



Defining the Scope of a Project

Neville Turbit

Scope v Time & Cost

When people talk about scope, they immediately think time and cost. Time and cost are outputs of scope. Determining scope is a different exercise. In the context of this white paper, when we talk about defining the scope, we are talking about developing a common understanding as to what is included in, or excluded from, a project. We are not talking about deciding how long it will take, or how much it will cost. That comes after the scope is defined.

If we were looking for a car, we would first define the scope. For example we want a 4-cylinder front wheel drive with seating for 2 adults and 2 children, and less than 2 years old. Maybe you also want it to be a red convertible.

Having defined the scope, you can calculate cost and time. How much you will have to spend and how long you will take to buy it. If you get the scope wrong, the time and cost will be wrong.

Why is Scope important?

Anyone who has ever done a project will have tales of how scope changes caused grief. Scope is bound to change, and this is to be expected. As the detail becomes clearer, more complications creep in. These are not foreseeable at the start and hopefully we build in a contingency for what we cannot see.

The scope changes that usually cause problems are those where the perception of what was in and out of scope was different between various parties. The Project Manager assumed there would only be four or five reports, and the business assumed ten to twenty. Nobody felt it was worth talking about because they assumed the other person thought the same way they did.

How scope is usually defined

Scope definitions often account for a paragraph or two in a Business Case or Project Charter. Often, they are qualitative and/or focus on general statements.

"We will improve service by providing an information system to respond to customer inquiries."

Is it a real time system? Is it all screen-based? What reports can be produced? Where does the information come from? What manipulation is required for the data? Is all the data compatible? Do you want to generate standard letters? How many letters? How customisable are the letters? Do you want to store the questions? Do you want to store the answers? Etc. etc. etc.

Define the Outcome

We will cover several different ways to successfully define scope. All should start with an agreement on the outcome. The outcome is the change that will occur when the project is complete. Examples are:

- We will be able to answer customer queries regarding statements over the phone.
- All licensing details will be accessible on-line and we will be able to identify when they are due.

Assumptions

In order to define the scope, there will be assumptions that need to be made. There is no point in waiting until everything is clear to define scope. By that time, the project will probably be finished. Each of these assumptions should be documented and followed up at a later date to validate the scope. If the assumption is false, it may have an impact on the scope.

Which way to define Scope?

There are numerous ways to define. Ideally several ways should be used. Each looks at the situation from a different perspective and will elicit different information. We look at three main ways in this paper. They are:

- Define Deliverables
- Define Functionality and Data
- Define Technical Structure

Define Deliverables

One method to focus people on the scope, is to define the internal and external deliverables.

- External deliverables are things the project delivers to the users eg screens and reports. Users typically think of a system in these terms. It also includes any hardware or software required by the users or the project team.
- Internal deliverables are things the project generates internally eg Project Charter, Business Requirement Specification etc.

It is likely that the users will not be absolutely clear on all the deliverables. In this situation you can make generic assumptions. For example, you might not know exactly what reports are required but you allow for 12 unspecified reports.

Once the external deliverables are defined, the Project Manager can define the internal deliverables.

Example External Deliverables:

Name	Description
License Detail Screen.	Screen to enter and view license details
Company Summary Screen	Screen to view all licenses issued by a particular company. Facility to drill down to License Detail Screen.
License Due Report	Report listing all licenses due in the next period. Facility to select a period eg 1 week, 4 weeks, quarter
5 Reports	Allow for 5 unspecified reports
Server	Server to run the application
Etc.	

Example Internal Deliverables

Name	Description
Project Charter	Document identifying how the project will be managed
Business Requirement Specification	Document identifying the requirements for the project
Weekly Reports	Status reports to be issued weekly
Prototypes x 3	Three prototypes will be allowed for in the development.
Etc.	

Define the Functionality

Another technique is to define the functionality. This should not be either a long or detailed process. Typically, depending on project size, the exercise can be completed in a one hour to half-day workshop. A good technique is to use a functional decomposition. If using a spreadsheet and a projector, a scribe can create the scope as it is discussed. Remember to start all functionality with a verb.

It is useful to do the functional decomposition in conjunction with a data definition. If this is not possible, once the scope is discussed, it will become reasonably clear what data is required.

The Project Manager can determine if there are any situations that need to be clarified with the users, and finalise the scope definition. If for example, in defining the functionality it becomes evident that considerable information will need to be transferred from a legacy system, which is known to be inaccurate, data cleansing can be factored into the scope.

Example Functional Decomposition

1.0 Capture License details

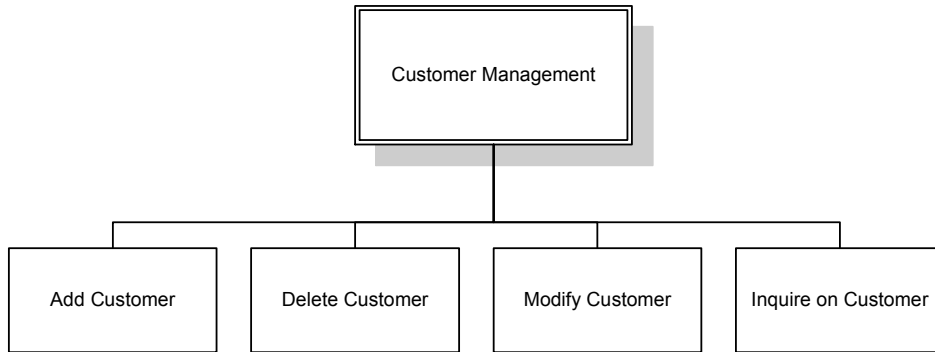
- 1.1 Set up companies
- 1.2 Set up products
- 1.3 Create licenses
- 1.4 Modify licenses
- 1.5 Delete licenses

2.0 Generate payments

- 2.1 Create payment report
- 2.2 Authorise payments
- 2.3 Notify accounts

Etc.

It can also be defined as a diagram:



Defining the Data

This approach is similar to functionality, and should be used in conjunction with functionality. The process is likely to capture what users expect to see in a system. The intention is not to make the business users, data modelers. The intention is to get the business users to verbalize their requirements for information in a structured manner. Ask the users what are the people, places and thing they want to keep track of. In this case, the focus is on nouns.

This approach will not capture data that may be required to technically make the system work. For example, it will not capture things like transaction log files, archive files, SQL script files etc. Post workshop, the Project Manager will need to sit with a data modeler to sort out what else is required. The hardest part is to stop doing a data model. Keep the focus on where the data is to come from, and identify what is new, where the interfaces are likely to be, is existing data suitable, is the data currently captured etc.

Data Definition Example

Name	Description
Companies	Details of the company including address, overseas offices, and up to ten contacts
Licenses	Licenses for all software and hardware used in the organisation. Include contracts, correspondence, quotes and any other related documents. Does not include manuals
Renewal dates	Dates the license is due for renewal and the cost of the renewal.
Etc.	

Technical Structure Definition

This technique can be useful in defining scope where the project is focused on infrastructure. It can also be useful in a situation where an existing system is being modified. The output can be either a table, or a diagram. A table might just list the components to be modified and the modification. The structure diagram might identify the whole system and highlight which components are being modified and

how they are being modified. It may also be appropriate to indicate the purpose of each component, however it will probably be vague at this stage of development.

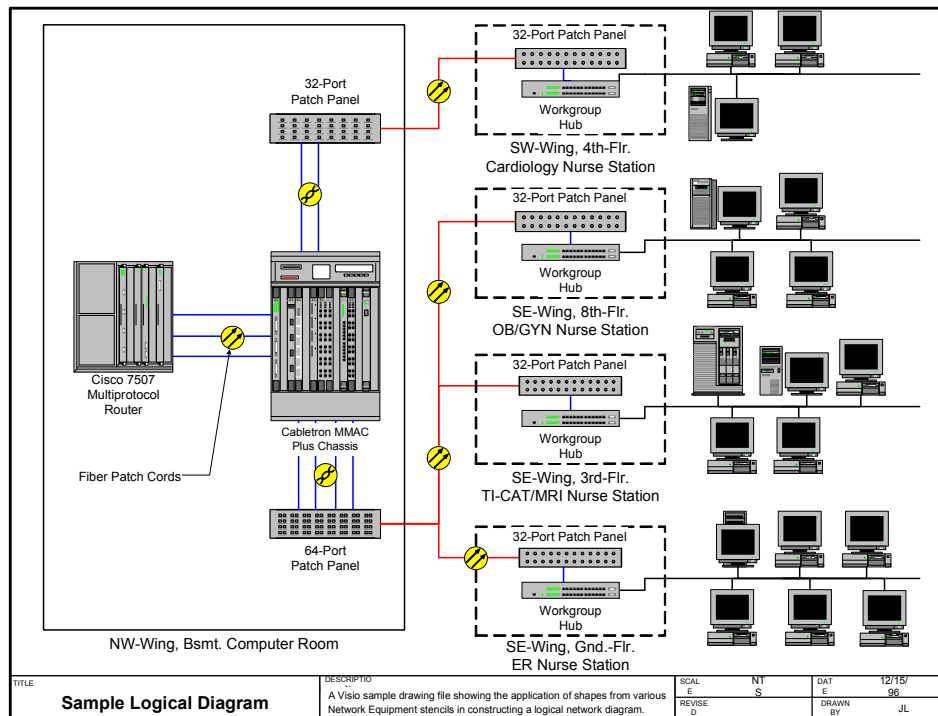
Example:

The ‘outputs HTML’ module takes information retrieved from the database and inserts it into an .asp document for output to the server. It also updates a transaction log with the database information and time of the output. If an error occurs in retrieving data from the database, an error log is updated and an error page sent to the server.

Example Technical Structure Table

Component	Description
Subsystem1	Handles all customer processing and interfaces to CMS (Customer Management System).
Subsystem2	Carries out inquiries on billing systems (2) and combines data into common format. Sorts data by date of payment.
Etc.	

Example Technical Structure Diagram



Other Considerations

In documenting the scope of the project, also consider describing the project boundaries, identifying the major business events, locations, divisions, functions and

processes affected by the project, as well as the groups of people impacted both inside and outside the company.

Consider also:

- Business processes that will be affected;
- Business areas/units that will be affected;
- Business locations that will be affected;
- Business data that will be changed;
- Business applications that will be changed;
- Technologies that will be changed

All of these may have an impact on the project. For example if numerous locations are to be effected, there may be issues of bandwidth, switching hardware, travel and training, remote support etc.

Other Work

The following is a list of work that may need to be specifically included or excluded. Make sure it is clear if this is to be included, and that it is documented. When we come to planning, it should be clear what needs to be catered for in the plan:

- Preparation of training material
- Delivery of training
- Business Process documentation
- Business Process Re-engineering
- Rework
- Project management and administration
- Vendor management
- Security
- Disaster recovery plans
- Business continuity plans
- Provision and setup of equipment
- Software
- Communication
- Support after go-live
- Recruitment of permanent or contract staff
- Staff performance management and evaluation
- Hardware upgrade or purchase
- Hardware installation
- Data preparation for transfer
- System documentation

Summary

Defining the scope is a neglected area in most projects. It is however the foundation on which the schedule, budget and resource plans are built. Get it wrong, and everything else will be wrong. If the scope definition does not run to a few pages, it is probably too short.

Take the time to workshop the scope with users. Make sure there is a shared understanding. Force the business to think through the project. Use a number of techniques to cross check. Finally, unless you get the scope right, the project will

never be under control and scope creep will likely cause the project to be considered a failure.

Neville Turbit has had over 15 years experience as an IT consultant and almost an equal time working in Business. He is the principal of Project Perfect. Project Perfect is a project management software consulting and training organisation based in Sydney Australia. Their focus is to provide creative yet pragmatic solutions to Project Management issues.

Project Perfect sell “Project Administrator” software, which is a tool to assist organisations better manage project risks, issues, budgets, scope, documentation planning and scheduling. They also created a technique for gathering requirements called “Method H”™, and sell software to support the technique. For more information on Project tools or Project Management visit www.projectperfect.com.au