**PROJECT DOCUMENTETION**

**Statement of Work (SOW)**

*Generally required if you are acting as a vendor in a project*

Assuming that you are doing a project as an implementation service provider or outsourcing vendor, this is perhaps the most important document that is created in the pre-project phase.

In most companies, the sales/pre-sales/marketing department prepares the SOW, and this creates an issue down the line, as in most cases the project manager is unaware of the agreements that were made between the client and the vendor before the project.

A SOW is a technical document apart from the commercials. Therefore it makes sense to get it prepared by a Project Manager or someone with project management experience.

**WHAT IS AN SOW?**

In most cases, a client and a vendor sign up on a legal master services agreement (MSA) for a fixed time. This MSA usually covers multiple projects. A SOW on the other hand is an agreement **covering a particular project**— so if you have ten projects running with a vendor, you will have ten SOWs.

**What does an SOW contain?**

Typically, a well-written SOW will contain the following:

* Scope
* Assumptions
* Technology platform/Architecture
* Roles and Responsibilities of the client and the vendor
* Project Methodology
* A draft project times schedule with milestones
* Dependencies/Risks
* Acceptance Criteria
* Change Management Process

Sometimes, I have heard many people say, that before the project, how can I identify all dependencies and risks, and that is a fair question. The answer is that you do not have to identify “all” dependencies. If you can identify a “few” dependencies that would be good enough.

Remember, that the Project Manager of the project would read this SOW to create his project plan, so the more detailed it is, the higher the chances that the Project Manager will be able to create a good project plan, leading to a greater chance of project success.

**2. Project Charter**

When a project starts, this is the first important document that should be created. If there is an SOW and you are a Project Manager, then one of your first job is to read the SOW thoroughly and prepare a detailed project charter.

Like a SOW, while starting a project many things in the project may not be clear, and that is normal, but try to detail out your thoughts in the project charter with what you know, and check back with the customer, your team and stakeholders regarding points that need to be clarified before you start the actual work.

Remember, that a well written project charter will also make it easier for anybody joining your project to be an effective contributor, because they can refer to the project charter where they can find all the key information about the project. And should you ever hand over the project to somebody else, the handover will be a lot easier for you.

Most companies that I have worked with have a well-defined template/structure for a project charter, but in case your company does not, then you can refer to the following to build up your own project charter.

Typical contents of a good project charter:

* Project overview
* Project Stakeholders
* Scope
* Project schedule
* Budget incl. estimates
* Assumptions
* Responsibilities
  + Client responsibilities
  + Vendor responsibilities
* Project methodology/phases
* Project Metrics that you plan to collect and the point in time at which you plan to report the metrics.
* Major risks — see below section about the Risk Log
* Change management process — defines how change requests are handled
* Project Governance mechanism including audits
* Communication
  + Communication plan – who communicates to whom
  + Status Reporting – frequency, templates
  + Escalation path
* Any deviation/customizations that you have made from the organization’s standard project process if any

**3. Configuration Management Plan**

*Recommended for large organizations with big (distributed) teams*

A Configuration Management Plan is usually developed as a separate plan and linked to the main project charter.

The purpose of development of a configuration management plan is to identify the process/methodology of:

* Version Control
* Change Management (for handling change requests)

Typically, a project would have project documents, customer supplied documents and source code. In general, any project artifact that undergoes changes, and it is important to refer to the correct version, must be placed under version control.

Examples of such documents may include:

* Project Plan
* Project Schedule
* Requirement specification
* etc.

For version controllable documents and code, you must define:

* Access control – who can read, who can create?
* Approval – who can approve?
* Where is the current version kept?
* Where are previous versions archived?

Change Management – recall that the SOW already lists out the agreed change management process that will be followed. As a project manager, it is your job to detail this out and identify:

* Who can raise a change request?
* How will change requests be recorded? – An excel template may be required
* What will be the methodology that will be followed once a change request is received, including re-estimation of project effort/price and schedule?
* Who can approve a change request?
* What will be the process of implementing the change?

Of the project failures that I have seen, a weak change management was one of the key reasons for the failure.

**4. Project Estimates**

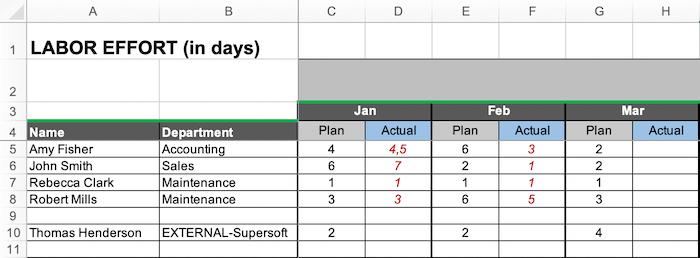
This is one of the key project documents that I have seen mostly missing in a badly managed project. In most organizations, the project estimation is done by the sales/pre-sales team during the pre-project phase and then this is forgotten. But that should not be the case.

A project as mentioned above may go through various changes.  Therefore it is absolutely necessary that you as the Project Manager make a re-estimation of the project effort and change the project schedule every time there is a change request in the project. Whether the customer will pay for the changes or not is a different question and is entirely commercial, but the fact is a change will impact your project estimates (your manpower requirement) and your project schedule.

At the end of the project, when someone challenges you on the ‘delays” in the project, you, the Project Manager, would be able to defend your stance only if you have maintained your change request history/log, you have re-estimated your project effort and have made changes to the project time schedule and have stored all the previous versions, along with all necessary communications.

The best way to maintain a project estimate is to maintain the project estimate in a different file (such as your budget tracker) and link the latest version to the project charter.

*I record effort estimates in my budgeting sheet, side by side with the actual (or real) effort, which allows me to spot problems early on:*

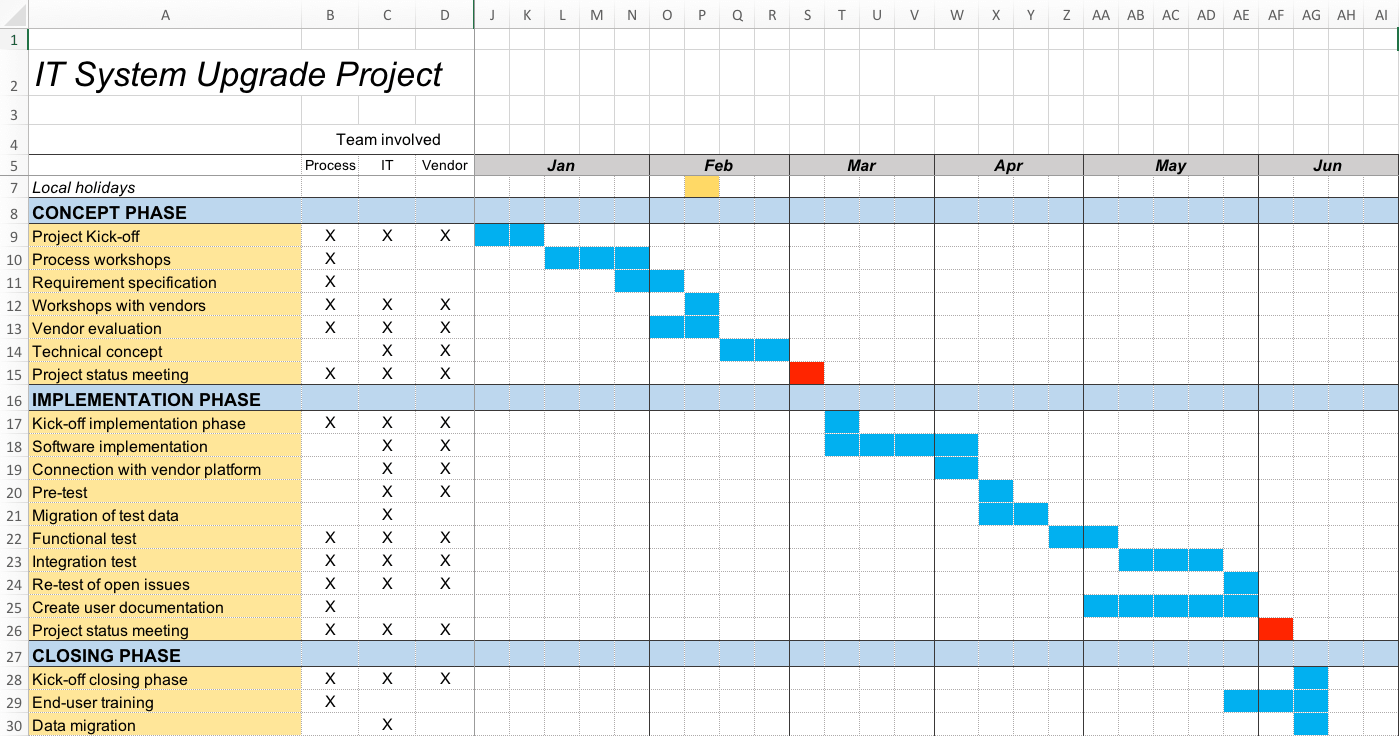


I can offer you different types of [budgeting sheets](https://www.tacticalprojectmanager.com/templates/project-budget-template/) which include effort tracking. The months-based budgeting sheet (image above) is included in my [Project Template Pack](https://www.tacticalprojectmanager.com/project-templates-pack/).

**5. Project Schedule**

One of the key project documents that you should create and maintain is the project schedule/time-plan.

A good project plan gives you a visual of when the project can end, can show you changes to the schedule as and when you make updates.



*I like to create my project schedules in Excel – you can get this template as part of my*[*Project Template Pack*](https://www.tacticalprojectmanager.com/project-templates-pack/)

You should also maintain versions of the project schedule, especially if the project is undergoing changes, so that at any point if you are asked why there has been a delay – then you can answer this question.

The best way to maintain a project schedule is to maintain it in a different file and link the latest version to the project plan – remember that a project plan is the master document of your project.

**6. Risk Log**

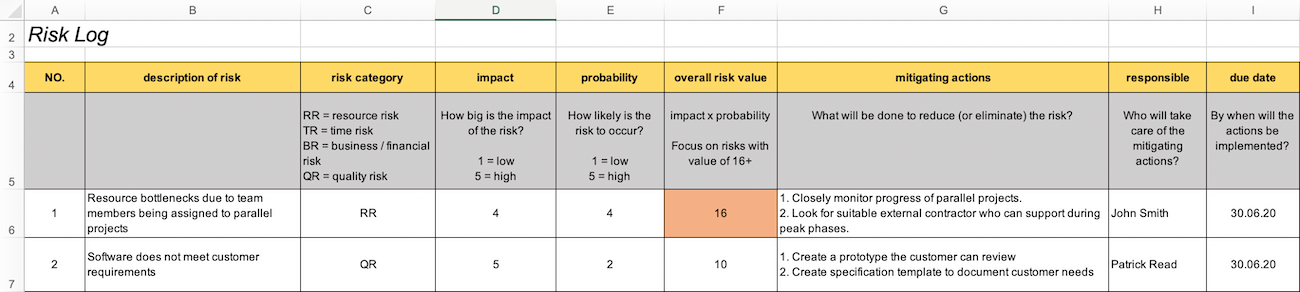
A risk log is again a key document that you should create and maintain during the project lifecycle.

A Risk Log is basically an Excel sheet with the following columns:

* Description of the risk
* Probability – High, Medium, low
* Impact – High, Medium, low
* Your mitigation/contingency plan

Most organizations that I have seen have a well-defined risk log template (also called risk register). However, if you don’t have a risk log template available, such a template is included in my [*Project Template Pack*](https://www.tacticalprojectmanager.com/project-templates-pack/) (along with other must-have templates).

This is what my risk log looks like:



[*You can get my Risk Log template here*](https://www.tacticalprojectmanager.com/project-templates-pack/)

You should also maintain versions of the risk register. Many times, people ask me how often I should update my risk register and there is no defined and set answer to this. In some projects that I have managed I have updated my risk register on a weekly basis and in some projects, I have updated on a fortnightly basis – but one thing for sure is the fact that in all the projects that I have managed, the frequency of updating has never been monthly.

The best way to maintain a risk log is to maintain it in a different file and link the latest version to the project charter – or a project plan as the master document of your project.

**7. Issue Log**

Several times I have faced a question on what the difference between a project risk and a project issue is.

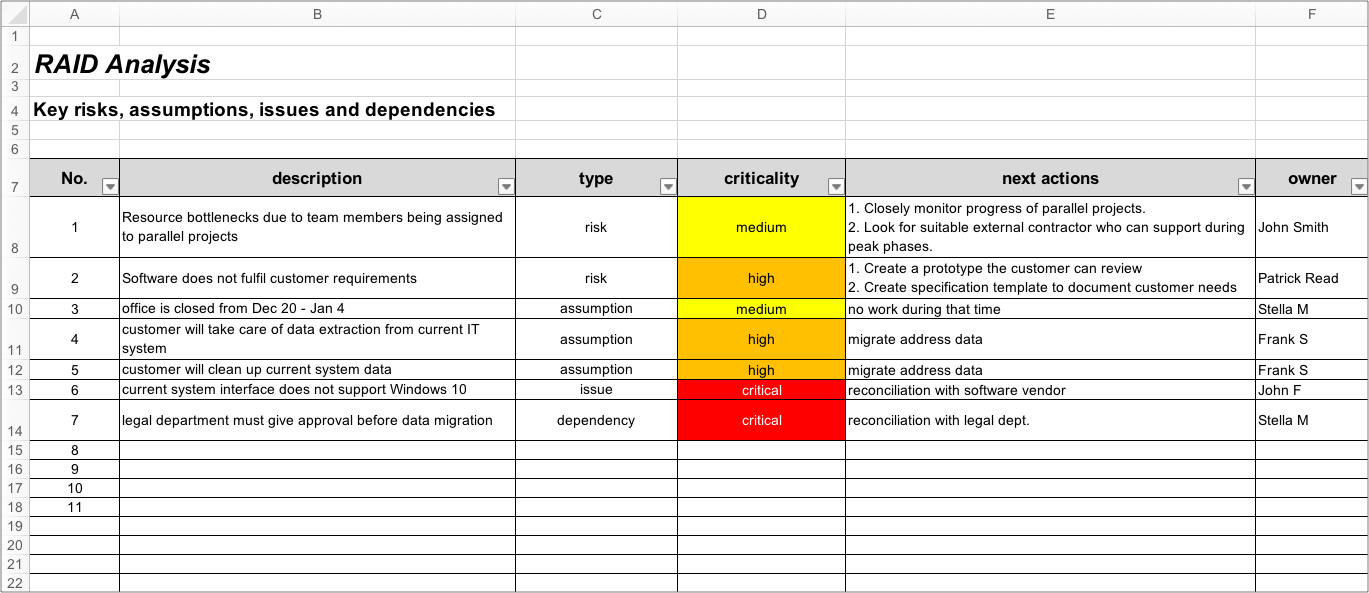
A project risk is something that you perceive may happen or may not happen e.g., attrition of a key project team member and you make a plan if the incident happens or does not happen.

A project issue is different, these are items that are hampering your project progress today – example you have not received the project’s database login id and password.

A good project manager usually updates his/her issue log on a daily/weekly basis and discusses the same with the customer/vendor on a daily/weekly basis, as the case may be.

You should also maintain versions of the issue log.

The best way to maintain an issue register is to maintain it in a different file (for example in a RAID Log, which is intended for tracking issues) and link the latest version to the project plan – remember that a project plan is the master document of your project.



*You don’t necessarily need to create a dedicated Issue Log — you can just record issues in a RAID Log, whose purpose is to capture current issues.*

**8. Status Report**

A well-managed project always has a status report (in a properly defined template) sent out to the customer. Many project managers have told me that they do not send out a project status report because more or less everyday there is a discussion and his/her customer never asks for one.

This I would call a bad practice, irrespective of whether the customer asks for a status report or not, irrespective of the fact if you are having discussions with the customer on a daily basis, as a project manager it is your duty to provide your customer a status report – I would suggest that you send out a status report on a weekly basis.

Trust me – It has been my experience that nothing delights the customer more than a good status report on early Monday morning in his/her mailbox.

You should provide the following in a customer’s status report:

* Brief summary of the project progress
* Items completed last week
* Items planned to be taken up next week
* Snapshot of the project schedule/milestones – with upcoming milestones highlighted
* Updated Project risks and their mitigation
* Current project issues and their status

It takes years of experience and practice to design a good status report template that fits the requirement/project/the customer, but you can take the above points as a start.

One more point, you would have to remember is that not all projects will run smoothly, and if at a particular point in time a project turns bad, then if you have sent out a project status report that acts as documented evidence that the issues in the project have been highlighted to the customer.

**9. Test Cases/Plan**

It is mandatory to draw up test cases/ plans for your project, especially if it is a development project. Test cases can be at multiple levels:

* Unit test case – drawn up at program level
* Integration test case – to test integrations with other systems, APIs etc.
* System test cases – to test the system as a black box

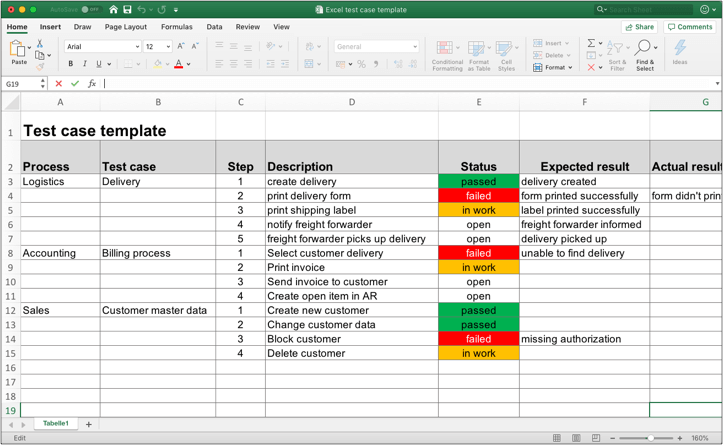
If you, as a project manager, have to ensure the quality of your software project, trust me, you cannot miss out on test cases. Most team members do not want to detail out test case and this is where you have to enforce as a project manager.

Without detailed test cases, no one can ensure the quality of the output irrespective of how good the team is in programming – it is human nature to miss out on a feature or that extra “IF clause” or that “obvious validation”.

If you are creating a test case template yourself, then you should ensure that it contains the following:

* Test case number – a unique number that identifies this test case.
* Feature
* Pre-conditions to this test scenario
* Post conditions to this test scenario
* Test scenario
* Expected Result
* Actual Result – you can use the test case template to log the actual result on running the test – in this case it becomes a test case and a test log.
* Pass/Fail – you can use the test case template to log the actual result on running the test – in this case it becomes a test case and a test log.

Below you can see the template I use for test cases. Have used it a million times in ERP system implementations, but you can use it for basically any IT project:



[Get my test case template](https://www.tacticalprojectmanager.com/test-case-template-excel-with-example/)

**10. Requirement specification**

If you are leading a software development or system implementation project – such as an ERP rollout, then this is a must have document that is required to be prepared.

A requirement specification is a detailed document that serves as a guide to the development team on the various features that need to be present in the system. Usually, a requirement specification follows a modular approach which means that you identify various modules (or processes) and then drill down/detail out the requirements at a detailed level.

To describe each feature, a diagrammatic approach along with a detailed logic of the feature is usually the best way to go about. Examples of diagrams that may be included are:

* **Context diagram:**A bird’s eye level view of the module/feature showing the various users that interact with the system
* **Process flow diagram:** usually a flow chart or swim lane diagram defining the proces

When you have had a business analyst prepare the above, what you have is the functional requirement – meaning the various features/functionality to be present in the system. However, it has been my experience that many times, people forget to write out non-functional requirements.

A non functional requirement is, in short, a system requirement but it cannot be directly connected with any module/feature/functionality. Example of non-functional requirements are:

* Number of concurrent users, of the system
* Any particular performance criteria that is present e.g. a transaction should be approved/rejected in 20 seconds

The next phase of the project after analysis is the design (or implementation) phase and in order to do a good design, non-functional requirements come in handy. The point that you have to remember is that if you have not catered to the non-functional requirements, then even if during the final user acceptance testing, it is seen that you have catered to all the functional requirements, even then your project will not get passed.

In short, non-functional requirements are usually a very small part of the requirement, but they are perhaps the most neglected/least documented, and as a good project manager it is your job to document the non-functional requirements and see to it that the system design caters to these. (Check out this page from California Polytechnic State University for an [overview of non-functional requirements](http://users.csc.calpoly.edu/~jdalbey/SWE/QA/nonfunctional.html)).

If you are looking for a Requirement Specification template, check out [my requirement specification template](https://www.tacticalprojectmanager.com/requirement-specification-template-requirement-numbering/) which supports **automatic numbering** of requirements and **matching of customer & product requirements**(to ensure traceability).