



The Rock Hard Project Manager

Software Project Management Fundamentals

Break free from the chains of flawed software project management and adopt the 7 key secrets to running any successful software project.

Contents

Introduction	3
Purpose of this Book	4
The Project and Software Development Life Cycle	5
Identify and Manage the Key Project Stakeholders	7
Identify Stakeholders	7
Communicate with and Manage Stakeholders	8
Stakeholder Checklist	9
Identify the Goals of the Project	10
Control Change to the Vision, Scope and Requirements	11
Project Goals Checklist	12
Organise and Plan the Project Activities	13
Estimate and Schedule the Activities to Produce the Deliverables	14
Determine and Organise the Project Roles	15
Planning Checklist	16
Manage the Project Team	17
Organise the Project Team	18
Manage the Project Team Checklist	19
Manage the Project Activities	20
Controlling and Tracking Progress	21
Manage Issues and Risks	22
Concluding the Project	23
Manage the Project Checklist	24
Create Clear Accurate Project Deliverables	25
Keep Documentation and Code Deliverables Available and Safe	26
Deliverables Checklist	27
End Game	28
The 7 key secrets to any successful software project	29

1

Introduction

Software Projects can vary considerably in difficulty, size and type.

This has led to many associated project and software development management methodologies being proposed and adopted.

Often these are of an ever-increasing complication in themselves and can contradict each other out of context.

Sometimes the core principles of project success that underpin all of the best formal methods are lost in the complexity that has been created to deal with differing development life-cycles and technology and in an effort to create faster, better project delivery.

Methods often become difficult to master because they try to provide a unified way forward for all types and size of project.

The good news is that the fundamentals are not complicated or difficult; they just have to be understood and undertaken. Following these fundamentals alone will help lead to successful projects, whatever the size, type and intensity of the project.

This document distills the experience that has been learnt across decades of active project management into the basics that will give a good grounding for project success, independent of any methodology, software tool-set or life cycle.



Purpose of this Book

To clearly state the fundamentals of Project Management success as applicable to Software Development;

Provide summary checklists for Project Managers to aid the effective running of Software Projects;

Give a straightforward Project Management grounding that allows related software project methodologies to be more easily understood;

Act as a contextual aid for the further books in the series by www.RockHardProjects.com.



This book is the first in a series aimed mainly at Technical and Administrative Managers involved in Software Development Projects.

The Rock Hard Project Manager Software Project Fundamentals is:

- a starting context for those who may be new to Project Management
- a refresher for those who have not led a software project in some time
- an informational aid for those currently in a purely technical or administrative role

Specific roles that this book may be of interest to within the Software Development process:

- Project Sponsor
- Project Managers
- Development Team Leaders
- Project Support Officers
- Technical Support Officers
- IT Staff Managers
- Departmental Managers

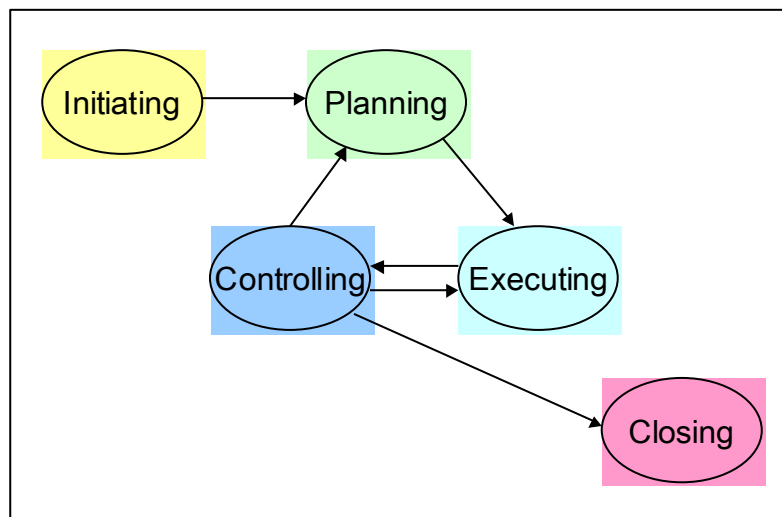
The Project and Software Development Life Cycle

Project Management is accomplished through the use of processes such as initiating, planning, executing, controlling, and closing.

A standard project will typically have the following phases:

- Initiate
- Plan
- Execute
- Control
- Close

The diagram below illustrates the flow and order of activities within these high-level phases:



These processes (without, for the moment, worrying what they consist of in too much detail) are common to any type of project.

For software development, the consideration of the Software Development Life Cycle (SDLC) is also required.

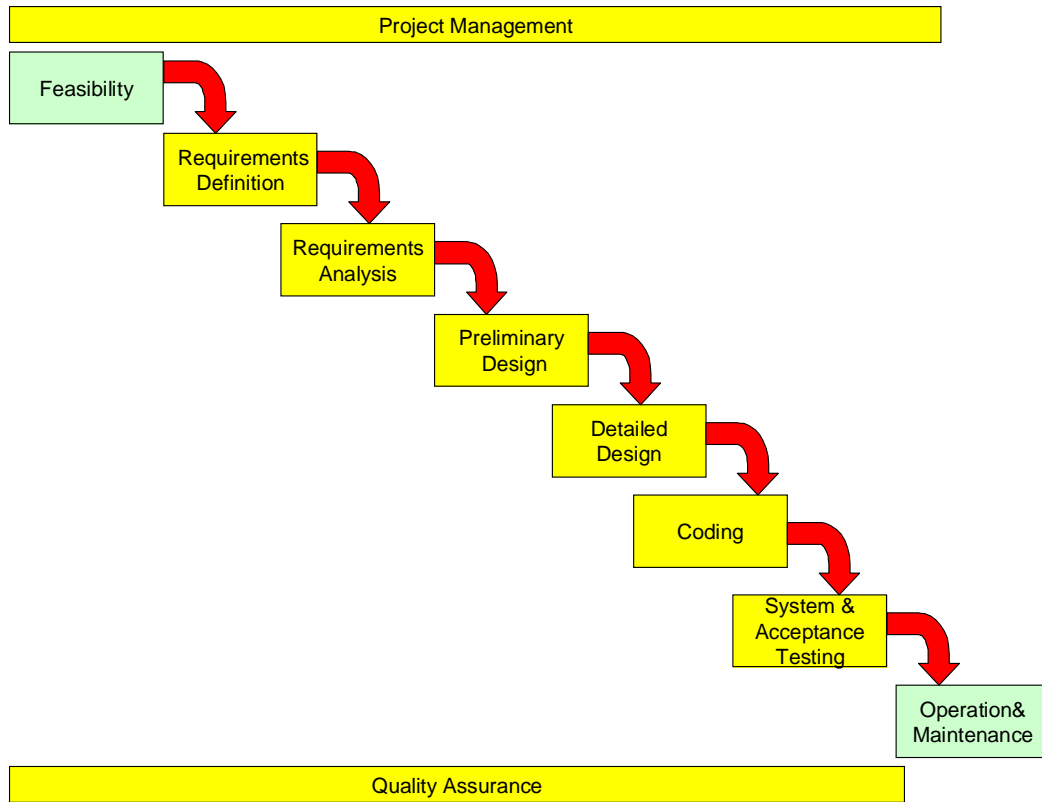
Discussing the merits or detail of any particular Software Development Life Cycle is not within the scope of this book, but most of the recommendations outlined will contribute to the success of any software project.

As Software Project Management processes oversee or underpin (depending on the point of view) software development life cycles, a typical 'waterfall' SDLC is illustrated on the next page to allow the rest of this book's recommendations to be placed in context.

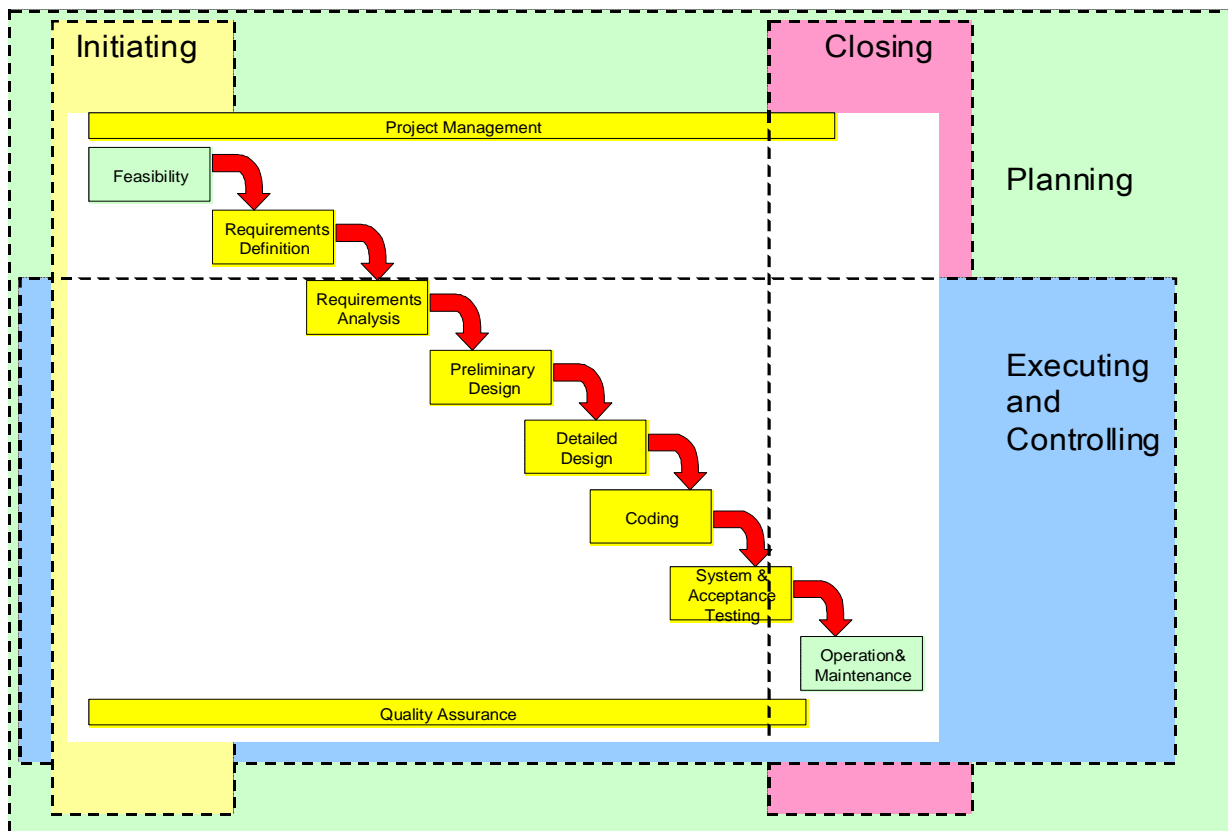
But what does this actually mean? What do these abstract models imply?

Well mainly that there is a practical order of events from the start of a project to the end. The fundamentals for success are obscured within these and other project models:

The Waterfall Software Development Life Cycle:



The Project Life Cycle phases overlayed with a Software Development Life Cycle:



2

Identify and Manage the Key Stakeholders of the Project

Stakeholders are anyone who has an interest in the Project, those who are actively involved in the Project, or whose interests may be affected as a result of the completion of the project.

Identify Stakeholders

The Key Stakeholders are required to be identified first, that is, the people who exert the most influence over the Project's objectives and outcomes.

The Project Manager must identify these stakeholders. Examples are Clients, Customers or Upper Management in a corporate environment.

Ideally one person should be identified or nominated as the Principal Stakeholder, the Project Sponsor.

The Project Sponsor may also be the 'End User' of the system, but often the Project Sponsor is the person who is accountable for the Project's success, whilst other stakeholders are the actual End Users of the System to be developed.

With the Project Sponsor's help, these stakeholders must also be identified.



Communicate with and Manage Stakeholders

The Project Sponsor exerts the main influence on the Project during the initial Project stages.

The Project Manager is striving to achieve the wants of the Sponsor using the Project as the framework for this.



The Project Manager must be able to challenge the Project Sponsor when necessary and it is imperative that a good working relationship is struck between the two parties.

Indeed, Project Manager, Project Sponsor and End Users (of the system) must all work in harmony in so far as possible.

Lack of good working relationships is nearly always down to poor communication.

The Project Manager must ensure that regular formal meetings are held with Project Sponsor and End Users.

This ensures there is always a forum to allow feedback and control of the Project with the people who matter most – the customers, the people using and paying for the system.

Different Sponsors/Clients will have differing approaches to the type of communication they want, but they all revolve around:

Is the Project going to deliver on time, is it going to cost more or less than estimated, will it do what is expected?

How to manage these aspects is covered in later sections.

Stakeholder Checklist

No	Item	Complete?
1	Has the principal stakeholder, the Project Sponsor, been identified?	
2	Have the key End Users of the system been identified?	
3	Are these stakeholders aware of the project and happy to proceed in their roles as key contributors to the success of the project?	
4	Has a regular meeting forum been set-up for these stakeholders to meet with the Project Manager?	
5	Is the Project Manager prepared to communicate key progress updates on cost, schedule and closeness of the solution to the vision?	



3

Identify the Goals of the Project

The foundations of all projects begin with identifying what the objectives are.

Not how these are to be achieved (not yet anyway), but as clearly as possible, what the aims are – what is the reason for the project to exist.

Together the Project Manager and the Project Sponsor (with agreement of the End Users) must gain a clear, high-level, understanding of the Project's goals.

If the Project Manager can't acquire a clear vision of what is required, or if these goals are not agreed with the key stakeholders, then this stage cannot be considered complete and pushing on with the project is much more risky.

Once the vision is understood (the objectives, goals, aims ...), and just as importantly what is not required (together these provide the scope of the project), then this Vision & Scope needs to be documented for final review and sign-off.

Only then is the Project Manager ready to have these objectives investigated and documented at a lower level of detail.

This definition is usually performed by Business or System Analyst roles within the Project's later activities, and results in a Requirements Definition document.



Control Change to the Vision, Scope and Software Requirements

One thing that can be relied upon, is that the Requirement Definition will change and often so can significant parts of the Vision and/or Scope of the project.



A process mechanism must be put in place to control these changes, so that when requirements change, there is a controlled process that ensures the system being built or to be built also changes.

The larger and/or the more complex the project, the more formal the effort needs to be in the control.

Common to all project change control processes, whatever the type, is the need to be pragmatic.

Project Requirement Definitions will change, so this fact needs to be built into Project estimates and schedules as an uncertainty tolerance (this is recovered in the appropriate section).

Therefore not all changes will affect the predicted cost or time taken to complete the project.

Indeed, most should not. Only significant deviations from the initial Vision should affect the stakeholders (by way of change to total cost or expected delivery date) – and most sensible stakeholders will realise this.

Project Goals Checklist

No	Item	Complete?
1	Is the high-level objective (the Vision) of the project understood by the Project Manager?	
2	Has the Vision been clearly documented and agreed by the key stakeholders?	
3	Has the process to control change been agreed and communicated to the key stakeholders?	



4

Organise and Plan the Project Activities

Another key to successful software delivery is to generate a realistic, usable plan, for this to be communicated to the team and for it to be followed.

The early stages of the project are critical to lay the foundations for effective project management and control.

Inexperienced Project Managers often dive in without taking time to think and plan things out.

Start by listing the phases and the deliverables of the project.

Usually software projects produce deliverables (i.e. outputs, documents, systems, products) from the following sequential phases:

- Requirements Definition
- Preliminary Design / Architectural Approach
- Detailed Design
- Development (Coding)
- System Testing
- Acceptance / Verification
- Delivery / Operation of System



Estimate and Schedule the Activities needed to Produce the Deliverables

The high-level goals and scope of what is to be achieved are now known. The full Project scope needs to be estimated to determine, effort; cost; system delivery date. The initial estimates of any Project are usually inaccurate, but they are necessary for the purposes of initial planning and for checking the viability/feasibility of the Project.

Distribution of Time and Effort over Phases:

Phase	% of Time Schedule	% of Effort
Requirements	12	6
Preliminary Design	8	8
Detailed Design	15	16
Programming	30	40
System Testing	20	20
Acceptance	15	10

To achieve a “first-cut” indicative estimate for the project use estimates gained from a previous project of a similar nature.

Otherwise, if historical estimates are not available, estimate how long the initial phase should take with how many people.

Then use the table above to derive the full time-estimate (most projects are estimated in person/days or person/months) into phases and cost.

Estimation of software projects is covered in detail in other books.

Add 10% to estimates for a change tolerance and a further 10% for contingency – a total of 20% to be added to both the number of staff required (from the % of Effort) and the amount of schedule time needed (from the % of Time Schedule).

Estimates for software projects must also include any other related costs, not just staff time/cost, e.g. equipment for the team (allowing the staff, for every task, to do their jobs), servers, and system software. If in doubt at this stage (and this is likely) find a similar project and make contingency provisions.

The activities must now be scheduled with start and end dates built around the order of sequential events (e.g. systems can't be developed before it's known what the software is to do).

There may be deadlines to be hit in which case the planning has to work backwards from the deadline given. A schedule contingency of 20% is prudent as already stated.

Determine and Organise the Project Roles

At a high-level a Milestone Schedule now exists of activities and their sequence.

Project progress is only made when people perform these activities and handover key deliverables to the next phase.

A Project Resource Plan is required to schedule people to the activity plans, to ensure that people are available to conduct the planned activities.

Therefore, across the same time span as the Milestone Activity Schedule, a schedule of roles (people with certain required skill and experience) is created.

This effectively shows the number of analysts, developers, testers etc that ideally need to be acquired for, or attached to, the Project in each week or month as relating to the Milestone Activity Schedule.

The Resource Plan is a request for people with certain skills at a certain time. It is an idealised concept at the moment. It is only an executable plan when it is confirmed that the estimated numbers of people with the correct skills are available and can commence on the dates required.

Where there are deviations from the ideal plan, these need to be reflected in the Project Resource Plan and the associated Milestone Activity Schedule – this may well affect the final delivery date of the system (as people may not be available to start the tasks at the earliest time suggested by the Project Schedule).

Deviations from any plans already agreed with stakeholders need to be agreed up front – there is no point in hiding schedule slippage that has occurred before the Project has even begun in earnest!

A draft Responsibility Assignment Matrix (RAM) is required. All the tasks on the Milestone Activity Schedule need to be allocated to an actual person by name (who will do or manage the task – is responsible for its completion).

The names will be found from the realised (as adjusted for when people are actually available, not when ideally they would be required) Project Resource Plan as noted by the Role required for the activity.

The Responsibility Assignment Matrix is to be used during the team set-up task by the Project Manager.

Planning Checklist

The latest versions of the Project Vision, the Milestone Activity Schedule and the Cost Schedule need to be agreed with the Project Sponsor.

Once agreement has been made to continue with the project, essentially based on total cost, delivery date of the system and expected scope of the solution (as compared to the benefits the Sponsor expects to gain on introduction of the system), then the planned activities can commence.

No	Item	Complete?
1	Has the project been estimated (staff time <i>and any associated costs</i>)?	
2	Has the estimate been transformed into a total cost and has this cost been agreed with the Project Sponsor?	
3	Has the estimate been transformed into a timeline schedule of high-level activities?	
4	Have the numbers and type of staff role been determined and added to the timeline schedule?	
5	Has the final timeline schedule (the indicative delivery date of the system) as adjusted by staff and other resource availability been agreed with the Project Sponsor?	
6	Has the Project Manager sufficient understanding and knowledge of the Project to brief team members on what is required?	
7	Has a draft Responsibility Assignment Matrix been created?	



5

Manage the Project Team

Contrary to some advice project heroes are vital to the success of a project. If the Project Manager can get very experienced, highly-skilled, hard-working and supportive staff on to the project (i.e. project heroes), then get as many of these as possible, every time.

A team must not be made-up of all junior or inexperienced staff; this is a recipe for failure even if their hearts are very firmly in the right place.

Make sure the project team is balanced. All 'heroes' on the team is great (but unusual), all juniors is not (but possible). Usually a balance has to be achieved in a corporate world and from a cost perspective.

Make sure a deputy is appointed to the Project Manager (and in due course, make sure he/she and the rest of the team know who the deputy PM is).

For small Project teams (up to say 10 people), get the team in a room all together for the initial briefing. If the full team is not allocated or available from day one of the project, then make sure that all the project team members get the same initial messages and communication *personally* as and when they join the team.

For larger projects, it is likely that it will be team leaders and senior analysts/consultants who are involved first. This is in fact a recommended approach for large capital projects, bring on the numbers of staff slowly, ramp-up across time.

Nevertheless, the same tenets apply; make sure all who join the project get initial verbal briefings of the Project's objectives and their roles within it (current information). For very large projects it may be that the Project Manager has to delegate this to Team Leaders, but the Project Manager must make sure this does happen.



Organise the Project Team

The following should be covered in an initial meeting (and re-covered for all those who join the project later):

- **The Vision of the project and the Scope, as agreed with the Project Sponsor and End-Users.**
- **The Milestone Schedule, as agreed with the Project Sponsor and End-Users.**
- **The Responsibility Assignment Matrix, as drafted by the Project Manager**
- **How the Project Team will be managed, as drafted by the Project Manager**



These are all key planning and communication activities, especially though the Responsibility Assignment Matrix (RAM) – everyone on the team should know what they are responsible for and for what everyone else is.

It is likely that the RAM may change under discussion (discussion on this is to be encouraged!), but don't let people 'off the hook' there and then, consider what everyone says in response to their allocated responsibilities and talk to them afterwards (not during the meeting) if issues arise.

If the above is followed, it is likely that the project team will feel involved, understand the overall goals, the timescales, what they have to look forward to in terms of being managed, and specifically, how they will play a part in its success.

The Project Manager will have started the project well and will already be gaining a sense of the team dynamics.

No matter what the Project type, how large they are, how experienced the staff may be, it is imperative that:

- regular focused meetings are held, and those actions that emerge from meetings are noted and formally assigned to individuals.
- roles are adjusted as appropriate as the team comes together and learns to work as a unit
- communication is clear and honest.
- mistakes are admitted and rectified.
- the team spirit is kept high.

Manage the Project Team Checklist

For large projects, it is time well spent to formally document how the project will be managed, resulting in a formal Software Management Plan (what management processes are to be employed; how the team(s) will be managed in regard to the development of the software; how the software will be technically constructed).

No	Item	Complete?
1	Does the team have the right blend of experience, skill and seniority?	
2	Does the team know who the deputy Project Manager will be in the PM's absence?	
3	Are all project members (who are available) invited to the Project initiation meeting	
4	Has the Vision, the Milestone Schedule and the draft Responsibility Assignment Matrix been distributed to the team	
5	Has a Project initiation meeting been held? Notes taken of actions to be followed-up?	
6	Has the Project Manager talked to the team members individually after the Project initiation meeting? Has their feedback been embraced without letting them 'off the hook' inappropriately?	
7	Have regular meetings with the team been set-up? Do the team members know they must attend?	
8	Has it been scheduled into the project plan to repeat the initiation meeting for everyone else who joins the project later (or responsibility has been delegated for this)?	
8	Are the project team sitting together (not always possible!) for the duration of the Project?	
10	For large projects, has a Software Management Plan been drafted?	



6

Manage the Project Activities

Most of the preparation work that has been done now comes into play and bears fruit.

If the Project Manager hasn't done the basics illustrated in the checklists, it will cause a problem at some point that could have been avoided.

Project Managers that have previously moved within the technical ranks may feel they know exactly how software is to be technically developed (the language, the platform, the hardware requirements etc). Perhaps in a corporate world, this is already laid out (it often is).

The Technical Approach and the Technical/Development staff on the project can (and should) directly influence how the project is to be technically driven. *Influence, but not control the Project!*

Special attention needs to be given to projects that are large, outsourced, technically complex or for those embracing a new technical way of working.

These types of projects require a Technical and/or Software Reference Architecture. These are key (specialised) deliverables that are needed early in the project.



Controlling and Tracking Progress

Review the agreed Milestone Schedule with the team in light of the technical approach.

It's usually best for the team's specialists or leaders (as appropriate) to derive and validate their own more detailed plans.

These need to support the high-level plan and be checked against it (to make sure it doesn't break the Milestone Schedule as it stands)



As part of the weekly meetings with the team, check the progress of the detailed plans and overall Milestone Schedule. Make sure things are on target. Where they are not, depending on the severity of the reasons why, then re-planning may need to take place. The Project Manager (with the team's help) must try to tackle the issues first before re-planning (see next Issues and Risks section).

Make the agenda as wide as it needs to be to cater for the scale of the project. Don't forget that decisions made in meetings (and their resultant actions) need to be recorded and distributed to the Project Team.

As part of the progress tracking, Project Sponsor and End User stakeholder expectations must be continually updated.

Typically projects will re-estimate & re-plan at the following points;

- At the end of every phase
- When significant changes are requested to the solution (via the agreed Change Control process)
- As requested by the Project Sponsor

The purpose of the planning review is to:

- Measure the success of objectives to date
- Review deliverables achieved
- Assess dates of, and risk to, future phases based on completed activities

Manage Issues and Risks

As the project progresses issues will emerge, these will be the things impeding progress against the plan.

As they emerge they need to be tackled. The first hurdle is knowing they exist and picking them up early, this is achieved through the staff management activities and the formal project tracking techniques.

Usually the issues (somewhat obviously) are that the planned activities finish late and start late.



Resolving issues requires the team's help. Some of the direct mechanisms open to the project manager are:

- Find another way of achieving the same result that avoids the issue
- Use up contingency (schedule and/or cost)
- Ask staff to work longer hours
- Get more staff on to the same activity (if possible)
- Agree not to deliver some features of the software at all
- Agree to deliver the main features on time, and the rest at a later date
- Move the delivery dates for the full system completely

Apart from the first two options, resolving problems implies compromise over individual task objectives of time, cost or functionality and can lead to mis-managed expectations – the Project Manager must be honest with the Project Sponsor and End Users – but there is no point in having 'knee-jerk' reactions to issues – give the team time to tackle or by-pass these before escalating to Sponsor. This is where it is imperative that the Project Manager has not scheduled activities too tightly and has built in cost tolerances.

There are whole methodologies based around identifying and tackling risks (issues that have not arisen yet, but could). There are some common general high-level lessons and risks that crop up in every software project (most are tackled by following these steps), so except for major capital projects, get on with the project activities and don't stew over what might be going to happen too much.

Concluding the Project

Once the project is coming close to delivering the system to the End Users, preparation needs to be made to ensure this transition is as smooth as possible.

Projects only exist to serve their aims, which in this instance is to deliver software to someone.

In a corporate world this is usually to another set of people who look after the continued well-being of the system and keep it operational for the end users.



Handover from the project team to the operational team should be conducted in the same inclusive fashion as the Project has so far been operating under – don't drop the ball at the last moment. The primary aim of Operational Acceptance or Final Customer Sign-off is to ensure that all the relevant departments and/or End Users have the people, tools, procedures and documentation in place to be able to carry out the operational use and maintenance of the system.

Formal mitigating plans must be developed (and tested) to ensure that deployment does not compromise any existing current system.

Finally most Project Managers and Teams want to learn from their activities with a 'post mortem' or more positively 'post implementation review' phase. The purpose of the exercise is to assess the success of the project as a whole. It is also an opportunity to identify any lessons learned and to carry these forward into other projects.

The purpose is to:

- get agreement from the Project Sponsor and End Users that the Project has achieved its aims
- discover cost or time overruns and reasons why
- highlight any significant issues that were faced
- review the project management and development practices employed

Manage the Project Checklist

No	Item	Complete?
1	Is a Software/Technical Architecture vision required? If so is it complete?	
2	Have detailed plans been created and referenced to the Milestone Schedule?	
3	Are the plans being tracked (progress) at weekly and one-on-one meetings?	
4	Are regular planning reviews schedules at end of phases?	
5	Is the team managing to circumvent and prevent issues?	
6	Are plans being made for system handover towards completion of the system?	
7	Is a review of the full project to be undertaken once complete?	



7

Create Clear, Accurate, Available Deliverables

Throughout the Project Life Cycle important supporting activities are required.

Much is made of quality assurance and most software developers will be used to code inspections and all levels of testing.

The most used and by far the best way to ensure a system is produced of good quality (it does what it is supposed to do and does it well) is to use peer reviews of key deliverable documents upstream of the development itself as well as during the code development phases and beyond.

A *peer review* is one of the most successful processes for driving up the level of software quality. It is a methodical approach to the inspection of documents and other project deliverables such as code to ensure that any defects are trapped and corrected as early as possible.

The inspection is carried out by the technical peers of the author and is moderated by an experienced chairperson to determine an agreed list of errors for the deliverable. Research has shown that up to 80% of errors found during system test could have been found by an effective peer review process.

There are two main types of deliverable that requires peer review; project documentation and code. For all of the deliverables described in this document it is prudent to use peer review. If it is to be used, peer review must be done by a group of peers in an organised meeting.

Ad hoc distribution of deliverables by email with an associated request for comments *will not work*.



Keep Documentation and Software Code Deliverables Available and Safe

All of the deliverables that are produced as part of the project need to be held centrally and be available to all stakeholders, especially the project team.

Appropriate measures need to be taken to ensure this information cannot be lost (regular copies of the information backed-up and held elsewhere – if the work completed to date is lost, so is the project).

Project Management software tools that help teams schedule, collaborate, keep in touch, track documentation, ensure version control etc. are a great addition and if they are available then they should be used.

Sometimes they must be put in place and used if team members are in different geographical locations and time zones!

One caution, do not change these tools across a Project unless there is absolutely no way to get out of this.

More time than enough is wasted in evaluating new tools and getting team members up to speed in their use. Project time should be spent on achieving project aims.



Deliverables Checklist

No	Item	Complete?
1	Is there a list of key deliverables? Are these deliverables being peer reviewed?	
2	Is all documentation and source code being kept secure and available to the full team	
3	Are appropriate software tools being used to help the team?	



8

End Game

Project Management fundamentals are not complicated or difficult; they just have to be understood and undertaken.

This book is an overview of the basics, but these fundamentals are often forgotten by experienced Project Managers who sometimes disappear into the depths of a methodology.

For people new to Software Project Management, whether from a technical background or not, if the spirit of these suggestions is followed (better yet, the letter), then there is a much greater chance of project success.

Further concepts and more detail can be found in the further books in this series from www.RockHardProjects.com.

This site also endeavours to support Project Managers of all experience levels with its forums, newsletters, training material and related software project information.

We hope you have found this book of value and it helps you achieve your objectives!

Regards from the,

www.RockHardProjects.com Team



The 7 Key Secrets to any Successful Software Project

No	Item	
1	Stakeholders. Focus on the key stakeholders of the project in this order – Project Sponsor, The End user, The Project Team, The Project Manager	S
2	Understand the overall goals (business and technical) of the project. Ensure the Vision & Scope (the business goal) of the project is documented at a high-level and is understood and agreed by the Project Manager (yourself), the Project Sponsor, The End Users and all of the Project Team members. Ensure a Software Architecture framework (the technical goal) is understood by all the Software Development Team	U
3	Communication is paramount. The larger the full project team the more effort must be placed in keeping everyone ‘in the loop’. Make sure regular formal update meetings are held with all stakeholders	C
4	Control the Roles and Responsibilities of the Project Team members – these must be formally allocated and monitored for effectiveness. Get people into the roles that suit them best, replace them if they do not become effective in any role that is open in the project.	C
5	Encapsulate Quality. Ensure key deliverables (documents and source code) are peer reviewed.	E
6	Software Requirements. Make sure the Software Requirements are documented and version controlled. By far the best way to get these agreed is to develop a wide but shallow prototype – show the End Users what the software interface could look like. A formal change request process must be understood and employed.	S
7	Scheduling is critical. Schedule the activities and monitor progress formally. Tackle Issues and Risks as they emerge, allocate ‘owners’ to the Top 10 Risks and Issues.	S

