



PROJECT **PERFECT**
Pty Ltd

Scope Management User Guide

Prepared By: Neville Turbit
Version 1.0
1 Feb 09

Table of Contents

Document Origin	2
Change History	2
Overview	3
Definitions	5
Breaking down Projects	7
Scope Views	9
Other Project Work.....	14
Establishing Project Boundaries	15
Establishing Scope.....	16
Define Outcome.....	17
Define Deliverables	18
Define Functionality	20
Define Data.....	21
Define Technical Structure.....	22
Scope Variation	23
Appendix A – Project Variation Request	24
Appendix B - Cumulative Variations	28

Document Origin

No.	Author	Department
1	Neville Turbit	Project Perfect

Change History

Version	Date	Changes
1.0	1 Feb 09	Initial Version

Overview

Purpose of this Document The purpose of this document is to outline how the scope of a project should be documented.

Scope definition A project's scope defines the deliverables of that project. The deliverables are all the products and services to be provided by the project and thus define the work to be performed by the project team.

The existence of a specific scope is a defining attribute of a project.

Reason for establishing scope Scope needs to be established in a project for three reasons.

Reason	Explanation
Common understanding	<p>There is a need to create a shared understanding between the ultimate owners of the “Output”, and the people who are creating the “Output”.</p> <p>Normally there are some grey areas on both the business and IT side of the project at inception. In defining the scope, decisions on what is to be included or excluded can be made.</p>
Time & Cost	<p>When the scope is agreed, the project team will use this as the basis to calculate:</p> <ul style="list-style-type: none">• How long the project will take• How many people will be required• How much it will cost
Scope variation	<p>It is normal for scope variations to occur in a project. Unless however, there is a benchmark scope established at the start of the project, there will be confusion as to whether a particular component is part of the original scope, or a variation.</p>

Scope of this document This document covers the identification and documentation of the scope of a project.

Continued on next page

Overview, Continued

Scope exclusion This document does not cover the quantification of the project in terms of:

- Time
- Cost
- Resources

For more information see the user guides for Time, Cost and HR.

Definitions

Constraints A “Constraint” is something that has happened, or a situation that exists, and cannot be changed. The Project Team needs to accept it as a fact, and work within the boundaries of the “Constraint”.

Example

- We requested three developers but when the project was approved to proceed, only two were allocated to the project. There is no negotiation on this allocation.
 - Key Business User is on leave for the next four weeks. Leave cannot be changed. The project needs to wait four weeks to begin.
-

Assumptions An “Assumption” is an interpretation of the situation made at a point in time, which is yet to be validated. An “Assumption” still needs actions to validate. If the “Assumption” is later proven to be false, it may become an “Issue”.

Example

We make an assumption that a certain skilled resource will be available from a certain date. If the person is not available, it becomes an “Issue”.

Business Problem The business situation, which has caused the project to come into existence.

Objectives What the project will achieve to overcome the “Business Problem”

Outputs "Outputs" are the tangible deliverables of a project. There are two types of deliverables:

- "External Deliverables" are deliverables produced and delivered to the end users e.g. new system, particular functionality and reports.
- "Internal Deliverables" are deliverables produced within the project that contribute towards the "External Deliverables", or final product of the project e.g. Business Requirement Specifications, Development Plans

Continued on next page

Definitions, Continued

Outcome

An "Outcome" is the result of the delivery of particular "Outputs" e.g.

- A 10% improvement in processing time
- Ability to process a particular transaction online.

Outcomes are the answer to the question “What will be different in the organization when we finish the project?”

Breaking down Projects

Reason History has proven that the larger the project, the more chance of failure. The reason is that if a project is too big, it is impossible for people to understand the implications of a particular situation that might arise. It is better to break a big undertaking down into a number of smaller projects. Each project should deliver a component of the final solution.

Conversely, an activity that may take a week for one person is not a project. It does not require the rigor of project management disciplines and is probably just a series of half a dozen related tasks.

Granularity Taking a hierarchical view:

- A Program consists of Projects
- A Project consists of Phases
- A Phase consists of Activities
- An Activity consists of Tasks

Example:

Hierarchy	Example
Program	Financial Systems Program
Project	Accounts Receivable Project
Phase	Project Planning
Activity	Prepare Project Definition
Task	Discuss scope management process with Sponsor

Project Size A "Project" should not exceed six months. Any "Project" larger than six months should be broken down into several "Projects" which may form a "Program". Each "Project" should have clear deliverables.

A project that is less than a month is probably an "Activity".

Continued on next page

Breaking down Projects, Continued

Phase Size

A "Phase" exists to enable a project team to focus on delivering something tangible in a manageable period. That period is typically between two and six weeks.

If the "Phase" exceeds eight weeks, there is the danger that a project team will loose focus and the amount of work to be undertaken will not be properly managed. Delays can occur but not be apparent for some weeks. If delays occur in a two to six week phase, the delay becomes evident almost immediately and action can be taken to rectify the situation.

Scope Views

Different views Scope can be viewed by different people, in different ways:

- Outcomes
- External deliverables
- Internal deliverables
- Functionality
- Data
- Technical structure

Outcome Defining the scope by an outcome is a useful way to focus the team, however it is not sufficiently detailed to put in place proper project scope control. How the outcome is achieved can result in significant differences in scope.

Example:

- Integration of customer billing and customer payment information to enable real time inquiry of customer account status.
 - Ability to process customer payments at a rate of x per hour
 - Customers able to pay accounts over the Internet
-

External Deliverables If you talk to business users about a new software system, their view is likely to be screens and reports. It will not usually include internal deliverables such as project plans and test plans. Nor will it initially be seen as functionality and data. Users see the system as the external deliverables.

Example:

Deliverable	Description
Customer Screen.	Screen to inquire on, and maintain customers
Billing Enquiry Screen.	Displays details of the billing and payments for a customer
Overdue Report	Shows all customers that are overdue.
Etc.	

Continued on next page

Scope Views, Continued

For an infrastructure project, there can also be external deliverables.

Example:

Deliverable	Description
25 PCs	Windows 2000 SOE installed in Accounts Dept
2 Printers	Installed in Accounts Dept
Internet Access	15 of the PCs with Internet access
Etc.	

Internal Deliverables

In running the project, certain documents or code will need to be produced as a step towards the production of the final or “External Deliverables”.

Example:

- Project Definition Document
- Project Plan
- Communication Plan
- Business Requirement Specification
- Three Prototypes of system xyz
- Etc.

WBS (Work Breakdown Structure)

A WBS can be used for scoping however it is more likely the WBS will be prepared as part of the Time Management part of planning the project. The WBS does not necessarily give a clear view of the scope. For more information on WBS see Time Management User Guide.

Functionality

Scope can be defined as functionality and data. Functionality should be defined using a functional decomposition to level 2 or 3. All functionality should start with a verb below level 1. It may be presented as a list or a diagram.

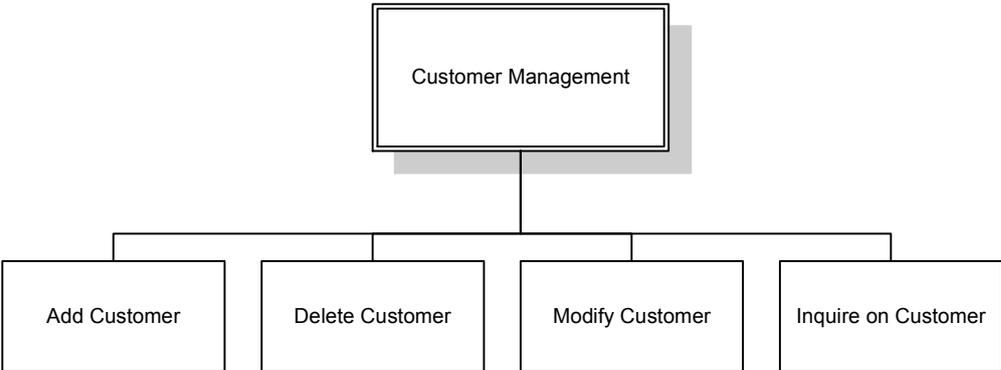
Example Functionality List:

- 1.0 Customers Management
 - 1.1 Add customers
 - 1.2 Delete customers
 - 1.3 Modify customers
 - 1.4 Inquire on customers

Continued on next page

Scope Views, Continued

Example Diagram:



For any significant project the diagram is likely to have anything up to 4 or 8 level 1's (e.g. Customer Management). It is appropriate to spread the diagram over several pages.

Data

Data can be listed as entities, which are in business terms, 'things we want to keep track of'. At the scope definition level, little effort is required to normalize data or separate entities from attributes.

For example, it may be in the final database, one table may hold customer names and another customer phone numbers. If it is not immediately evident there is a 'one to many' relationship (e.g. Can a customer have more than one phone number?) treat them as one entities.

Entities are nouns.

Example:

Entity	Description
Customers	Details of name, company, date set up etc.
Addresses	Can be mailing, physical, delivery addresses for each customer
Phone numbers	Can have mobile, fax etc.

Continued on next page

Scope Views, Continued

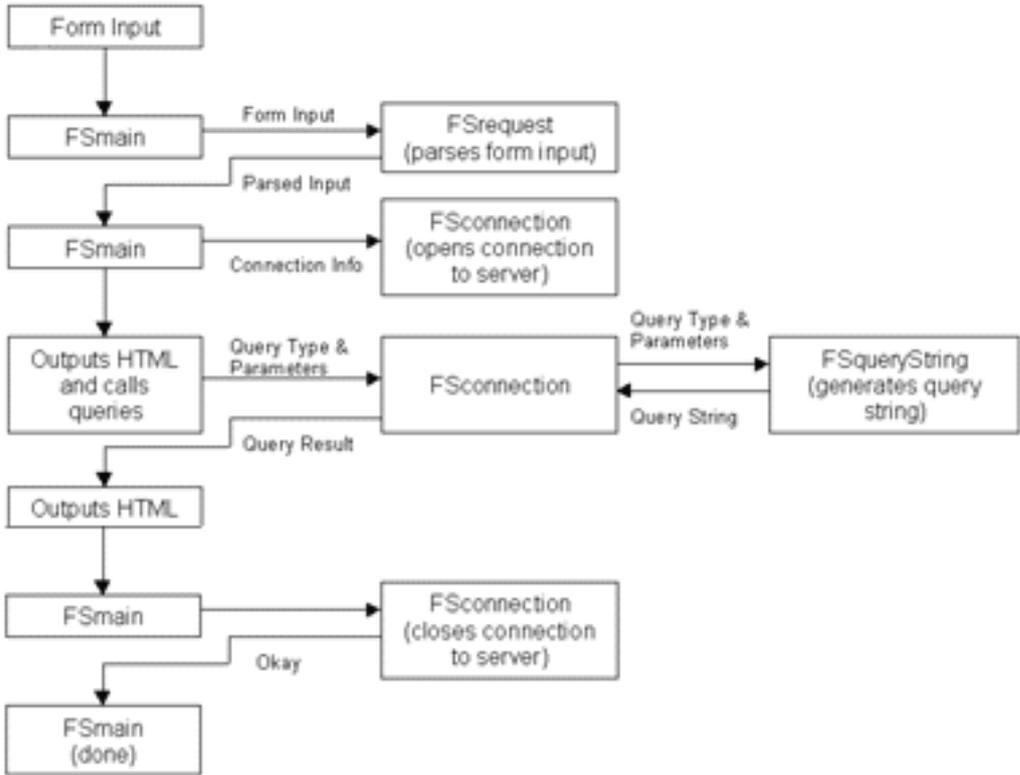
Technical Structure

From an IT perspective, the structure may be defined as systems and subsystems. This may be done either as a list, or more likely as a diagram.

Example:

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.	

A typical "Technical Structure" diagram is shown below:



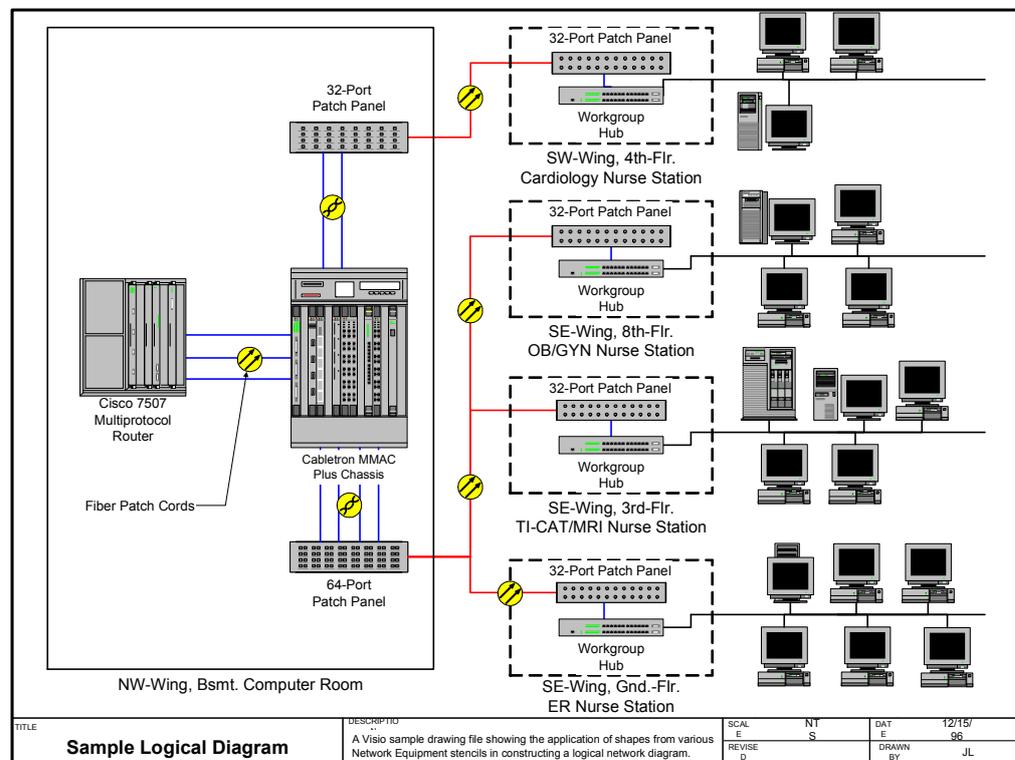
Continued on next page

Scope Views, Continued

Infrastructure project

A “Technical Structure” diagram can be useful for an infrastructure project.

Example:



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

Other Project Work

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
-

Establishing Project Boundaries

Establishing the boundaries The starting point for defining the scope is to gain agreement on what we are trying to make happen, and the limitations of the project.

- What we are trying to make happen - which is the "Outcome" - comes from the "Business Problem" and "Objectives"
- The limitations are documented as "Constraints"

Business Problem The project exists to overcome a problem. The first step is to identify and agree the "Business Problem".

Example:

The "Business Problem" is that we cannot provide up to date billing and payment information to answer phone inquiries.

Objectives From the "Business Problem" will come the objectives for the project.

Example:

The "Objective" is to combine information from billings and payments system into an online inquiry screen that can provide real time information.

Outcome Hence the outcome will be the change in the way we operate.

Example:

The "Outcome" is that we will be able to answer customer-billing inquiries immediately instead of searching out the information and returning their call.

Constraints Identify any "Constraints" that the project needs to deal with. "Constraints" are 'non-negotiable' factors affecting the project.

Example:

- A solution needs to be in place within 3 months
- The solution must be integrated with the CRM system

Establishing Scope

Method to use There are three main methods to use as outlined in the section on “Scope Views”. They are:

- Define Deliverables (Internal and External)
- Define Functionality and Data
- Define Technical Structure

At least one method, and ideally all three methods should be used. The more ways the scope is defined, the more likely there will be a clear understanding as to what is being delivered.

Also the more clarity regarding the scope at the start of the project, the less likely scope variations will be required. Many scope variations are due to a misunderstanding as to what was in scope originally.

Participants The first two methods should be undertaken with both Business and IT people. The latter (“Technical Structure”) may be undertaken by IT alone. If a “Technical Structure” is being used, it should be used in conjunction with one of the other two methods.

In the case of infrastructure projects, the “Technical Structure” may suffice on it’s own.

Venue It is recommended that if there are more than two business users involved, the best method to capture the information is in a facilitated workshop.

Define Outcome

Purpose The purpose of defining the “Outcome” is to ensure the design of the “Output” is tailored to achieve a particular result.

Input The input for determining the “Outcome” are:

- Business Problem
- Project Objectives

Form of Outcome The outcome is usually expressed as Critical Success Factors (CSFs). These are the results that, if achieved, determine the project was successful. They should be measurable achievements.

Example:

- Improve transaction processing rates from 50 per person per day to 75 per person per day
 - Enable customer inquiries to be answered immediately.
 - Reduce the number of statistical faxes that need to be manually input by 90%
-

Define Deliverables

Purpose To define both the “Internal Deliverables” and “External Deliverables”. The “External Deliverables” should be defined first. The “Internal Deliverables” can be defined from the “Project Management Methodology” and “Applications Development Methodology”.

Inputs Definition and agreement of the:

- Business Problem
- Project Objectives
- Constraints

Procedure The following procedure can be used.

Step	Action
1	Draw up a high level business process on a whiteboard
2	Identify what screens, reports, enquiries, interfaces and new information are required at each stage
3	It is likely that not all reports will be clear in people’s minds. If necessary add a number of unspecified reports.
4	Document any assumptions
5	Document any areas where the scope is unclear, and how the scope is to be clarified

Assumptions It is probable that assumptions will need to be made. When assumptions are identified, they should be noted, and actions created to validate or clarify the assumption. Those actions may occur some weeks or months into the future when other decisions have been made, or the development has progressed to a certain point.

Example:

We assume the information from two billing systems can be combined into a common file and sorted in date order.

Action: J. Smith to confirm that the common file can be created. Due 01.05.03

Continued on next page

Define Deliverables, Continued

Scope to be clarified

Some areas of scope may be hazy. These should be noted and resolved as soon as possible.

Example:

The management reporting requirements are not clear.

Action: Project Manager responsible for holding a meeting with the Sponsor in the next week to clarify what information will need to be provided to management. Due 05.05.03

Define Internal Deliverables

The project team (and in particular the Project Manager) can then identify the methodology to be used and from that methodology define the “Internal Deliverables”

MS Project Plan

Standard templates are in the process of being produced to cover the typical deliverables in a project. These templates will be for a variety of situations such as different project size, application development, infrastructure, software purchase etc.

Define Functionality

Approach The high level functional decomposition should be done in a workshop if there are more than two business users.

Procedure Adopt the following procedure.

Step	Action
1.	Brainstorm on a whiteboard what the system is required to do. Use verbs to start each item.
2.	Try to group the functions into logical groups which will become level 1 items
3.	Give each level 1 item a name – not necessarily starting with a verb. From level 2 however, all functionality should start with a verb.
4.	Group the functionality into a hierarchy usually down to level 2 or 3
5.	Review for missing functionality and complete.

Level 1 definition Level 1 can be defined in different ways. Some of those are listed below. There can however be other ways depending on the individual project.

- Departmental e.g. Customer Service Dept, Billing Dept, Accounts Payable Dept, etc
 - Functional e.g. Payment Receipt, Payment Enquiries, Banking, etc
 - Input/Output e.g. Receipting, Daily Reporting, Monthly Reporting etc.
-

Define Data

Approach The approach is similar to the functionality approach. If there are more than two business users, use a workshop approach.

What will be captured/not captured This process is likely to capture what users expect to see in a system. It 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.

Level of discussion If business want to keep track of billing numbers and billing amounts, it is not necessary to discuss that both are attributes of a billing table. An exception may be where *every* attribute is being discussed. In this case the facilitator will need to move the focus to entity level rather than keep it at attribute level.

Conversely, if the workshop talks about customers and orders, IT may know that because there is a many to many relationship, there needs to be a customer order table. It is not necessary to discuss the need in the workshop.

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. Keep the focus on nouns.

Post workshop IT should do a quick review to both rationalize and expand the list.

For example business may identify the need to track customer data but IT would see that as perhaps a customer file, address file, phone number file and related customer file.

The list may need to be rationalized where several attributes are mentioned.

Focus Remember the focus is on defining the scope. It is not to create the definitive data model. If the project team finds they are starting to create the logical data model, they have probably gone too far. They should pull back and re-focus on defining scope

Output List of data tables within the scope of the project plus identification of existing tables to be accessed

Define Technical Structure

Inputs

A clear understanding of the expected “Output” will be needed in order to develop the technical structure. If the understanding is not clear, the functionality of each component will be unclear.

This method of defining the scope is better suited to a situation where an existing system is being modified. The structure diagram will identify which components are being modified and how they are being modified.

Another place where a “Technical Structure” can be useful in defining scope is where the project is focused on infrastructure.

Additional information

It may also be appropriate to indicate the purpose of each component, however it may 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.

Scope Variation

Process in place It is critical to the success of a project that an approval process is in place to handle variations to scope. This process needs to be established prior to the project commencing, and any variation after the initial agreement of scope needs to be included in the process.

Level of authority It is permissible for the approval process to have different levels of authority.
Example:
If the variation is under 2 days and \$1,000 the Project Manager can approve the variation. Anything over this amount needs to be approved by the Sponsor.

Cumulative variation approval It is also important to cap variations if there is a tiered approval process. For example the situation above is fine unless there are 10 or 20 small variations.
Example:
The Project Manager can approve up to 5 variations. After that point, a review of all variations needs to be carried out with the Sponsor and the Sponsor will determine the future approval process.

Tracking cumulative impact A register should be kept of the cumulative effect of variations. It is important to know how the project has changed since the initial approval. The cumulative register should include:

- Variation number
- Variation name
- Additional cost
- Additional time
- Who approved

Variation Request form There is a standard form for requesting and approving variations. An example is included as an appendix.

Appendix A – Project Variation Request

PROJECT VARIATION REQUEST

Variation Request Number: 1

Project Name: **Example:**
Finance Project

Project No: **Example:**
10000001

Requester: **Example:**
John Smith

Variation Type **Instructions:**
Identify what sort of variation is required
Example:
Creation of two new reports

Response required: **Instructions:**
Identify when a decision needs to be made to include or exclude the variation.

Variation Details **Example:**
Two new reports are required to provide end of month figures of actual sales versus budget and also actual sales versus estimate. Figures are currently compiled manually.

Impact of not proceeding **Example:**
If we do not have the report we will need to continue processing the information manually. This takes 1.5 mandays and costs \$1k per month.

Benefits**Instructions:**

Identify the benefits. They should be quantified. Such comments as better service or less problems should be avoided.

Example:

- Reduction in manual processing accounting for 1.5 mandays and \$1k per month
 - Reduced risk of errors. In the last 6 months we have provided incorrect information to the board on two occasions resulting in revisions having to be made. On at least one occasion, the correct information would have resulted in a different decision from the board on cash flow projections.
-

Costs**Instructions:**

Identify the cost of the variation in terms of dollars, time and resources.

Example:

The additional work will require a BA to spend 1 day to specify the reports, and a developer to spend 2 days to product the reports. Testing will require 1 day. In total, the cost will be \$5k and the time 4 days.

Implications**Instructions:**

What impact will the variation have on the project?

Example:

The variation cannot be included in the schedule without delaying UAT by at least 2 days. We will need to bring in another developer if we are to keep to the original schedule.

Approval**Instructions:**

Identify the action including the signature of the person who will approve or disallow the variation. Include the agreed cost and time variations.

Example:

Criteria	Agreement
Action	Agree to proceed using a new developer for two days.
Increase to Budget	\$5k
Increase to Schedule	2 days
Sponsor's Signature	
Date	

Implementation The approved variation has been incorporated into the project:

Scope Schedule Budget Resource Plan Status Reports

Appendix B - Cumulative Variations

Example

Scope Variation Register - Finance Project					
No.	Name	Date	Status	Cost	Time (Days)
	Total Approved			\$2,000	5
	Total Rejected			\$2,600	6
	Total Pending			\$8,000	12
1	2 Mthly Report	01.02.03	Approved	\$1,500	4
2	Lookup name	07.02.03	Approved	\$500	1
3	Add Yrly Totals	12.02.03	Rejected	\$2,600	6
4	Extra security level	19.02.03	Pending	\$8,000	12