

Part Two: Pre-Conditions for PM

2. External and Internal PM Pre-Conditions

2.1 The Business Plan

Projects, and in particular IT projects aim for a certain goal. This goal is typical a set of assumptions about the functional (technical) achievements of the project and the corresponding business advantages accompanying the realisation of the project. Depending on the project, this might be a direct and measurable economical benefit, or yield an indirect and perhaps a marketing advantage with respect to our competitors, or at least increasing the reputation against our customers, or supplier chain.

Today, it is a characteristic of an IT project, that the (technical) goals are pre-determined externally and do not depend on our behalf. Rather, we have to head for 'progress' because either competition forces us to renew our products constantly, or perhaps the public opinion or even standardisation's and requirements for conformity pushes us to re-define our product, thus new releases come out frequently.

Good examples for these requirements are the ever changing versions of Microsoft's Word; requiring ODF capabilities and actually providing OOXML instead the open standard. Other samples are Adobe's InDesign with now "Web 2.0" capabilities and or course the constant fight of all email software against malware and spam.

Thus, unlike a typically project, IT projects are driven from outside (except for a few). This makes management 'fed' about IT projects: "Why do we have to spend again money for this development; the old solution is working well". In turn, IT projects can be characterised to be

- market driven (by competitors, standardisation requirements, quality standards)
- cost driven (reduce production/operating/maintenance costs)
- technical driven (introduce innovation and/or substantial improvements).

Considering the volume of IT projects under the current market situation, the economy, the ratio between these dependencies is about 6:3:1 - to be optimistic. Or course, setting up IT projects may be driven by all three aspects. One example to mention is Apple's move from IBM PowePC platform to Intel CPUs.

2.1.1 Technical Evaluation and Market Analysis

The start of a successful IT project is a good understanding of the reasons why it should be carried out. The dependencies can be picked up from fig. 7:

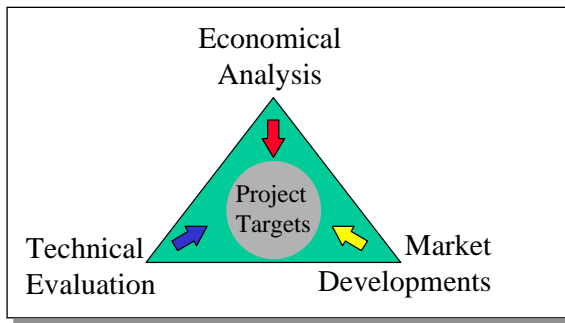


Fig 7: Dependency triangle for project targets

Upper management has to evaluate what fuels the forthcoming IT project and has to provide an impact analysis.

A Business Plan has to be carried out, which incorporates the current decisions and provides a guideline for the Project leader and the Project team - during every phase of the project. As indicated, while the project is running the goals and the dependencies *to* and *how to* finish the project may change, since neither upper management nor the Project leader has the definitive Crystal Ball to know the future.

On the other hand, a lot of IT projects focus on solutions for which the original assumption has already been superseded. In case of substantial deviations between the original targeted goals and those actually realised in the current project, a gap between the *real* project and the *believed* project opens up, which has important (negative) consequences for the project and perhaps makes the original Business plan invalid or obsolete. Certainly, this has an important impact of the recognition of the Project Leader, the Project team, and the project itself by upper management.

According to a recent analysis from the Standish Group [Standish2004], a significant percentage of IT projects is challenged or even fail:

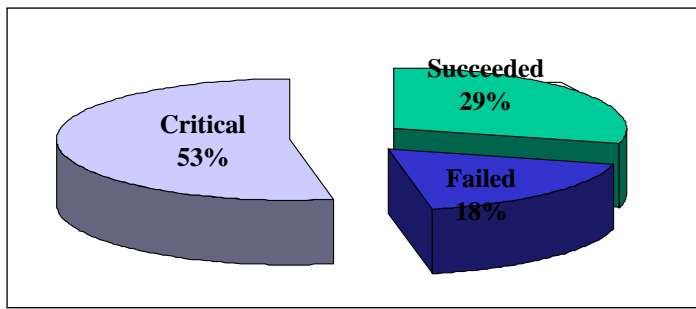


Fig 8: Chaos 2004 - Survey Results [Standish2004]

The reason for this situation can be characterised due to

- external factors: competition, moving market situation, technical risks
- internal factors: limited resources, challenging time-line, underestimated budget

While the external factors hardly can be controlled, it is not uncommon for the market leaders to pet the market such, the acceptance of the new products is raised and thus the risk of a failure is reduced and of course the market shares are maximised. The rollouts of new products from companies like Intel or Microsoft are often accompanied by such additional marketing mechanism.

2.1.2 Return-On-Investment Calculations

Let's assume an IT project has finished a certain software (product). From the company's perspective an IT project can be considered successfully, in case one of the following targets have been met:

- **Functional:** The product fulfils it's technical requirements.
- **Market share:** The products helps to improve the company's market position.
- **Economical:** The revenues realised by the product exceed the development costs (break-even point).

Any delays in the delivery of the product has significant impacts on all those targets. While functional and marketing issues can be considered 'soft' targets and the arising consequences are difficult to estimate, the economical dependencies can be estimated by means of a *Return-on-Investment (ROI)* calculation, as shown in figure 9.

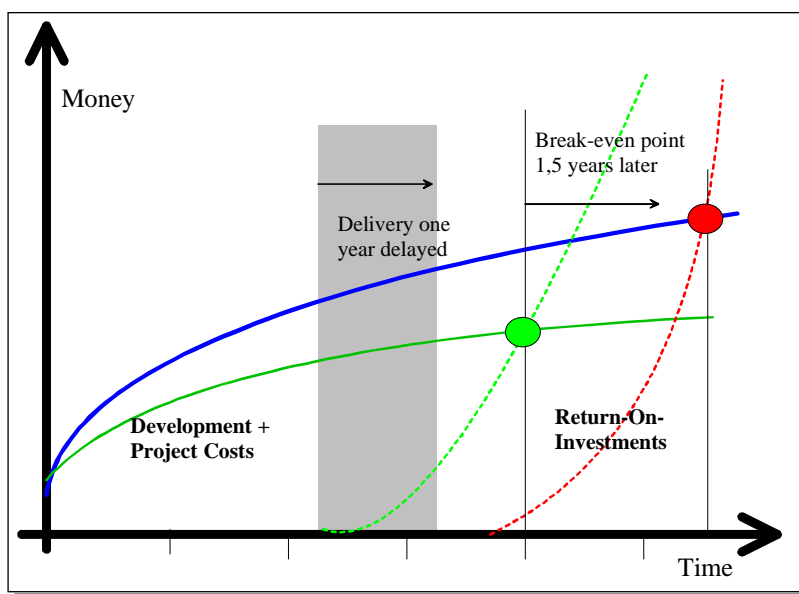


Fig 9: Return-on-Investment calculation [~ Litke2007]

Typical examples for such a calculation are the Roll-Out of Microsoft's Windows 98. Here it was clear from the beginning, that this update to Windows 95 (due to the difficult and almost complete change of the peripheral driver model [VMWD]) did not meet the required technical maturity, due to marketing constraints, the product had to be delivered in time.

2.1.3 The Mission Statement

Once the projects targets are defined and upper management has finally committed to provide the required resources (during the time, the project spans), the final 'Go' for the project has to be announced with the project's Mission statement.

The Mission statement has to fulfil the following criterion's:

- **Formal:** The wording has to be clear and concise; usual phrases have to be avoided (i.e. 'most important', 'lighthouse project'), the targets of the projects have to be named explicitly.
- **Content:** The project's targets have to be reachable; they have to be confined, thus do not depend on external circumstances.

The Mission statement should act as guideline for all project team members and of course the PLs. Upper management has to convince everybody that they stand undoubtedly behind the Mission statement and that the success or failure of the project is considered their own success or failure.

2.2 Project Management in Conflict

In general, any (IT) project has to deal with limited resources and competition among other projects. It is one of the initial task of project management to identify not only the risks but in addition the potential shortages and conflicts for his/her dedicated project. Conflicts arise in case the resources don't meet the project's requirement, as detailed in figure 10.

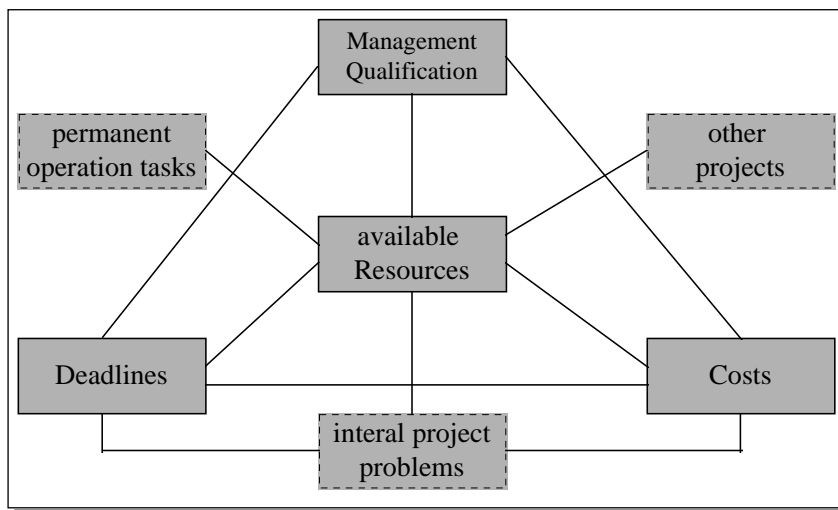


Fig 10: Sources for conflicts in projects [~ Litke2007]

While practically conflicts in particular among project (and staff) team members can not be avoided in the running project; conflicts with upper management are often more severe for any PL. Possible reasons may be:

- Project targets and their deadline are not clearly defined or deviations are not reported, thus hit upper management unprepared.
- Allocation of resources is not possible due to budget restrictions and/or the budget has already been used up.

Project managers should also consider during the definition phase of the project and before any commitments against upper management are carried out, what are the potential penalties in case the project does not succeed. In case no such penalties exist, the engagement for the project of in particular regular staff members may be reduced ('business as usual').

2.2.1 Balancing Budget, Deadline, and Quality

We have already discussed the impact of the Deadline with respect to the ROI (fig. 9). However, in IT project management it is common sense not only provide the product in time, but also to achieve a customer-acceptable quality. This includes not only to meet the functional requirements (as probably beacons by marking

already), but in addition to ship a product which is reasonable mature and considerable bug-free.

IT project management is always under pressure to achieve quality under the current project conditions. This is known as the *Magical Triangle of Project Management* [figure 11]:

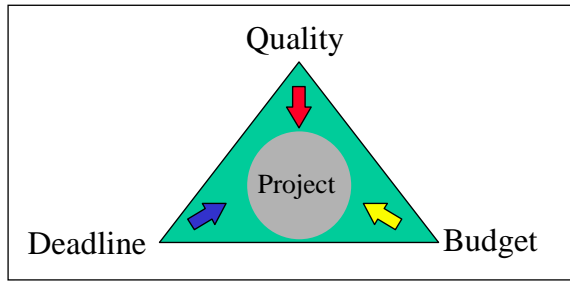


Fig 11: *Magical Triangle of Project Management* [~ LitkeKunow1998]

2.2.2 Effectiveness vs. Efficiency

In order to achieve the defined goals, project has to be effective. However, the limitation of resources requires in addition to organise the project efficient. After a project step or phase has been finished, it is task of the Controlling to evaluate both, Effectiveness and Efficiency. On the other hand, there are no 'golden rules' how to achieve efficiency initially, except for a very few:

- Avoid complex project structures and project plans; complexity is often aligned with imponderability.
- Allocate resources when it is required and don't delay them; delay is typically a synonym for less efficiency.
- Identify problems and solve them as soon as possible; otherwise problems will turn into risks.

Figure 12 shows a breakdown of potential factors impacting efficient project management [WieczorrekMertens2007]:

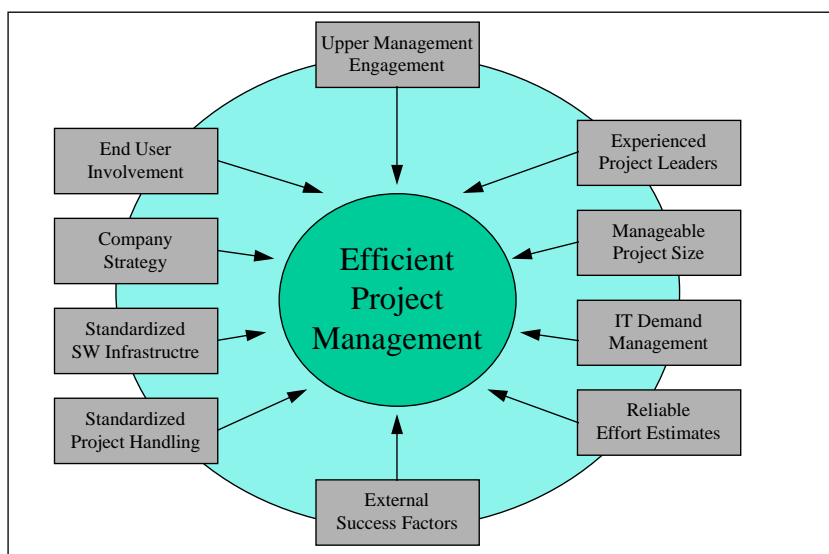


Fig 12: *Success factors for efficient project management* [~ WieczorrekMertens2007]

2.3. Project Conditions

Today's IT projects don't happen in a virgin environment; rather they are set up by companies which have already finished several other projects. Thus, the foreseen project leader has to make himself (or herself) familiar with the existing conditions the forthcoming project has to be carried out under [figure 12].

We have to discuss these issues regarding

- the existing technical framework,
- the organisational circumstances, and - most important -
- the administrative competencies and reporting structures.

2.3.1 Existing Technical Framework

While studying offers for IT project managers in the big newspapers often a particular development framework (like Visual Studio or Eclipse) is mentioned, for which the demanded project leader should have special knowledge. To be honest, either here a sub project leader (SPL) is required, or the human resource (HR) department has no good understanding what the tasks of project managers are.

Certainly, the knowledge of the technical software development methods is beneficial and a deep understanding here allows to judge the potential risks and perhaps quality of the code produces by the developers.

Development frameworks

Integrated Development Environments (IDEs) are typical for certain computer languages. Today's, those framework allows to collaborate; thus developer share a common repository and the access may be client/server or web-based. Most common today, in particular for the programming language Java a lot of development frameworks exist which allow an efficient usage of the object-oriented features of this language.

The development frameworks may additionally include software modelling techniques which in today's understanding is UML (Unified Modelling Language). Those frameworks have to stay ahead the current state and version of the software modelling language. From here class- and object hierarchies can be retrieved, which makes coding effective and efficient; though not guaranteeing that the code is efficient as well.

Frameworks are required as well in order external resources have to be included. This is the case if PL/SQL code has to be developed, which is only possible accessing directly the database instance.

Source Control

While occasionally IDEs include a (favourite) source control system, in general this is set up distinct. Typical choices are the client/server based systems Subversion or CrossVC to mention the public domain tools in the first place and IBM's (Rational's) ClearCase. Most of the systems provide a commonly shared repository to be accessed by UNIX and Windows clients, which is not uncommon for e.g. Java software projects.

Of course these systems differ significantly the way the code is represented to the developer and how to derive a 'release' out of the sources which is called the 'build' process.

Document Management

Though, it is not uncommon to store development documents in the Source Control system as well, more practical and more efficient is to use a dedicated system which allows to organise and to review the documents having one central repository.

While developers have access to and work with their IDE, architects, quality managers, and stakeholders included for revisiting the documents require a separate document management system automatically changing revision numbers, indexing the document, and providing a history of changes.

However, in practice often documents are send by mail, and the last version resides in a private folder without changed the document's revision.

In addition, a clear structure and reliable filing to the documents is required, not to search for documents to long. Thus the document filing should be realised in terms of URLs (Universal Resource Locators) which is commonly called hyperlinking. A good system achieves this not only per document but rather per section, in case the document templates are structured adequately and obey a defined hierarchy.

Quality Management

Quality Managers can be set up in the current project as sub project leaders or they may be organised separately. In this case, they report directly to upper management and don't need to follow the advises of the PL.

Typically the quality of software is tracked by a special tool, which is essentially a database application with a (Web) front-end to enter and to follow bug reports. In software development terminology, deviations of the tested software with respect to the specification is called a 'defect'. Such systems (ie. HP Quality Center) not only allow insert defects, and to define a 'lifeline' for it in terms of the attributes priority and status, but also to combine defects with the same source.

While the single defect is important for the individual developer, the regular obtained quality report, showing the distribution of defect priorities and correlating them with the sources tells much about the current state of achieved quality.

In order to obtain a complete picture, not only the individual reports are of interest, but rather how the development of defects took place, in terms what has been tested and how intensive the tests were done. A complete picture can only be obtained, if in addition the Design Documents are considered and the complete *Use and Test Cases* are filed.

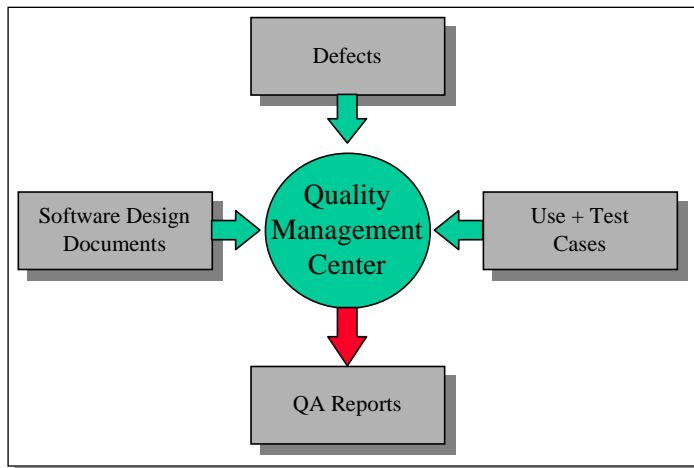


Fig 13: The role of the Quality Management Center as Hub

2.3.2 Existing Project Organisation

Entering an existing project team as PL, you first may want to see and judge how this team performed in the past. The best way is to check

- the way the Project Office works,
- how do people report their activities in terms of Time Sheets, and perhaps
- which projects were already completed and in what state.

Project Office

In order to achieve continuity a well organised Project Office (PO) is indispensable. Filing of documents, ordering of trips and hotel rooms, booking meeting rooms, account management, managing the team's agenda, preparing meeting documents, and many other activities are on the list of support activities.

The PO may act in addition as Center of (informal) contact. As project manager make yourself familiar with the organisation of the PO, streamline document filing and co-ordinate important agenda items with the responsible people here. Common and repetitious tasks should be delegated to the PO, which in turn has to understand it's task as *Project Assistance* and not just administration.

Time Sheets

While the project is ongoing, the project members have to report their activities on time sheet lists, in order to monitor the time required to complete a specific task. Of course, keeping the Time Sheets to-date takes attention and time and depends on the granularity of the reports.

Together with the SPLs, the organisation and the items to be subject for the time sheet has to be defined and communicated. Limits on the particular items (in terms of project hours) have to be included and the project's progress has to be correlated with the reported hours.

The time sheets are subject of budget control and thus have to be reported to upper management. Considering the financial foundation of the project, it is important to understand the following distinctive sources of the project budget:

- **Capex** (*Capital expenditure*): For the budget a certain fixed amount of money is allocated to be spent until the project is finished. In particular, external consultants (like developers) are hired under the condition of a limited budget or time frame.
- **Opex** (*Operational expenditure*): This money is taken out of the budget for general operations. Internal project team members may be funded by Opex means.

It is not uncommon for a budget to run on Capex and Opex means; however for internal reports, this differentiation might not be helpful. On the other side, a good on-line reporting system, which has hooks to the IDE and perhaps the QA system, relieves the team members from stupid recording of the actives and in addition improves the level of correctness. The scope here is, not to control the team members bur rather to identify budget and time shortages.

Project Heritage: Styles, successes and failures

It is certainly a good idea to gather information about successful and less successful projects realised by the company or under the current project organisation. The bottom line here is the quality of the information.

A good project organisation would provide a project completeness report; a badly organised project probably will leave the remnants of its work (directories and files on the file servers, code in the Source Control system) unchanged and visible for everybody.

In addition it is worthwhile to interview the former project members and perhaps managers for 'lessons learned'. For any company and organisation it is important to steadily improve the quality of project management. This can be done by internal and/or external course, allowing project members to achieve certificates, and file important documents and templates as guideline for future projects.

2.3.3 PLs Competence's

One of the most important tasks of (new) PL is to discuss and receive the necessary competence's from upper management. Bigger projects have at least an

- **administrative Project Manager** being responsible for risk management and in general controlling, and a
- **technical Project Manager** in charge of technical co-ordination and quality management. Typically, several Sub Project Leaders (SPLs) are foreseen, dedicated to co-ordinate fixed tasks and reporting PLs.
- In addition, a **Chief or Business Architect** can be established, who is responsible to co-ordinate development with Demand Management or Marketing and perhaps acts as co-ordinator with respect to customers or partners, to whom interfaces are designed.

Reporting Chains

Apart from 'managing' the project, the PL has to report the state of the project to upper management. Two types of reports shall be considered:

- **Standard reports**, defining in a defined and concise way the current status of the project, w.r.t. to particular sub-projects or fields.
- **Exception reports**, which need to be raised in cases the project runs suddenly out of plan, a particular risk has been identified, or the team (project) suffers due to a substantial crisis.

Reporting chains and acceleration schemes have to be clearly defined, in order not to deadlock the project. It is the bare responsibility of the PL to react 'well-behaved' during a crisis: regarding the project team and in particular concerning upper management. The current financial crises even of large international banks show clearly their incompetence setting up and utilising corresponding exceptional reporting channels.

Budget Planing, Recruiting, Ordering

While the budget for the project has to be allocated by upper management, the (efficient) spending of the money is one of the responsibilities of the PL. Responsibility might not coincide with competence. We already have discussed the different budget sources: Capex and Opex.

In case the projects does not exceed the allocated budget, certainly it is up the PL to decide where to use the money for. As outlined this situation characterises perhaps 2/3 of the project lifetime. Overspending is common for most IT projects. Whether upper management is willing to put more money into the project depends definitively on a convincing PL. It is certainly helpful, if the project management made up their homework and could provide that

- the project until now is under (budget) control,
- the financial risk of not spending the expected amount of money to (almost) finalised project is higher rather than allocating N more Dollars or Euro to the ongoing project.

Project Controlling

As discussed, the main function (according to upper management) of the PL is to control the ongoing project and to focus it in the interests of the company. In effect, controlling is a rational method to estimate risks; thus controlling and risk management's are twins. Of course there exist a thin line between real risks and reported risks. The example of the current bank and financial crises shows, that *real* and *reported* risks might not coincide:

- In order to estimate real risks one needs measures.
- In order to convince upper management one needs to present established methods.

In the situation of a PL, an underestimated risk may break the whole project, while an overestimated risk may block it. Here, a good relationship to upper management is one key to survive and eventually master the crises.

References:

[Standish2004]
<http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS>

[WiezcorrekMertens2007] Management von IT-Projekten, 2. Editon, Springer
Verlag Berlin Heidelberg, 2005 and 2007