

# 10 things you should know about software development in SAP. Today ...

## 1. ...define development guidelines

Development guidelines are essential in order to create programs with a high degree of flexibility, maintainability and performance.

- Define development guidelines! What is allowed? What is not? Should there be naming conventions for individual programming modules? ...
- Train your programmers based on the defined development guidelines instead of just documenting the guidelines! That is the only way you can ensure that they will actually be understood and implemented.
- Make sure that the guidelines are observed and there is no difference between practice and theory!

It is not until you have defined the development guidelines that you will become fully aware of quality issues. Find out how the quality of programs should be defined in the next part.

## 2. ...define quality

Before a program is created, you need to be fully aware of where you see the quality of the program. You will certainly discover some conflicting goals.

- Consider what criteria are the most important to you! It is rarely possible to achieve all the quality aspects at the same time.
- Define measurable quality targets! For instance: a response time below a defined threshold, the number of maximum clicks until the result is shown etc. instead of making general statements such as "user friendly" or "quick".
- Clarify with your customer in advance when a solution is successful for him and when it is a failure!

Particularly with the last question, your customer will usually need support in order to define meaningful and achievable criteria. Read how you can clarify and combine the expectations of all the parties involved in the next part!

## 3. ...clarify expectations

Before any programming project, clarify exactly what is expected of the program to be created and put it down in writing! Define measurable targets and prioritise them. An improvement in one point often entails a decline in another point.

Good targets meet the following criteria:

- They describe an achievable state.
- They are challenging but can be achieved.
- Measurable criteria are used to describe them.

If it can be foreseen that the target cannot be achieved after all, it should be corrected as soon as possible. This could occur if, for instance, there are changes in customer requests or changes in the project team. Subsequent change requests require efficient change management. Read more on this in the fourth part.

## 4. ...change management

All the change requests are taken down and recorded in a structured manner in change management. They are then checked, budgeted and planned in terms of their effects by a team. Ensure that people from the following groups are members of this team:

- representative(s) on the business side,

- representative(s) of the developers,
- project manager,
- representative(s) from the top management.

This is the only way to ensure that a consensus can be reached on the assessment of change requests and their effect on the overall project because all of these groups have very different views of the project. Written documentation ensures that a change request is carefully considered in terms of the aim of the project and is assessed objectively.

You will find an important aspect for smooth communications between these different groups in the next part: understanding processes.

## 5. ...understanding processes

It is important that the individual developer also has an overview of the specialised process. Developers working on a problem in detail too often have in-depth understanding but are not aware of matters surrounding it. Make sure that all the developers are involved at an early stage of the project!

This will allow

- the team of developers to gain a basic understanding of the process in which the project is currently in,
- the specialist department's process of understanding to be supported by the views of developers.

In particular, the preparation of feasibility studies and prototypes will help to steer the discussion towards realisable concepts at an early stage.

The aim is a program design that meets requirements and ensures that less information is lost in communications.

If a project involves developments in a SAP ERP environment, developers should also know the fundamentals about the relevant modules. That is precisely the topic of the next part.

## 6. ...knowing ERP modules

Don't expect too much! No developer has to be able to do everything. But he should have basic knowledge of the ERP modules relevant for his programming. He needs greater in-depth expertise for typical standard expansion options such as BADIs and UserExits as well as standard interfaces such as BAPIs and IDOCs.

Additional important know-how for developers:

- A general overview of the process. It helps to rapidly become acquainted with individual topics if necessary.
- Knowledge where documentation and information can be found that supplements one's own know-how.

The experience gained from a whole host of development projects has shown that concentrating on logistics as one topic and finance as another sub-area is advisable.

Focusing on individual modules causes one to lose sight of the process as a whole.

The interplay between advisers or other project participants is important so that their know-how is shared. But how can knowledge be transferred from one person to another? The time factor should not be underestimated in this connection.

Read more about transfer times in the next part.

## 7. ...taking account of transfer times

With shorter projects and smaller teams each team member usually works consistently on his task as a specialist from beginning to end. Long-term projects (lasting at least one year) require a larger project team in order to make up for changes in staff (e.g. due to holiday, sickness, individual employees leaving the project team) that will always occur over longer periods of time.

- Make sure that all the members generally have a basic understanding of the various sub-tasks in their team! That is helpful for their own tasks and acts as a safety net for sudden absences.
- Allow for times for transferring know-how from the outset!
- If you know the date when an employee will be leaving, have a different team member take over a part of his or her tasks in good time. The free time gained from this for the staff member leaving is used to train the new employee.

Apart from changing the composition of the project team there are other risks that threaten the project's success. Active risk management is advisable in order to cope with these risks. Read more about this in the next part.

## 8. Active risk management

Consider the risks of your project! Analyse external and internal project risks! Plan possible countermeasures before the risk situation actually occurs! Communicate risks openly as well as the possible ways of managing them in the steering committee but without losing sight of the project's core task.

Typical external risks:

- Change in requirements.
- Planned resources are not available (budget, participation of external parties such as the department concerned, external programmers etc.).
- End result does not meet the customer's expectations.

Typical internal risks:

- Tasks are not completed within the time schedule.
- Developers do not have a sufficient extent of know-how.
- Communications do not run smoothly in the project team.
- Changes in team members.

In the next part, we will introduce code reviews as a measure to cope with risks.

## 9. ...Code reviews

By conducting code reviews on a regular basis, you ensure that the software created meets quality standards and technical requirements.

A code review is the analysis and critical questioning of the programs created and their design to determine whether they satisfy quality criteria.

To this end the respective developer checks the code he has developed.

As a next step, another review is conducted, this time in a group with an experienced developer because other developers see the program code through different eyes and possibly more objectively. It is important to be objective about the code (the code is reviewed, not the developer!). All the developers can learn from errors and constructive criticism by conducting a code review as a group.

Results of the code review:

- Code sections that are in need of improvement.
- Things that are left as they are but should be done better in future.
- Programming that can serve as a reference for other projects.

Supporting tools are e.g. SLIN tests and an analysis with the SAP Code Inspector. But they should only be seen as a supplement to a review by the developers.

In addition to code reviews, tests should be planned on a regular basis and at an early stage. We will consider these in the last part.

## 10. ...testing, testing, testing

During the term of a project it is advisable to carry out program tests at various points in time. In this context, we differentiate between function, integration and mass tests (particularly for testing performance).

### Function test

- Who carries out the test? - The developer of the function module.
- What is tested? - The functionality of individual modules is tested based on the technical specification. Make sure that any dependence on other modules is reduced as far as possible.
- When is the test carried out? - During the development process.

### Integration test

- Who? - Key users and the Informatics Department.
- What? - The interplay between the various modules based on defined process chains.
- When? - At the end of the development process.

### Mass test / Performance test

- Who? - All the developers
- What? - Especially critical sub-processes or functions are tested to see if they meet defined quality targets (performance test). With the mass test, the parallel use of the program by a typical number of users is also tested.
- When? - During the development process in order to identify performance bottlenecks at an early stage and to initiate corrective measures.
- In addition, at the end of the project in order to identify possible bottlenecks for operations later in the final version.