Der Rechnungshof

Success in information technology projects[1]

Introduction

(1) IT is a major prerequisite for efficient administration work. Today, desktop PCs with office software, electronic communications, electronic filing and archiving, IT systems for accounting, materials management and HR management are being used by government authorities both at the federal and regional levels.

(2) IT projects are subject to the influence of a large number of critical factors. Not paying enough attention to these factors frequently leads to major delays, budgets being exceeded, or even projects being aborted.

A study conducted by an international IT consultant on roughly 200 000 IT projects showed that, in 2002, only 34% of all these IT projects worldwide managed to stay within time and budget as planned. 15% of the projects failed, while the remaining 51% could be completed only by increasing the time and budget frameworks or restricting the planned scope of functions.

(3) In 2003, federal IT expenditure - including outsourcing to external IT service providers - amounted to about € 339 million, with a further € 110 million spent at the regional government level. Legal entities whose financial operations also have to be audited by the Austrian Court of Audit, such as local authorities in communities with more than 20 000 inhabitants, social insurance institutions, health care institutions, professional associations (chambers), parts of successor companies of the former state-owned postal and telegraph services, other former government-run entities, as well as the Austrian Railways, accounted for additional IT spending.

[1] Bruno Walter, Austrian Court of Audit

The Court of Audit estimates public-sector IT expenditure in 2003 to have amounted to more than € 1 billion.

(4) In the course of financial audits, the Court of Audit identified two key problem areas in the context of IT projects.

- IT projects that significantly exceeded the approved time and budget frameworks or did not provide the planned scope of functions, and
- IT projects that, in spite of high costs, were not conducted as a project should be, i.e. without adequate structuring.

Phases of an IT project and potential problems

Using the sequence of phases in an application development IT project structured according to best practices as an example, we will describe the purpose of each project phase as well as the problems most frequently identified during financial audits.

Definition of objectives

(1) The definition of objectives is what guides the whole project and provides a benchmark for measuring its success. Its purpose is to define the content, purpose, and benefit of the project.
(2) Potential problems:

- lack of concrete objectives;
- lack of benchmarks to verify goal achievement;
- decisions on technical solutions without comprehensive knowledge of project requirements.

Initial identification of requirements and resources

(1) The project approval process typically includes an estimate of cost, timeframe and human resources needed. This necessitates the prior compilation of a list of requirements. The functional requirements can be translated into development effort figures on the basis of either empirical values gathered in previous projects or technical methods for effort estimation, e.g., the function point method2.

(2) Potential problems:

- list of requirements is compiled only for parts of the project, with the rest being extrapolated;
- requirements are described in too much detail in some sections;
- incorrect (typically too low) effort estimation due to lack of experience.

Project functions

(1) It is absolutely vital for the project to define functions (contract awarder, steering committee, contractor, head of analysis team, analysts, head of programming team, programmers, system technicians, project management, controlling, testing) and appoint people to fill them. The project team members must be assigned to report to the project manager for the entire duration of the project.

---

2 A method for estimating the effort required for software projects on the basis of the program functions to be implemented.

(2) Potential problems:

- project functions are not assigned; e.g., controlling, management, head of analysis, head of programming;
- no definition of and/or non-compliance with decision-making structures between contract awarder, steering committee and contractor;
- project manager is not given authority over project team members;
- project team members are not fully assigned to the project, but handle the project in addition to their usual work;
- lack of expertise;
- lack of experience in project management;
- teams are formed without taking account of inter-personal and intra-personal needs.

Approval

(1) The approval process has to cover the requirements, resources, time and budget as well as the assignment of project functions to team members. Escalation mechanisms and/or reactions to any exceeding of planned resource thresholds for each section must be agreed upon.

(2) Potential problems:

- not all the key project data as mentioned are fully covered by the project approval process;
- no agreements on what to do when approved resources thresholds are exceeded;
- no definition of milestones to enable output and input to be compared against planned values.

Requirements analysis

(1) In this phase, which accounts for about 25% of overall project effort, it is necessary to fully identify the requirements in the various departments affected by the IT applications. Requirements should be classified by subject matters and prioritized by frequency of use and/or development cost/benefit ratio.

In the case of functions which are rarely used, but need a lot of effort to implement, as well as in special or isolated cases, an IT implementation within the scope of the overall project should be avoided.

After the requirements analysis, the full list of requirements, which is now available for the first time, has to be used as the basis for calculating subsequent effort in terms of HR, time and budget and to compare these figures against the values planned so far.

(2) Potential problems:

- requirements analysis is initiated without a work breakdown chart;
- not all the affected departments are involved;
- the requirements are not adequately structured;
- all the individual requirements are taken into consideration without reflection;
- project costs and resources are not re-calculated after the requirements analysis.

Application design

(1) In this phase, which likewise accounts for about 25% of overall project effort, it is necessary to translate the functional requirements into a technical workflow. This includes first of all the definition of system and software requirements as well as the decision for a particular software development environment and the definition of interfaces. All of this furnishes the basis for the software design as such.

To ensure a consistent level of quality for the software design, which is drawn up by various team members, it is possible to use a set of sample solutions and routines (after consultation with the head of quality assurance). As a rule, it is useful to split the design up into more or less independent modules and divide these in turn into sections with defined milestones. This provides a clear structure for the work to be performed and makes it easier to monitor project progress by comparing the actual status with the planned values.

(2) Potential problems: application design is initiated without a work breakdown chart;

- lack of expertise for technical decisions taken in the run-up to the project;
- decision for software tools with low market penetration;
- decision for software tools where no in-house experience is available and which therefore require a lot of familiarization and training;
- no uniform solution methods for software design because of lacking coordination between analysis and quality assurance;
- lack of modularity in application design makes it difficult to work in parallel and to monitor progress.

Programming

(1) Programming, too, has to be structured on the basis of a work breakdown chart and be standardized by means of programming guidelines. As a rule, the software design will be created in a development environment that supports graphical representations of logical sequences. This makes it possible to subsequently generate the program code automatically, in which case programming is reduced to merging the various process routines and programming the required interfaces. The automatically generated code sections have to be commented to ensure the future serviceability of the program.

If programming languages without a program generator are used, the program code has to be structured and adequately documented. Subsequently, the respective developers have to test the program modules against the defined requirements.

(2) Potential problems:

- programming is initiated without a work breakdown chart;
- programming without guidelines, without adequate structuring and sufficient documentation;
- use of automatically generated program code without sufficient documentation;
- no individual testing of each module.

Test, acceptance, and integration

(1) Testing also requires a work breakdown chart with test data and coverage of all the individual functions. The subsequent test cycle consists of at least three sections.
System testing checks whether or not the overall program meets the requirements. Integration testing checks how the program is embedded into existing IT systems; load testing measures the system's behaviour under normally expected, but also under exceptional loads.

After successful testing, the application can be acceptance-tested on the basis of a comparison of actual against requested performance (functional objectives) and of a verification of time and budget objectives.

(2) Potential problems:

- - no work breakdown chart for testing;
- - not all the program functions are tested;
- - no load tests based on real-life conditions;
- - a reduced acceptance procedure for time and budget reasons.

IT project examples

During the past few years, the Court of Audit has examined a number of IT projects. The ones presented here refer to the in-house development of special application software (case studies 1 and 2) and to the outsourcing of software development (case study 3). Case study 1 lists all the key problem areas mentioned above, while cases studies 2 and 3 refer only to three problem areas that had a particularly negative impact on the project.

Case study 1

1995 saw the start of an in-house project for application software development, with a budget of € 7.85 million. Development work was distributed over 5 locations in Austria, with a planned timeframe of 3 years. In June 2002, work on the project was suspended due to inadequate project progress combined with continuously increasing cost. The project had been completed to a degree of only about 70% while having caused costs of about € 14.71 million.

The Court of Audit considers the following problems to have had the strongest impact on the project:

- The requirements had been defined only for parts of the project; time and budget estimations had been based on an extrapolation of these results without knowledge of the functional requirements for the rest of the project.

- The controlling, head of analysis and head of programming functions had not been planned for and were thus not provided for.

- The project team members distributed over 5 locations in Austria did not report directly to the project manager. Availability rates for project team members, including the project manager, were less than 50%; employees were not assigned fully to the project, but worked for it in addition to their other duties.

- The project was approved without knowing the project start date; a project end date was not defined during the approval process. No procedures for what to do if approved resources were exceeded and no milestones for actual-planned comparisons were defined.
- The requirements analysis included all the individual requirements without verification of their cost-benefit ratios.

- The application design phase was started without a work breakdown chart for the analysis team.

- A software development environment with low market penetration was chosen; no experience with this product was available, which resulted in long familiarization periods. As the vendor has discontinued this software, it now has to be serviced in-house.

- The actual status (progress) of development work was not regularly monitored.

- Programming took place without a work breakdown chart and without timeframes for each module.

- Upon recommendation by the Court of Audit,

- the project was split up into independent parts and distributed over new locations, which were each fully responsible for the part assigned to them;

- the project was placed under external project management;

- IT support and external controlling were set up additionally;

- work breakdown charts for analysts and programmers were introduced; and

- milestones were defined to make the project amenable to controlling through actual/planned comparisons.

Case study 2

A comprehensive program system for the administration of specific personal data was to be developed within the framework of 16 individual contracts. A program platform to serve as basis for further development work was developed and tested under three individual contracts worth a total of € 6.6 million.

The subsequent contract for the development of a prototype failed because the previously developed platform relied on outdated, no longer adequate technology. Four additional contracts for the development of a new basic system led to a six-fold increase of the originally proposed budget of € 1.1 million, with a delay of more than five years.

The basic system was to serve as a platform for individual application programs for the existing 68 organization units. The requirements had already been identified. After three years and due to the extraordinary delay incurred, an amendment agreement had to be concluded to bring application development down from 68 solutions to three standardized categories. In spite of this reduction of functional scope, the cost for the application programs rose from a budgeted € 6.5 million to € 8.4 million.

- The Court of Audit regards the following problem areas as particularly negative for the project:

- The contracts for the development of a program system were not based on a technical concept.

- The requirements for the technical basic system were not analyzed, and prototyping was based on a technically inadequate system.

- The unreflected consideration of all user requirements submitted by 68 organization units did not take account of the more or less identical workflows in these units and resulted in exceptionally high development effort.

Case study 3

An IT project worth € 96.55 million with a planned runtime of twelve years was to equip 4500 employees with desktop PCs running office software plus a new, specially developed application program, which was to be specifically adapted to the requirements of each individual unit. To provide for an optimal implementation of these individual requirements and to ensure high acceptance rates for the new software among its users, a one-year period for adaptations was planned for each unit.

Several changes in PC operating system and office software made by the vendors led to considerable and unexpected additional effort for re-installation and data conversion. The need for a change in product for the application program and a subsequent version change caused high programming effort and several years of delay. Work in the central units is expected to be completed by 2005, which means a four-year delay against initial plans.

The Court of Audit regards the following problem areas as particularly negative for the project:

- The planned project duration of 1 years did not take account of the 3-year technology change cycle in the PC and office software field and all the subsequent effort this entailed.

- The combination of a variety of IT applications of different levels of complexity led to different degrees of completion in each phase and made it more difficult to complete the software installation process in each unit.

- The one-year adaptation period planned for each unit neither took account of similar workflows nor limited the requirements to a reasonable and feasible extent.

IT project management

There is a lot of specialized literature describing how IT projects should be organized. The two main versions are the lifecycle model and the process model.

Lifecycle model

The lifecycle models divides the process into discrete, chronologically subsequent phases. Once a phase is completed, its results are summarized. The next phase can be initiated only after a positive decision that the goals of the previous phase have been reached.

Process model

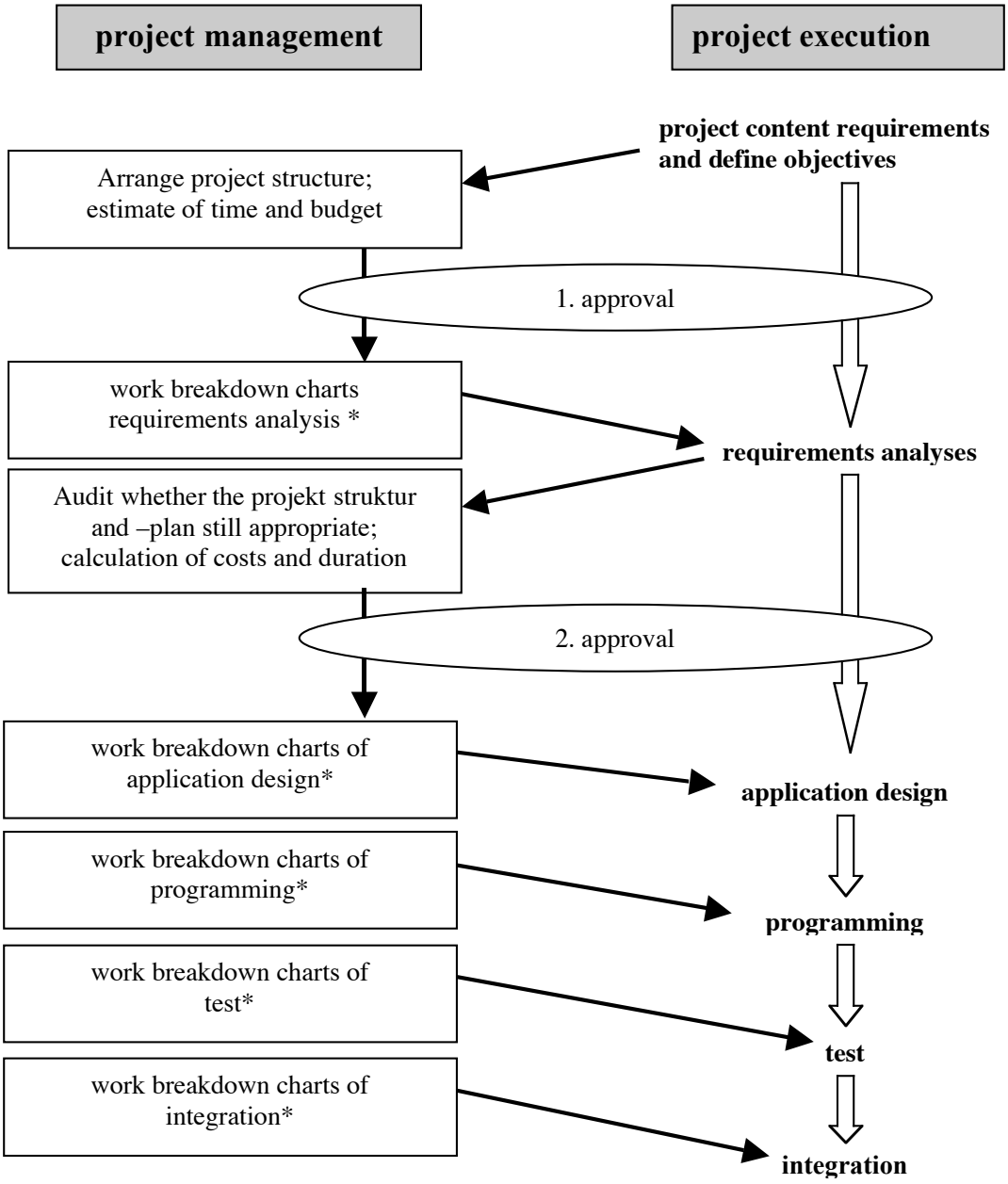
The process model adds factors such as project, configuration and quality management to the principles of the lifecycle model. The process model of project organization provides concrete instructions for each single activity, but does not define the chronological sequence of activities and how they interact. For this reason, it frequently occurs that the process model is used for project organization, but a lifecycle approach is applied to the sequence of the individual activities.

While the Court of Audit considers these theoretical models adequate, though not easily translatable into practice, it regards the choice of a model for handling an IT project as a major challenge in itself.

Court of Audit project model

For the purposes of its financial audits, the Court of Audit has merged the characteristics of all of these models into a model of its own. This model combines elements from the process and lifecycle models to form a separate, very simple structure that lends itself as a project execution guideline for immediate application. It is not intended to replace the project approval procedures and project structures already developed by government authorities, but represents only one of many possible structures, which can also be used for controlling purposes.

Court of Audit project model flowchart



*Work breakdown;
definition of resources for each part and agreement on milestones

The work breakdown charts for requirements analysis, application design or programming divide the work to be performed into modules and sections. Detailed planning for each section must include maximum HR needs, timeframe and budget, and a suitable milestone. The documentation of work performed and the presentation of results for each milestone make it possible to compare output and input against the respective detailed plans.

The Court of Audit relies on this model to evaluate the project phases in a given project. In this context, it makes no difference whether or not these project phases are specifically mentioned in the approved project plan. What is essential is that the functional tasks of each project phase as described in the model are fulfilled during the course of the project being audited.

Multi-level approval procedure

The Court of Audit model is based on a double approval of the project: first, on the basis of the defined objectives and the estimated costs; second, on the basis of a full analysis of the requirements and subsequent renewed project time and budget calculations. This approach has the following key advantages:

- In the requirements analysis phase, which consumes about 25% of project resources, it is possible to consider new or supplementary requirements - as compared against the status at project initiation - without a cumbersome change request procedure.
- After the requirements analysis phase, it is possible to agree to limit additional requirements or change requests as far as possible; this does not apply to absolutely necessary changes, such as caused ,for example, by errors in underlying documents or changes in the legal situation.
- After completion of the requirements analysis, it is possible to calculate project time and budget so that reliable figures are available for further project approval.

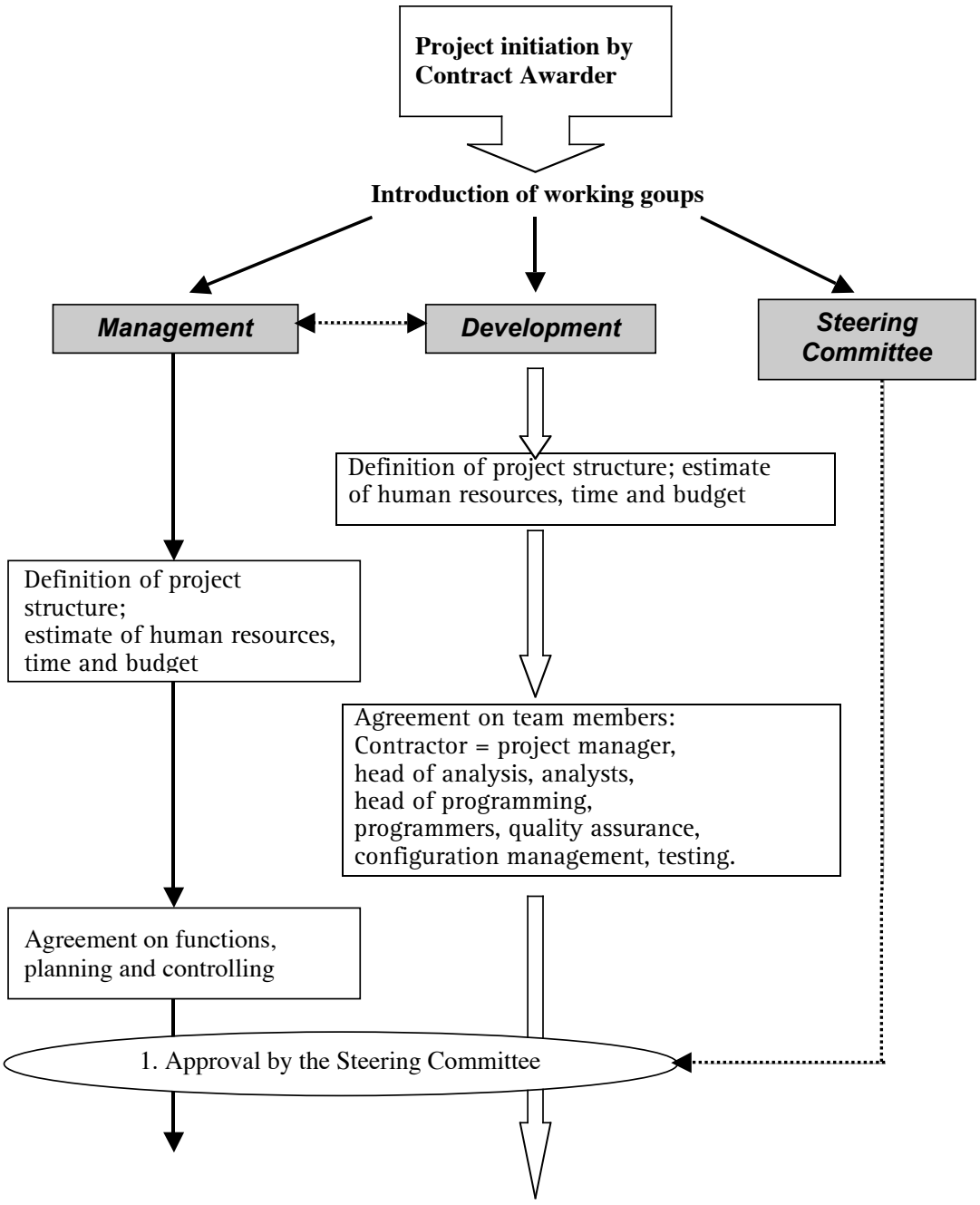
The drawback of a two-level approval procedure lies in the fact that a final decision on whether or not to continue the project can be taken only after about 30% of overall project resources (5% for project initialization and 25% for the requirements analysis) have been used up. However, this disadvantage is outweighed by the fact that even if project continuation is not approved on account of too high cost in the future, the loss or damage suffered is lower than if the same project would be aborted later on because it had exceeded its budget.

The contents of the individual project phases are the same as already described in the section on IT project phases. The Court of Audit sees the key problem areas in project initiation, application design, and controlling. Accordingly, the model has been extended for these phases.

Project initiation

The project initiation phase comprises the following steps, which are vital for project success. Functions to be assigned include contract awarder, contractor (project manager), management working group including planning/steering and controlling, and the steering committee.

Project initiation in the Court of Audit project model



The steering committee approves the individual phases of the project, reviews controlling reports and approves any necessary changes in project planning and resources.

The Development working group has the operative control of the project; functions such

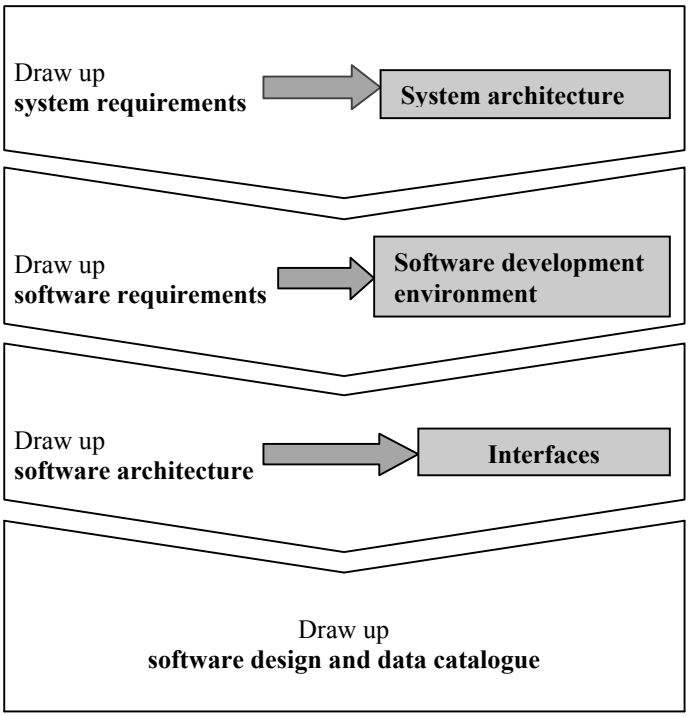
as project management, analysis and programming have to be assumed by team members with special IT skills who should, where possible, be fully assigned to the project. Project approval thus not only covers human resources, project time and budget, but in particular also the appointment of team members for the various project functions.

Application design

The application design phase shown in a simplified manner in the flowchart translates the project contents identified in the requirements analysis into a technical concept. Prior to software design as such, fundamental decisions regarding the technological environment for development have to be taken.

It is necessary to select a system architecture and a software development environment on the basis of the identified system requirements. Then the technical interfaces have to be defined, governed by the software architecture that has been chosen. Only after this technical architecture and the respective software tools have been defined is it possible for the contents of the requirements analysis to enter into the software design.

Application design in the Court of Audit project model

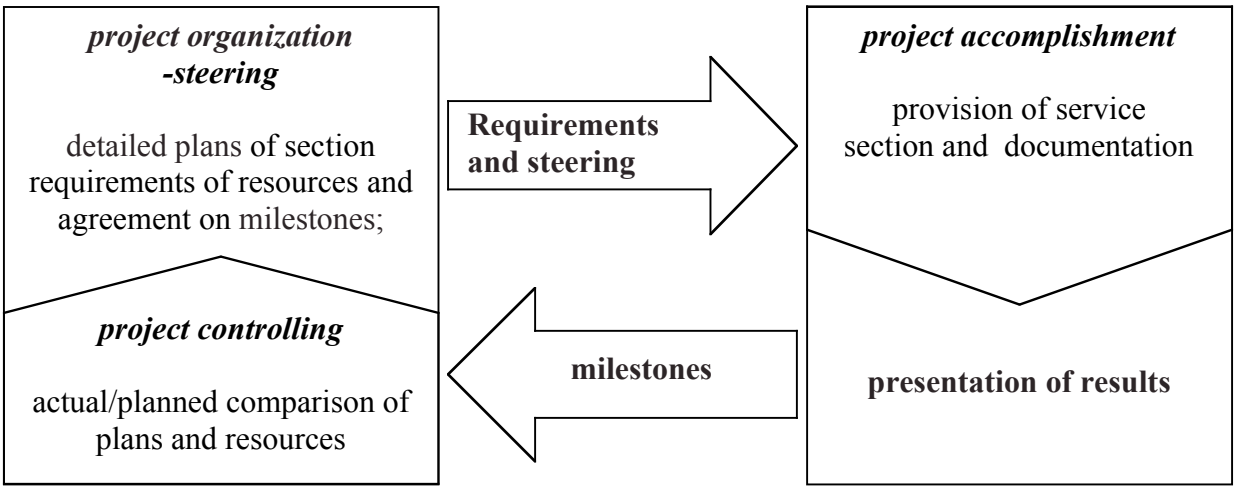


Controlling circuit

Basically, the Court of Audit considers it necessary to install comprehensive controlling. The plans have to be checked for plausibility and/or feasibility in advance in order to arrive at realistic specifications for subsequent evaluation. Controlling must be reflected in the project organization by periodical actual/planned comparisons (milestones) of output and input. If deviations from plan are found, the steering committee must be informed, and solutions must be proposed.

The detailed plans and work breakdown charts from the requirements analysis, application design and programming phases, plus the agreement on approved resources, serve as a yardstick for controlling. Output has to be continually documented, with results being summarized at regular intervals. The actual/planned comparison of output with plans (milestone) provides evaluations for project controlling purposes.

Controlling circuit in the Court of Audit project model



Court of Audit recommendations

The project management has to find an adequate project organization and adapt it as necessary. The project organization depends on the type and scope of the IT project, taking account of timeframe, number and skills of team members, as well as user requirements.

The Court of Audit recommends to additionally evaluate this project organization on the basis of the Court of Audit project model as follows:

(1) Are the tasks as described in the section on IT project phases, i.e.

- definition of objective
- initial identification of requirements and resources
- project functions
- approval
- requirements analysis
- application design
- programming
- test, acceptance, and integration

being mapped and being taken account of in terms of functions in the specific project organization?

(2) Does the logical sequence shown in the Court of Audit project model have benefits also for the specific project organization?

(3) Can the two-level approval procedure as shown in the Court of Audit project model be purposefully applied in the specific project?

(4) The definition of functions plays a very important role in the Court of Audit project model. This applies above all to contract awarder, project management, controlling, contractor, heads of analysis and programming and steering committee. Accordingly, the project approval process has to incorporate the appointment of persons for these functions.

(5) The Court of Audit 'application design' project model stipulates that the system architecture, software development environment, software architecture and corresponding interfaces be defined first, on the basis of the contents of the requirements analysis, before the software design and the data catalogue are implemented.

(6) The documentation of output and the presentation of results for each milestone make it possible to compare output and input against the respective detailed plans. Deviations found by controlling must trigger project steering measures. The project management has to report major deviations to the steering committee together and propose possible solutions plus a calculation of potential costs.

(7) The most frequent problems as listed by the Court of Audit in the section on IT project phases constitute a sort of 'black list' for controlling to avoid known errors in projects.