows is a summary of some key 'variables' and what they mean.

MS Project Basic Functions

MS Project is a database, linking time with tasks and resources. The link is made using three things:

- Information you tell the software about your tasks (dependencies, hours required, etc)
- Resources (principally, people and their available time, which MS project calls Effort Units)
- The fundamental equation of Duration, Work and Effort (see below)

MS project can also account for staff costs, and other resources; we shall not consider this further.

Fundamentally, MS Project solves the equation:

$$\textbf{\textit{Duration}} = \frac{\textbf{\textit{Work}}}{\textbf{\textit{Effort}}}$$

Understanding this is crucial to using MS Project. The inputs to this equation are the tasks and the effort assigned to each task. The equation is solved by the constraints you impose on each task. For most purposes, you will specify tasks as having a fixed amount of Work associated with them. Thus, once you have assigned Effort to a task (i.e. people's time) the overall Duration is just calculated. Using MS Project in this way is the most obvious and intuitive.

Unfortunately, the program may be used in several other ways which can sometimes cause confusion. For example, you can specify a task as having a Fixed Duration, regardless of the Effort assigned to it. In which case, the amount of Work done on the task *will change* as the Effort changes. This may be perfectly legitimate in some cases, but is not likely to be that useful to you. (An example might be providing supervision of a large team of workers; if several supervisors are free, you could assign them to the same task, and the total amount of 'work done' on the *same task* increases.)

Since MS Project is essentially a database program, there are a large number of different ways to view the core data associated with the tasks and resources. Experiment!

Some important MS Project definitions:

Duration: This is the time a task will *actually take to complete*, usually specified in days.
Work: The time that is *required* to complete a task (usually specified in hours).
Effort: The percentage of a person's *standard* working day; i.e. the daily *Effort* applied to the task.

These parameters all have the dimensions of *time*…

A suggested workflow with MS Project is as follows:

1. Decide and set the default *type* of scheduling (will tasks be fixed-duration, fixed-work, etc).
2. Make 'global' settings (Start date, Non-working dates, standard work-day length in hours).
3. Enter tasks from your WBS, including the dependencies established earlier.
4. Assign human resources to each task and identify any over-allocations (Resource Allocation View).
5. Set a baseline (think of this as 'freezing' the *initial* schedule for comparison with actual progress).
6. Select views required for reporting (Gantt, Critical Path, Milestone, Resources, etc).
7. Monitor progress by updating the '% complete' on tasks.

Hint: Before adding tasks to a new MS Project file, add a new column in the task window to show Work (in hours) for each task. Use View > Table > Work.

The best way to learn this software is to actually use it. The 'Project Guide' (i.e. 'wizard') is a useful starting point.

Important: Don't be tempted to simply adjust the hours associated with each task to make all your tasks fit the allowed duration of your project! Doing this makes the whole exercise futile. Your options are to work harder (= more hours) or reduce the number of tasks (i.e. the scope of your project). Pretending the tasks will take less time is *not* good Project Management.

Also… the default working day in MS Project is 8 hours. Use this with caution; if you know that you will only spend 2 hours a day on your project, specify this as 25% Effort Units.