How an Engineer Manages a Project

A Two-Hour Seminar presented by
Dr. Erik W. Aslaksen
at
Sydney University, 27 March 2003, 1200-1400 h

1 INTRODUCTION

1.1 Purpose of the Seminar

The purpose of the seminar is to introduce you to a way of approaching a project that will improve your efficiency both during your studies and in your careers. We shall look at some practical rules, methods, and processes that you will be able to put to immediate use in your project. But in addition to that, you will hopefully gain an understanding of the underlying concepts and relationships, so that you can make your own intellectual contribution to the management of your projects. A technician knows what to do and how to do it; an engineer also knows why. As an engineer, you will be expected to approach any task or problem in the frame of mind that you, as an individual, will make a difference. Engineering is a creative activity, and while its purpose is different to that of science or art, an engineer relates to a technician somewhat like a sculptor does to a stonemason. Just learning how to turn the crank may get you through your engineering course, but it will never make you a really good engineer. So, start right now being creative and innovative in all aspects of carrying out your project, and the management is one of them.

1.2 Purpose of this Document

Two hours is not a lot of time in which to make you a good project manager, so we shall have to think carefully about how to use them - we have to *plan* what we are going to do *before* we start doing it. (As a matter of fact, this seminar is itself a little project, and you can use it as an exercise in judging if I am being a good project manager!). One purpose of this document is therefore to serve as the Project Plan.

The second purpose is to provide you with some initial material that will allow us to start the seminar with a common understanding of some of the basic concepts, and thereby save some time. Please study and give some thought to it; perhaps also hold a little discussion session in your group to satisfy yourself that you have got it right.

The third purpose is to provide you with some supporting notes on the issues we shall be talking about, so that you do not have to waste time on taking detailed notes during the seminar. You should only make brief notes to record your personal understanding of the subject matter. That is why this document has the wide right-hand margin.

Finally, in studying the material in this document, try to put it in the context of your own project. Again, if you are able to have a little discussion session in your group prior to the seminar, you might try to answer the question: What does this mean as far as the way in which we will manage our project? I intend to use some of your projects to illustrate the processes outlined below, and expect you to participate in that.

1.3 Program

It is proposed to follow the program outlined below. However, if you can make a case for a change to this, or if it becomes apparent during the seminar that an amended program would be more beneficial, it will be changed.

1200-1215	Introduction of the groups, consideration of any proposed changes to the program or other initial comments or questions.
1215-1230	Review and discussion of the concepts.
1230-1245	Planning the project.
1245-1305	Group work on producing a project plan (ten minutes to do the work, ten minutes to discuss the results).
1305-1320	Controlling the project.
1320-1340	Group work on a project control schedule (ten minutes to do the work, ten minutes to discuss the results).
1340-1350	The review process and quality assurance.
1350-1400	Questions and summary.

As you can see, there are two short group work sessions, so that members of the same project group should arrange to be seated together. Also, please be ready to start at 1200 h sharp.

2 THE CONCEPTS

2.1 The Nature of a Project

A project involves three entities:

- The desired result (or outcome) of the project
- The work required to achieve that result
- The boundary conditions placed on the work

The term “desired result” implies that *someone* desires this result. So the first thing to do is to be sure you know who is going to judge the result and decide if it is the desired result or not. Let us call this person or group of persons *the users.* The users come with a whole background characterised by such things as language, education, experience, current situation, etc., and it is important that we consider the desired result in this *context.*

The desired result is expressed in an initial version of the *user requirements*. Do you have a complete and unambiguous set of user requirements? Do you understand them? Do you understand how you would test to see if they have been achieved? For example, if the users define a problem and then requires “the best” solution to that problem, do you know what is meant by “best”? - A good practice is to rewrite the user requirements, including the test requirements, in your own words, and then get the users’ agreement to them. (The resulting document may be called the *Requirements Definition Document*, or RDD.) The discussions involved in reaching that agreement would often expose a number of misunderstandings, implied assumptions, and so on. - Don't be afraid or embarrassed to ask questions at this point; you'll be a lot more embarrassed if it turns out halfway through the project that you have misunderstood what the users really want.

When developing an RDD, be careful not to let any elements of what you think the solution might be creep into this document. And the same goes for the users' requirements; be sure they express what the user really wants and not what they think is the solution to getting what they want. This is the process of *requirements elicitation*.

> In the case of your project, can you identify the users; do you understand what the requirements are, and how they will be tested? I.e., what are the *critical success factors*?

Just as the user requirements must be considered in the context of the users, the work has to be considered within the boundary conditions imposed on it. Typical boundary conditions are:

- The timeframe for the project, consisting not only of the end date, but often also of such constraints as the availability or accessibility of resources, delivery times for material and services, and holidays.
- The available funds.
- The types of resources (people, laboratory and workshop facilities, etc.) available.
- OH&S requirements.
- Preferred suppliers/equipment

> In the case of your project, can you identify any boundary conditions? Are they all equally important? What is the difference between a boundary condition and a critical success factor?

2.2 The Importance of Planning

Once you are absolutely clear about what the desired result is, and have listed all the boundary conditions, the next step is to *plan* how you are going to go about reaching that result. It is difficult to overestimate the importance of planning, and more projects have come to grief because of poor planning than because of poor execution. It is inherently very tempting to just "get stuck into it"; often there is pressure to show early evidence of diligence; planning is seen as dry and boring, and, above all, some engineers have never been taught the basics of planning that make the task easier. However, the most frequently advanced argument against planning is that here is so much uncertainty about anything in the future that it is best to just get started and then handle each event or circumstance as it arises. There is a kernel of truth in that, but this argument misses one of the main benefits of planning - that it allows you to handle all those unforeseen changes in a much better way than you would otherwise have been able to.

> *The best thing about not planning ahead is that one avoids long periods of doubt and anxiety, with failure coming as a total surprise.*

The basic structure of a plan is always the same; it provides answers to the following questions:

- *What* has to be done?
- *How* is it going to be done?
- *Who* is going to do it?
- *When* is it going to be done?

We shall see how we go about planning a project in just a little while, but first we need to introduce a couple of more concepts.

2.3 Complexity and the Concept of a System

What do we mean when we say that something is complex? That it requires many parameters to describe it? That it consists of many parts? That we are likely to make mistakes when doing it? Possibly a combination of all these.

In all areas of professional activity - the law, medicine, business, and engineering - the level of *complexity* of the subject matter has been increasing, and there are at least three major reasons for this. One is simply the accumulation of knowledge, another one is the increasing interrelation between all elements of society - individuals, political, ethnic, and social groups, and geographically separate groups - due mainly to greatly increased means of communication. And a third one is the explicit inclusion of humans as individuals into processes where they were previously either ignored or only considered in an implicit, cursory manner.

A first response to this increase in complexity, as a means of handling the sheer mass of data and knowledge, was the emergence of *specialisation* and of the specialist that knew more and more about less and less. However, this subdivision into isolated areas of knowledge carried with it its own limitation - the narrower the focus, the less likely it was to provide a true representation of reality, because in reality, everything is interconnected. A second, and more sophisticated, response was to overlay the specialisation with a layer of activity, which, to any desired extent, provided a connection between the specialisations, and the central concept of this layer was that of a *system*. The system concept is a means of handling complexity (rather than reducing it, as specialisation does), and arises from the realisation that complexity is relative - for example, what is complex to the human mind may be simple for a computer, and vice versa. The mind can manipulate objects that are characterised by more than one parameter as entities; that is, it is able to consider the parameters simultaneously rather than sequentially, as a computer normally does. But there is a limitation to this ability; as the complexity of an object increases and the number of parameters exceeds something like seven or so, the mind finds it rapidly more difficult to consider the object as an entity, and automatically starts to group the parameters into smaller groups and to process them as separate objects. The most immediate evidence of this is language; in order to express something complex, such as a story, we use a limited set of vowels which can be combined to form words, the words are subdivided into groups (nouns, verbs, predicates, etc.) and combined to form sentences, and the whole story is a string of sentences. Another example is in the way we create organisations, and a third example is our number system. And that is the core of the system concept. A system is a view of a complex object as consisting of a set of less complex objects, the system *elements*. But, in order for the set to represent the original object, the elements need to be *interacting*; the original object is represented partly by the properties of the elements and partly by their interaction. Were they not interacting, they would simply form a collection of the less complex objects; in the case of language, the interaction takes place in the mind of the listener and is determined by the sequence of the vowels, words, and sentences.

There is, in principle, no limitation on the type of complex objects the system approach can be applied to. If the object is a complex physical object, the system elements are physical subsystems, such as the hull, the propulsion, the steering, the services, and the fitout in the case of a ship. If the object is a complex body of work, the elements are tasks, and to illustrate the application of the system approach, consider such a complex body of work, consisting of a vast number of individual activities, all interacting in some way. We first group these into a small number of main groups of activities, typically definition, planning, implementation, and verification. Already at this first level we can take some tentative decisions, such as allocating a duration to each group, allocating resources and determining who is going to be responsible for each group, etc. In a next step, each of these groups are again subdivided into smaller work packages and take more detailed decisions, and so on, until we arrive at a large set of tasks, but where each task is not more complex than that we can conceptualise it and define it as an entity in all its detail.

Such a breakdown of a complex entity into a set of less complex entities is called *partitioning* the complex entity.

> How would you rate the complexity of your project? What would be an appropriate scale to measure it on?

2.4 Systems Engineering

The final concept we want to define is *systems engineering.* When we use the term "engineering", we generally mean a structured approach to creating something that will meet predefined requirements in a cost-effective manner. An engineering project therefore involves two entities - the object that is going to meet the requirements (what we previously called the result), and the body of work involved in creating it. For any significant project, both of these entities are complex, and they are obviously intimately related. *Systems engineering is the simultaneous application of the system approach to both the object and its creation.*

The need for simultaneity arises because we develop both the partitioning of the object and the partitioning of the work in a step-wise, top down manner, and at each level we have to ensure that the relationships between the object and the work are maintained. The partitioning of the work will refer to the partitioning of the object (e.g. as in "Design of the propulsion system"), and the partitioning of the object will depend on how we want to execute and manage the work (e.g. "The part to be built by Lisa" and "The part to be built by John").

In the next part of this seminar, we shall concentrate on the project management, i.e. the management of the work, and not explicitly refer to the object or its partitioning. However, from the above, you will understand that it is implicit in everything we do.

> In the case of your project, is there an obvious partitioning of the object? How many partitionings can you come up with? Why are some "better" than others? What do you understand by "better" in this context?

3 PLANNING THE PROJECT

3.1 The "What"

The first part of the planning is to define what the work consists of, and this is done in the form of a *Work Breakdown Structure,* or WBS. In accordance with our systems approach, we break the work down into a hierarchy of *work packages*, and the work packages on the lowest level of the breakdown are called *tasks.* The WBS will become the key to the project database, and in order to fulfil that role, it is necessary to introduce a numbering of the work packages. This is normally done by using a number of the following form: a.b.c. …, where a is the number of the first level work package, b the number of the second level work package, and so on, for as many levels as required. If there are more than nine work packages on a level, you simply use a two-digit number on that level. (The work package 0 is, by definition, the whole project.)

Each work package has to have a title that is reasonably descriptive of the work contained in it, and in the case of each task, it is mandatory that the title describe an activity (and therefore contains a verb), such as "Develop a preliminary layout" or "Produce a final report".

The simplest and most efficient tool for documenting the WBS, without resorting to specialised project management software, is to use an Excel spreadsheet.

Each task is defined by the following three pieces of information:

- A brief description of the work
- Definition of what is to result from the work (i.e. the output)
- Identification of what is required in order to commence the work (i.e. the initial conditions)

To document the definition of each task, you can use either an Excel spreadsheet encompassing all tasks, or create a small form, using the Table function of Word, which can be filled out for each task.

3.2 The "How"

For each task, you need to describe briefly how you intend to execute the task. That is, the methodology (incl. any checking or QA involved), any special tools needed, services or materials (incl. software and data, such as digital maps) to be purchased, special skills required, etc. In short, any information that is required in order to later estimate the effort and cost involved. If the task has to be carried out in accordance with an existing procedure, that should be noted.

This information is added to whatever format was used to document the "What".

3.3 The "Who"

Now you have to define who is going to be responsible for the task. There has to be only one person responsible for each task; if you find that there needs to be two persons responsible, each for a part of the task, the task should be split into two.

This information is also added to the format used to document the "What".

3.4 The "When"

In principle, each task has a date by which it must be completed, in order to meet the overall project delivery date. However, this completion date is only directly determined by the project requirements for those tasks that result in a project deliverable (equipment, report, drawings, etc.). For the other tasks, this piece of information can only be determined after completing the estimates and taking into account the relations between the tasks, as explained in the next two sections.

3.5 Estimating Effort, Duration, and Cost

You now know what you need to do and how you are going to do it. But how much effort is involved in doing it? This has to be *estimated*, and such estimates can be based on previous experience or on a best guess. In the former case, you may have done this one or more times before, or you have done something similar that can be scaled, or you base it on published data of a general nature (e.g. the lines of tested source code that can be produced per man-hour). In the latter case, you simply have to try and imagine you are carrying out the work, or use any plausible argument you can think of. The main point here is that, as long as you document how you arrived at your estimate, *any estimate is better than none.*

Next, you estimate the duration of each task. If each task is performed by a single person, that is easy and follows directly from the estimate of effort, but in general it involves examining the available human resources and determining how best to apply them to your tasks.

Finally, the cost is estimated by multiplying the hours assigned to each team member by his or her hourly rate and adding all costs for materials and services, resulting in the *project budget.*

3.6 Relationships Between the Tasks

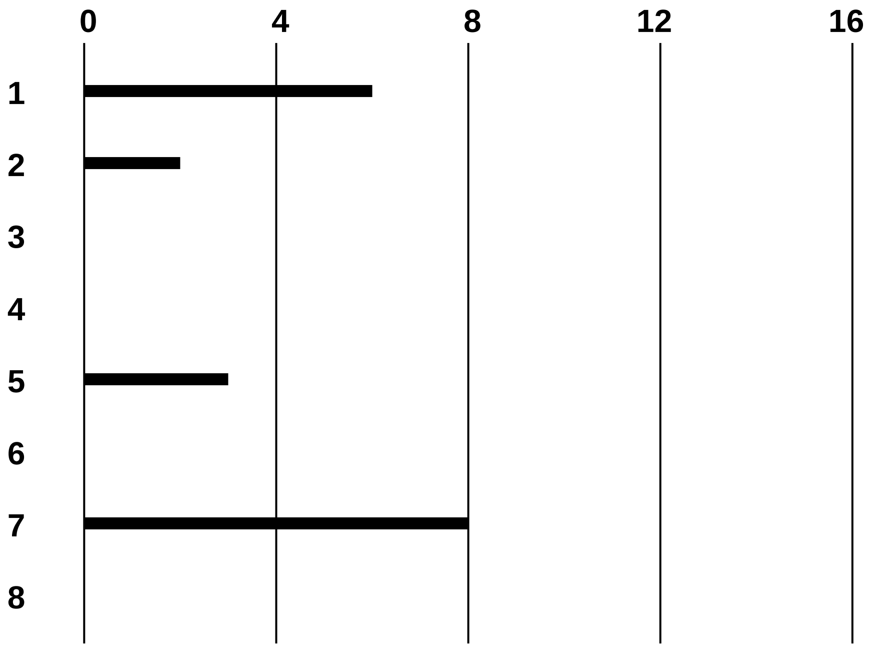
As noted in 3.1, each task requires certain inputs to be available before it can start, and these inputs are generated by another task (or by several other tasks). This introduces temporal relationships between the tasks, and this is best documented in the form of a bar chart (often called Gantt chart). The steps involved in developing such a bar chart are simple:

1. On the y-axis we list all the work packages (as per the WBS), the x-axis is the time scale.
2. To each of the tasks not requiring the input from any other task (i.e. it has no *predecessors)*, assign a bar starting at t=0 and having the length of the estimated duration.
3. To each of the tasks requiring inputs only form the first lot of tasks, assign a bar starting at the end of the last predecessor of the task, showing the predecessor relationships by lines and arrows.
4. Repeat 3, until all tasks are accounted for.

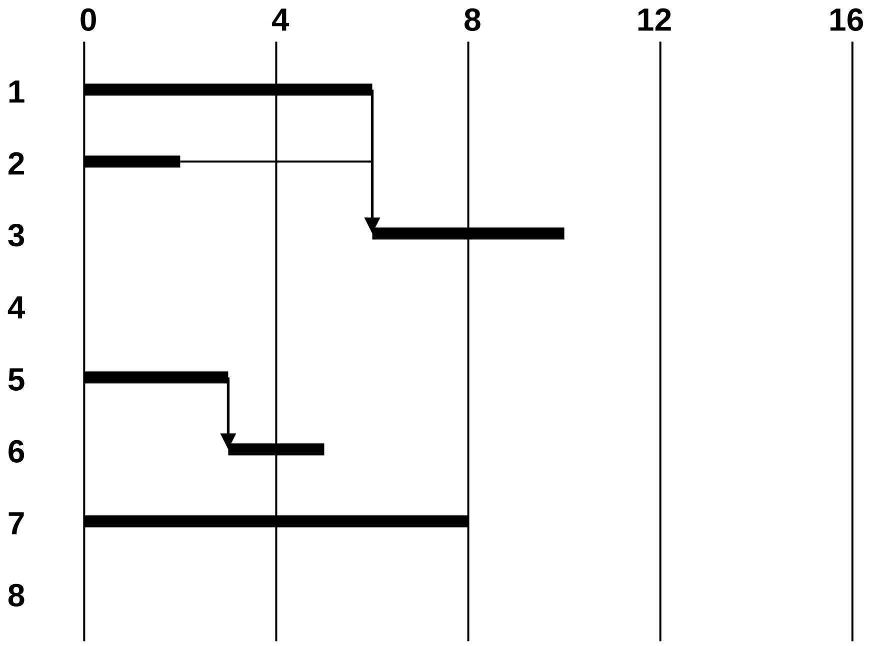
A simple example will illustrate these steps:

Task	**Duration**	**Predecessors**
1	6	
2	2	
3	4	1,2
4	5	3,6
5	3	
6	2	5
7	8	
8	4	6,7
End Milestone		4,8

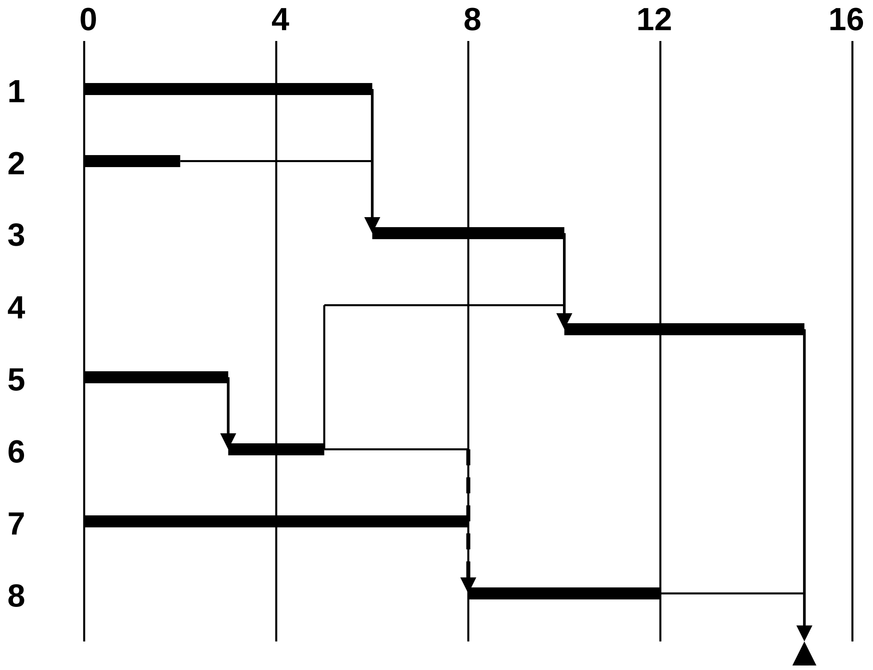
After Step 1:



After Step 2:



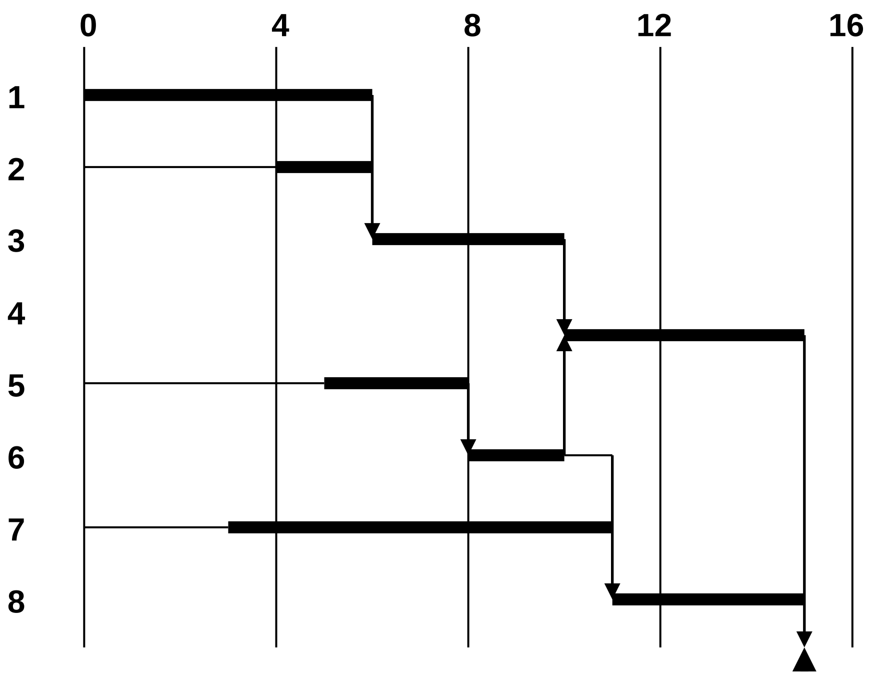
After Step 3:



There are many *paths* that lead from the start to the finish, but there is only one which consists of a contiguous sequence of tasks, without any *float*, and that is called the *critical path*. In the present case that consists of the sequence of tasks 1,3,4, and it obviously determines the total duration of the project. There are a few things to note about the bar chart as it now stands:

If we wanted to try and shorten this duration, we would need to shorten the duration of one of the tasks on the critical path; either by shifting resources from one of the other tasks into a critical one, or by employing additional resources. There would be no sense in putting additional resources into any of the other tasks.

The endpoints of the tasks now represents the *early start and finish* dates, i.e. the earliest the tasks could be started and finished. However we now have the opportunity to shift the non-critical tasks to a later date; if we shift them as far as possible (without changing the project duration), we obtain the *late start and finish* dates, as shown below:



In between these two cases there is obviously a wide choice of possibilities, and the final or "best" choice may be determined by the boundary conditions, by the desire to level the resource utilisation, by the desire to minimise risk, or by financing considerations, to name a few.

The above steps can be carried out automatically if we use a software planning tool, such as MS Project, which will be available to you. But remember, while tools can make management tasks much more convenient and quicker to carry out, they are never a substitute for understanding what you are doing and why you are doing it, and for using your judgement as to what is important and what not.

4 CONTROLLING THE PROJECT

4.1 Cost/Schedule Control

Control consists of two functions: *Measuring* the deviation of the controlled parameter from its planned value (set-point), and then taking *corrective action* to reduce the deviation to zero. Note that this description of control presumes the existence of a plan. Without a plan, no control; that is why having a (documented) plan is so important.

Let us first consider the case where the controlled parameter is cost; then the first function is called accounting. The costs incurred by the project in a fixed time period, called the *accounting period*, are captured sometime during the period and summed up at the end of the period. Two important features of the cost accounting are already apparent: Firstly, if we are going to be able to compare the *actual cost* with the *budgeted cost*, and thereby establish the deviation from plan, the accounting system and the budget must have the same accounting period. Secondly, if the accounting period equals the project duration, there can be no control; all we will have is a historical record. If we are to exercise control, we have to be aware of any deviation as early as possible, and so we would make the accounting period as short as possible. The optimal accounting period is determined by the accuracy of our planning; how confident are we of the value and time of occurrence of individual cost items? If the accounting period is too short, the deviations lose their *significance*; if it is too long, the corrective action becomes less effective.

The accounting period is one of the two factors that determine the *resolution* of our cost control system; the other one is the number of *cost accounts*. The cost accounts should correspond to the work packages on a particular level of the WBS; the lower the level, the more cost accounts we will have, and the greater the resolution. In this case the number of accounts is limited not primarily by the accuracy of the budget, but by the practicality of allocating actual costs to small work packages (or even to individual tasks).

> In the case of your project, what would be a reasonable accounting period and a reasonable set of cost accounts? What do you estimate the cost to be, taking the cost per hour of your time to be $25? What is the difference between a cost account and a cost type?

With our accounting system in place, we have to consider what to do with the information. We immediately run up against a problem, which is best illustrated by imagining that you spend the money sitting in the pub, drinking at a pace that corresponds exactly to the budget (S curve). No deviation, no need to take any corrective action. Is that correct? Of course not; in the end you will have spent the money and have made no progress on the project. The cost information only becomes useful when it is combined with information about the progress of the project. So, how do we determine the progress?

Basically, there are two different approaches. In the first one, we *estimate* the progress on each of the tasks, and usually the person responsible for the task does the estimate. This approach, while at first glance very sensible, has several drawbacks. Firstly, people are likely to be optimistic and overestimate their own progress. Secondly, while 90 % of the planned work on a task may have been completed, all the problems encountered have been pushed into the last 10 %. Thirdly, non-labour costs may occur at a rate quite different to the progress as measured by the work done.

The second approach is what may be considered a binary one; a task is either finished or not, and if not, it is not counted as progress at all. Again, a similar situation as with the budget: The more tasks, the better the resolution. (In my opinion, the second approach is preferable whenever possible.)

In either case, the cost and progress (schedule) information is combined by using the following three quantities:

BCWS: Budgeted Cost of Work Scheduled. This is the budgeted cost of the work planned to be carried out in the accounting period.

BCWP: Budgeted Cost of Work Performed. This is the budgeted cost of the work that was actually carried out in the accounting period. This quantity is also called *Earned Value.*

ACWP: Actual Cost of Work Performed. This is the actual costs that have occurred during the accounting period.

We can now define two measures of project performance:

Progress Factor: BCWP/BCWS

Efficiency Factor: BCWP/ACWP

These two factors give a good indication of the status of the project; what remains is the *analysis* of any deviations (of the factors from the value 1), i.e. determining the *cause* of the deviation (which might be that the plan was wrong in the first place), and then deciding on the appropriate corrective action.

Let us finish off by applying this cost/schedule control system, $(CS)^2$, to one of the previous examples.

4.2 Controlling Information

Every engineering project is critically dependent on information; external information as input to the process of engineering, and information generated by the process. Controlling this information - acquiring it, storing it, retrieving it, and making it available to the appropriate project participants at the right time - is a major management task. In some cases projects are carried out in an environment (company, dedicated project infrastructure, etc.) which provides an ICT infrastructure that makes this almost automatic, but in the case of your project, that is not the case, and it will be an important exercise for you to set up your own information management system.

First of all, you should consider in what form you want to handle the information. Despite the hype about the paperless office, you will probably find that it is most convenient to have a hard copy of every document. So, one of the team members should be assigned the role of document controller, and he or she will maintain a file with a hard copy of every document (letter, email, specification, drawing, article, etc.). This would typically be in the form of one or more ring binders, and could be subdivided into the following sections:

External correspondence
Internal correspondence
External documents
University (i.e. Client) documents
Internal documents (incl. deliverables)

In the front of each section you should maintain a listing of the documents in the section, and the dates on which they were stored. If a document is taken out of the file, a sheet should be inserted in its place detailing who borrowed it and when.

As you will be working as a team of six or so members, you will have to consider how you are going to transmit information between the team members. Hard copies can be exchanged at weekly meetings, electronic copies are conveniently exchanged by email. You could also consider setting up a project website, where information can be stored for convenient access by team members in their own time.

Once you start generating project-specific documents, such as reports and other deliverables, you will need to ensure that you control the version of each document. In particular, if several of the team members are collaborating on a document, you will have to decide how you will ensure that you work on the same version, or that you do not "overwrite" each other. One way of ensuring this is a sequential way of working, marking up a single copy of the document, so that each member sees the changes made by the members that worked on the document before him or her (i.e. circulating the document), but this can be very time-consuming. Another way is for all to work on a copy of the document simultaneously, and then hand their contributions to a document "owner", who then edits and collates the contributions and raises the document version.

You should agree on a standard format for the documents you are going to produce (if this is not prescribed by the Client), and this should include a header and footer, with the header containing the issue (or version) number and date of issue, and the footer containing the page numbering. Any change, be it ever so small, must result in a new version of the document.

You will probably require some data as input to your project. It is very convenient to copy a page out of a book or download and print a page from the Internet, but most often this page will then have no indication of the identity of the source document. Be sure to note down where every piece of data comes from; that is, the full identity of the source document - URL, author, publication (book, journal, etc.), version (if applicable), etc. - so that you can refer to it in your work.

4.3 Reviews and Quality Assurance

Quality means "according to agreed requirements". The output or result of your work - generally, the deliverables - will be of high quality if it meets the user requirements, and any deviation from that is called a *non-conformance*. How can you ensure that you deliver high quality work? One approach is to *test* the deliverables, and then correct any non-conformances through *rework*. However, rework is expensive, and you should not view testing (primarily) as a method of catching your mistakes; the primary purpose of testing should be to *verify* (or prove) that you have met the requirements.

A much better approach is to design your *process* such that it has a very high probability of delivering a conforming product, and instead of focusing on testing and rework as a means of maintaining high quality, you focus on *controlling* your process so that it functions as designed and thereby *assures* a high quality of the product. Some major considerations in designing a production process are:

- People with the right skills and training for the job
- Appropriate and well maintained tools

- Clear and well documented procedures, instructions, and specifications for every operation or part of the process.

In engineering design, an important sub-process is the *review* process, in which somebody else goes through your work in a pre-defined manner. And it is a process; it is not something you tack on at the end. You have different types of reviews at different points in your project, e.g. at the definition stage, at the conceptual design stage, and at the detailed design stage.

Basically, there are three types of reviews:

Professional review - Am I doing the right thing? Am I doing what the users require? Will this approach lead to the desired deliverables?

Practice reviews - Am I doing things right? Am I using the right formulae? Am I adhering to the appropriate standards? Are my calculations correct?

QA reviews - Am I adhering to the process? Have the other reviews been carried out? Have all non-conformances been closed out? Is there documentary evidence of this? Is the document control in order?

Two issues are of particular importance. Firstly, the review process needs to be *documented*. Before you start a review, you should know what you are looking for, and it is useful to document this in the form of a *checklist*. A couple of examples are contained in Appendix A. Secondly, you need to document the outcome of the review in the form of a *review report* or *review minutes* (you can use the check list, if you design it in that way), where you list any non-conformances, and for each one document what corrective action is to be taken, who is responsible for taking it, and by when it has to be completed.

Secondly, there is the issue of *traceability.* As you go through the engineering process, it is easy to lose sight of the original requirements - the user requirements. Your final product should be *necessary* and *sufficient* to meet the user requirements. If you look at the specification for your product (i.e. the document you build it to), you should be able to justify each requirement by relating it to a user requirement, so that you have not over-designed your product by adding in "nice" features that are not required by the users and only add to the cost. On the other hand, you should be able to demonstrate that each user requirement has been met by relating it to requirements in your product specification. One way of doing this is to construct a *traceability matrix*, where the user requirements, and the columns label the rows by the requirements in the product specification. Where a row and a column are linked (as above), the corresponding matrix element is marked in some way, and there must be at least one non-empty element in each row and column.

And now:

Good Luck!