

1. The six phases of project management

This chapter provides a sketch of the traditional method of project management. The model that is discussed here forms the basis for all methods of project management. Later chapters go into more depth regarding a model that is particularly appropriate for IT-related projects.

Dividing a project into phases makes it possible to lead it in the best possible direction. Through this organisation into phases, the total work load of a project is divided into smaller components, thus making it easier to monitor. The following paragraphs describe a phasing model that has been useful in practice. It includes six phases:

1. Initiation phase
2. Definition phase
3. Design phase
4. Development phase
5. Implementation phase
6. Follow-up phase

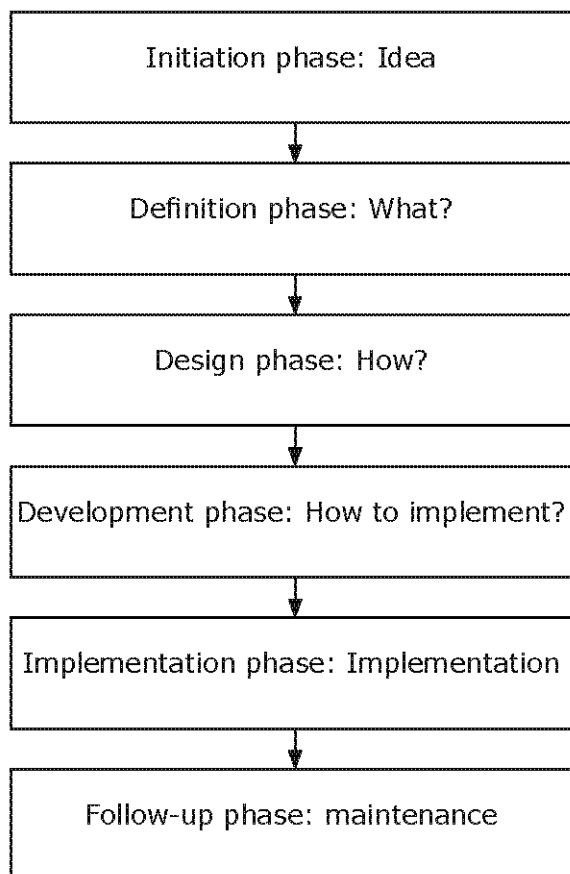


Figure 1: Project management in six phases, with the central theme of each phase

Initiation phase

The initiation phase is the beginning of the project. In this phase, the idea for the project is explored and elaborated. The goal of this phase is to examine the feasibility of the project. In addition, decisions are made concerning who is to carry out the project, which party (or parties) will be involved and whether the project has an adequate base of support among those who are involved.

In this phase, the current or prospective project leader writes a proposal, which contains a description of the above-mentioned matters. Examples of this type of project proposal include business plans and grant applications. The prospective sponsors of the project evaluate the proposal and, upon approval, provide the necessary financing. The project officially begins at the time of approval. Questions to be answered in the initiation phase include the following:

- Why this project?
- Is it feasible?
- Who are possible partners in this project?
- What should the results be?
- What are the boundaries of this project (what is outside the scope of the project)?

The ability to say 'no' is an important quality in a project leader. Projects tend to expand once people have become excited about them. The underlying thought is, 'While we're at it, we might as well ...' Projects to which people keep adding objectives and projects that keep expanding are nearly certain to go off schedule, and they are unlikely to achieve their original goals.

In the initiation phase, the project partners enter a (temporary) relationship with each other. To prevent the development of false expectations concerning the results of the project, it makes sense to explicitly agree on the type of project that is being started:

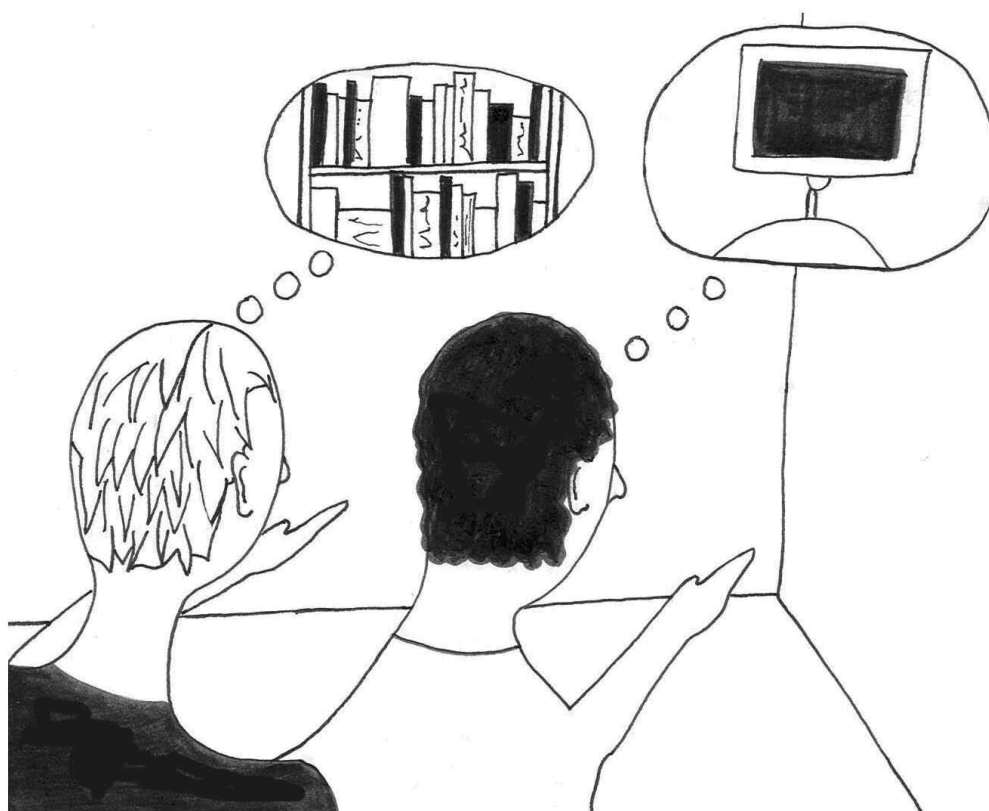
- a research and development project;
- a project that will deliver a prototype or 'proof of concept';
- a project that will deliver a working product.

The choice for a particular type of project largely determines its results. For example, a research and development project delivers a report that examines the technological feasibility of an application. A project in which a prototype is developed delivers all of the functionalities of an application, but they need not be suitable for use in a particular context (e.g. by hundreds of users). A project that delivers a working product must also consider matters of maintenance, instructions and the operational management of the application.

Many misunderstandings and conflicts arise because the parties that are involved in a project are not clear on these matters. Customers may expect a working product, while the members of the project team think they are developing a prototype. A sponsor may think that the project will produce a working piece of software, while the members of the project team must first examine whether the idea itself is technically feasible.

Definition phase

After the project plan (which was developed in the initiation phase) has been approved, the project enters the second phase: the definition phase. In this phase, the requirements that are associated with a project result are specified as clearly as possible. This involves identifying the expectations that all of the involved parties have with regard to the project result. How many files are to be archived? Should the metadata conform to the Data Documentation Initiative format, or will the Dublin Core (DC) format suffice? May files be deposited in their original format, or will only those that conform to the 'Preferred Standards' be accepted? Must the depositor of a dataset ensure that it has been processed adequately in the archive, or is this the responsibility of the archivist? Which guarantees will be made on the results of the project? The list of questions goes on and on.



'After the brain-storming session, all of the members of the "New Archive" project team were in agreement about the desired outcome'.

Figure 2: Expectations of a project (Illustration: Rachèl Harmsen)

It is important to identify the requirements as early in the process as possible. Wijnen (2004) distinguishes several categories of project requirements that can serve as a memory aid:

- Preconditions
- Functional requirements
- Operational requirements
- Design limitations

Preconditions form the context within which the project must be conducted. Examples include legislation, working-condition regulations and approval requirements. These requirements cannot be influenced from within the project. *Functional requirements* are requirements that have to do with the quality of the project result (e.g. how energy-efficient must an automobile be or how many rooms must a new building have?). *Operational requirements* involve the use of the project result. For example, after a software project has been realised, the number of malfunctions that occur must be reduced by ninety per cent. Finally, *design limitations* are requirements that involve the actual realisation of the project. For example, the project cannot involve the use of toxic materials or international partners for whom it is unclear whether they use child labour.

During the definition phase of a project that involved developing a web application for a consortium of large organisations, no agreements were made concerning the browser that would be supported by the application. The consortium assumed that it would be Microsoft Explorer, because it was the browser that 'everyone' used. The programmers created the application in Firefox, because they worked with the browser themselves and because it had a number of functions that were particularly useful during the development. Because most of the websites that are made for Firefox also look good in Explorer, the difference was initially not noticeable. Near the end of the project, however, the customer began to complain that the website 'didn't look good'. The programmers, who had been opening the site in Firefox, did not understand the complaint.

When the problem of the two browsers became clear, the programmers reacted defensively, 'Can't they just install Firefox? After all, it *is* free'. The organisations, however, were bound to the bureaucratic-minded system administrators who, for some possibly justified reason, refused to install Firefox in addition to Explorer. Even if they had wanted to install it, it would have involved a lengthy process, and there would have been extra costs for the time that the system administrators would have to spend on the task. It was ultimately decided that the application would have to be made suitable for Explorer. That involved considerable extra work, whereby the project ran even more behind schedule than it already had, and it was necessary to negotiate the extra costs. It was later discovered that the various organisations were working with different versions of Microsoft Explorer.

It is very important that all parties that are involved in the project are able to collaborate during the definition phase, particularly the end users who will be using the project result. The fact that end users are often not the ones that order the project perhaps explains why they are often ignored. The client, who pays for the project, is indeed invited to collaborate on the requirements during the definition phase. Nonetheless, the project result benefits when its future users are also invited. As a point of departure, it is helpful to make a habit of organising meetings with all concerned parties during the definition phase of a project.

During the development of an educational video game, the users (young people) were involved in the project only at a later stage. When the game was nearly completed, a group of young people was asked to test the game. Their initial assessments appeared mild and friendly. When pressed, however, they admitted that they had actually found the game 'extremely boring' and that they would 'certainly not play it themselves'. Had these young people been involved in the project earlier, the game would probably have been a success. As it stands, the game remains nearly unused on an Internet website.

The result of the definition phase is a list of requirements from the various parties who are involved in the project. Every requirement obviously has a reverse side. The more elaborate the project becomes, the more time and money it will cost. In addition, some requirements may conflict with others. New copy machines are supposed to have less environmental impact; they must also meet requirements for fire safety. The fire-safety regulations require the use of flame-retardant materials, which are less environmentally friendly. As this illustration shows, some requirements must be negotiated.

Ultimately, a list of definitive requirements is developed and presented for the approval of the project's decision-makers. Once the list has been approved, the design phase can begin. At the close of the definition phase, most of the agreements between the customer and the project team have been established. The list of requirements specifies the guidelines that the project must adhere to. The project team is evaluated according to this list. After the definition phase, therefore, the customer can add no new requirements.

A part of a new exhibit in a museum was comprised of a computer installation, the creation of which had been project-based. Because there had been no definition phase in the project, no clear agreements between the museum and those responsible for building the installation had been made. When the computer for the installation broke down halfway through the exhibit, the museum assumed that it would be covered by the project's guarantee. The project team had a different opinion. Negotiations between the directors were necessary in order to arrive at an appropriate solution.

Design phase

The list of requirements that is developed in the definition phase can be used to make design choices. In the design phase, one or more designs are developed, with which the project result can apparently be achieved. Depending on the subject of the project, the products of the design phase can include dioramas, sketches, flow charts, site trees, HTML screen designs, prototypes, photo impressions and UML schemas. The project supervisors use these designs to choose the definitive design that will be produced in the project. This is followed by the development phase. As in the definition phase, once the design has been chosen, it cannot be changed in a later stage of the project.

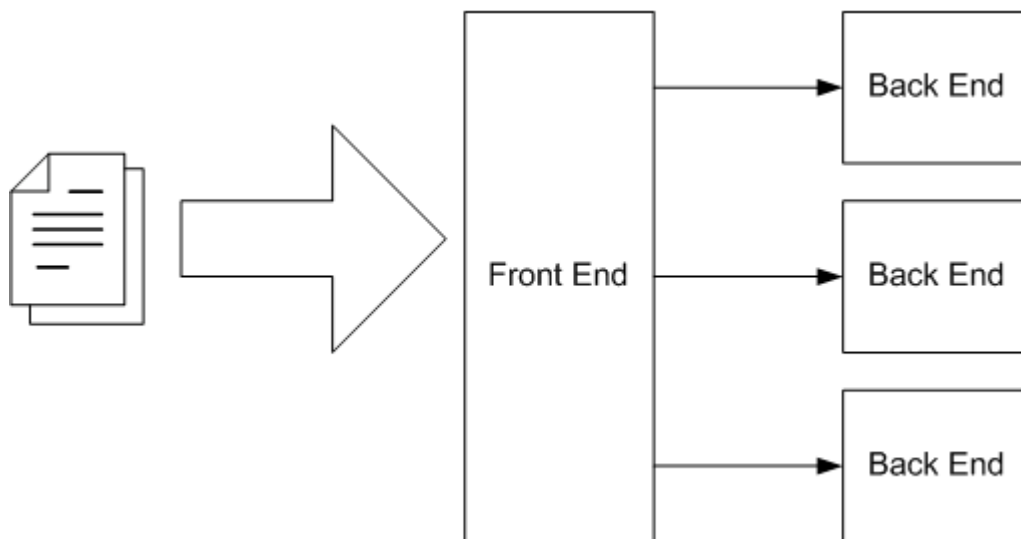


Figure 3: Example: Global design for the DANS Architecture Archive

In a young, very informal company, the design department was run by an artist. The term 'design department' was not accurate in this case; it was more a group of designers who were working together. In addition, everyone was much too busy, including the head of the department.

One project involved producing a number of designs, which were quite important to the success of the project. A young designer on the project team created the designs. Although the head of the design department had ultimate responsibility for the designs, he never attended the meetings of the project team when the designs were to be discussed. The project leader always invited him, and sent him e-mails containing his young colleague's sketches, but the e-mails remained unanswered. The project leader and the young designer erroneously assumed that the department head had approved the designs. The implementation phase began. When the project was nearly finished, the result was presented to the department head, who became furious and demanded that it be completely redone. The budget, however, was almost exhausted.

Development phase

During the development phase, everything that will be needed to implement the project is arranged. Potential suppliers or subcontractors are brought in, a schedule is made, materials and tools are ordered, instructions are given to the personnel and so forth. The development phase is complete when implementation is ready to start. All matters must be clear for the parties that will carry out the implementation.

In some projects, particularly smaller ones, a formal development phase is probably not necessary. The important point is that it must be clear what must be done in the implementation phase, by whom and when.

Implementation phase

The project takes shape during the implementation phase. This phase involves the construction of the actual project result. Programmers are occupied with encoding, designers are involved in developing graphic material, contractors are building, the actual reorganisation takes place. It is during this phase that the project becomes visible to outsiders, to whom it may appear that the project has just begun. The

implementation phase is the 'doing' phase, and it is important to maintain the momentum.

In one project, it had escaped the project team's attention that one of the most important team members was expecting to become a father at any moment and would thereafter be completely unavailable for about a month. When the time came, an external specialist was brought in to take over his work, in order to keep the team from grinding to a halt. Although the team was able to proceed, the external expertise put a considerable dent in the budget.

At the end of the implementation phase, the result is evaluated according to the list of requirements that was created in the definition phase. It is also evaluated according to the designs. For example, tests may be conducted to determine whether the web application does indeed support Explorer 5 and Firefox 1.0 and higher. It may be determined whether the trim on the building has been made according to the agreement, or whether the materials that were used were indeed those that had been specified in the definition phase. This phase is complete when all of the requirements have been met and when the result corresponds to the design.

Those who are involved in a project should keep in mind that it is hardly ever possible to achieve a project result that precisely meets all of the requirements that were originally specified in the definition phase. Unexpected events or advancing insight sometimes require a project team to deviate from the original list of requirements or other design documents during the implementation of the project. This is a potential source of conflict, particularly if an external customer has ordered the project result. In such cases, the customer can appeal to the agreements that were made during the definition phase.

As a rule, the requirements cannot be changed after the end of the definition phase. This also applies to designs: the design may not be changed after the design phase has been completed. Should this nonetheless be necessary (which does sometimes occur), the project leader should ensure that the changes are discussed with those involved (particularly the decision-makers or customers) as soon as possible. It is also important that the changes that have been chosen are well documented, in order to prevent later misunderstandings. More information about the documentation of the project follows later in this handbook.

Follow-up phase

Although it is extremely important, the follow-up phase is often neglected. During this phase, everything is arranged that is necessary to bring the project to a successful completion. Examples of activities in the follow-up phase include writing handbooks, providing instruction and training for users, setting up a help desk, maintaining the result, evaluating the project itself, writing the project report, holding a party to celebrate the result that has been achieved, transferring to the directors and dismantling the project team.

The central question in the follow-up phase concerns when and where the project ends. Project leaders often joke among themselves that the first ninety per cent of a project proceeds quickly and that the final ten per cent can take years. The boundaries of the project should be considered in the beginning of a project, so that the project can be closed in the follow-up phase, once it has reached these boundaries.

It is sometimes unclear for those concerned whether the project result is to be a prototype or a working product. This is particularly common in innovative projects in which the outcome is not certain. Customers may expect to receive a product, while the project team assumes that it is building a prototype. Such

situations are particularly likely to manifest themselves in the follow-up phase. Consider the case of a software project to test a very new concept.

There was some anxiety concerning whether any results would be produced at all. The project eventually produced good results. The team delivered a piece of software that worked well, at least within the testing context. The customer, who did not know much about IT, thought that he had received a working product. After all, it had worked on his office computer. The software did indeed work, but when it was installed on the computers of fifty employees, the prototype began to have problems, and it was sometimes instable.

Although the programmers would have been able to repair the software, they had no time, as they were already involved in the next project. Furthermore, they had no interest in patching up something that they considered a trial piece. Several months later, when Microsoft released its Service Pack 2 for Windows, the software completely stopped functioning. The customer was angry that the 'product' once again did not work. Because the customer was important, the project leader tried to persuade the programmers to make a few repairs. The programmers were resistant, however, as repairing the bugs would cause too much disruption in their new project. Furthermore, they perceived the software as a prototype. Making it suitable for large-scale use would require changing the entire architectural structure. They wondered if the stream of complaints from the customer would ever stop.

The motto, 'Think before you act' is at the heart of the six-phase model. Each phase has its own work package. Each work package has its own aspects that should be the focus of concentration. It is therefore unnecessary to continue discussing what is to be made during the implementation phase. If all has gone well, this was already determined in the definition phase and the design phase. For a more detailed description of the six-phase model and the task packets for each phase, see Wijnen (2004) and Kor (2002).