The Customization Process for Organizational Package Information Systems:

A Challenge for Participatory Design

Anita Krabbel, Ingrid Wetzel

Software Engineering Group

Department of Informatics

Hamburg University

Vogt-Koelln-Str. 30, D-22527 Hamburg, Germany

++49 40 5494 2301

{krabbel, wetzel}@informatik.uni-hamburg.de

ABSTRACT

Today, participation seems to be an established, necessary as well as valuable approach during system development. However, because of the shift from individually developed to package software participatory design needs to be addressed in new ways.

In this article we evaluate our consulting experience during the customization of an integrated hospital information system in the different units of an acute hospital. We address the important role of the users in the customization process and identify main problems and tasks which often arise. On this basis we provide means for overcoming the difficulties on three levels. We introduce the notion of system stages for structuring the customization process, we present committee structures to be implemented within the user organization for executing the process and we provide proven communication means for the close cooperation between the user and the vendor organization.

Keywords

Package systems, organizational information systems, customization process, participatory design.

1 INTRODUCTION

Package systems are quite successful. More and more they support organizational goals [4] [5] and provide integrated solutions for main business processes, cooperation facilities and controlling issues. Organizations buying such package systems expect better solutions than from in-house or contracted out development in the following directions:

- The offered solution is mature since it is used and proved good in several other organizations already, mistakes or insufficiencies have been expunged and improvements have been made. The own organization is not needing to go through all the starting problems which normally arise.

- The provided functionality is extensive and sufficient since various requirements are already captured.

- The software - even supporting complex processes affecting the whole organization - is quickly available since it is already developed.

- A long term development of the software is usually guaranteed, the vendor is trusted to be powerful enough to cope with the rapidly changing legal requirements today and in future.

- The system includes efficient process organization and communication devices.

- Costs are less because the development costs are distributed among various customers.

- As a consequence the organization can concentrate on its main business because it can buy the software together with appropriate services and know-how.

So the management of the user organization expects that after a short customization effort the system will run for the benefit of the organization. Usually and naturally, the vendors confirm this trust.

However, in practice the whole process of buying and making package systems a success for the organization is very different from these expectations.

Problems start with a difficult selection process, where the users are lost in an unclear and unstable market situation and unknown sales strategies as well as conditions. Additionally, user organizations often don't have much experience how to handle this task [7].

The following customization process seems to be even more complex. Out of our experience we become convinced that, from the viewpoint of the users, the effort for customization is just a little less than that for system development. And this in the face of a management often misjudging this situation expecting an easy system implementation. The consequences are at hand. Effort for participation is not much accepted, resources for participation are barely sufficient.

Therefore we want to address the customization process as being a challenge for participatory design. For doing so in this article, we first identify main problems with participation during customization (section 2). Then we list the kind of tasks arising during customization and show that nearly all of them have to be carried out by the users or user groups in a close cooperation with the vendor (section 3). In the following sections (section 4-6) we provide strategies and means for overcoming the mentioned problems. For participatory customization we propose a way to structure the high amount of tasks and make it transparent to the user organization. We present useful new organizational structures which need to be introduced in the user organization and we give examples for means to establish a fruitful and quarrel-free cooperation between the vendor and the user organization.

The examples throughout the paper are chosen from our hospital project. Briefly, this project started two and a half years ago. We as a team of three computer scientists (one of us being also a nurse) made a requirements analysis on the basis of workplace studies and participatory techniques for understanding the cross-departmental processes (s. [7]) in the hospital. Out of this and in close cooperation with representatives of the hospital we worked out criteria for the future system and carried out a market analysis for hospital information systems in Germany. We proposed a system and after a decision process in the hospital the system was bought and is now in its customization process. Currently, the system runs in the patient administration and starts to get introduced at the wards for the physicians and nurses.

2 PROBLEMS DURING CUSTOMIZATION

During customization usually lots of problems arise [9]. They have quite different roots, the main ones are: wrong expectations, different interests, missing understanding of the tasks and ways to proceed, decisions concerning organizational changes, cost factors, missing arrangements for cooperation between the user and the vendor organization, unclear system support for customization. We order the problems around the main actors and factors - the management, competing and heterogeneous user groups, the vendor, the partnership between user and vender organization and the system itself.

Management

The importance of the management is manifold as usual since the management decides over the costs, resources and possibilities for participation within the project. Its relationship to the project and interest is crucial.

- **Involvement**: The management might not be related to a project with package software as to a project being an in-house or contracted out software [5]. In case of package systems, always the vendor/system can be blamed (alone). Additionally, organizational development or cooperation support - as being main intentions of organizational information systems - is not always at the core of interest. This might come true especially in the area of hospital information systems where the investments in the clinical sections seem to be of higher importance.

- **Expectation**: The expectations are according to the one mentioned in the introduction. A system which is already elsewhere used is expected to have a smooth introduction without big problems requiring not many resources. Furthermore, the system is expected to improve the organization and direct the necessary changes. It will give the right answers and free the organization from decision making.

Competing heterogeneous user groups

As pointed out in [6] participatory design in the context of large integrated information systems supporting cross-departmental processes has to address not only „the user" but different heterogeneous user groups. These groups have quite different, often competing expectations and needs as well as status and influence. The customization process needs users from these different units doing the actual adaptation work.

- **Expectation:** Each unit expects from an integrated package system that it will provide a sophisticated and optimal solution for the own unit. But vendors offering integrated systems for different units not seldom have their specialties in one or two units and others are just „done" and therefore not excellent. So the chosen system might disappoint some user groups in not being sophisticated enough.

- **Unclear customization tasks**: Often it is unclear for the user organization and those users involved in the process what tasks the customization includes. As a consequence it is nearly impossible to plan how much time the tasks will take. And this on the background that the customization effort is often an addition to the normal work load.

- **Competition for resources**: The different units need to work in parallel on different customization tasks.

Internal as well as external resources need to be subdivided. Resources include supporting time from the computer department, money for investments, vendor time for adaptation and extensions etc. A missing coordination and negotiation can lead to severe ill-feeling of certain user groups. This ill-feeling can burst out possibly very lately.

- **Organizational development**: The use of an organizational information system will affect the cooperation between employees and units. The system will provide flexible, innovative or just different cooperation means or work division. According decisions which have to be made during customization are difficult because the acceptance and the success of new solutions are hard to anticipate [8]. Additionally, the decisions have to be made by a group of representatives from different departments and have manifold aspects, s. e.g. [1]. A consequence is that sometimes the adaptation requirements will change back and forth because of uncertainty and negotiation. If there is no documentation the argumentation might even go in cycles. If interim solutions are given out to the vendor, as often happens, changes in requirements will cause irritations on the cooperation with the vendor. Sometimes important decisions require even management directives which might take much time and paralyze the work.

The vendor

The role of the vendor - its own organization, its knowledge of ways to proceed, its skills in shaping a cooperation with the customer organization - is quite crucial for the customization process. But the vendor has interests in his own right.

- **Unclear politics**: As usual, the vendor is eager for customers. In the context of selling package systems to organizations this has special impacts. Often the cost factor for the customization process is covered up. This is due to fact that the vendor captures customers with cheap software prices and makes his money with the services like consulting, customization and training. If the management of the user organization isn't willing or able to spend the money for these services this has severe consequences for the customization process and all of its participants.

- **Lack of ways to proceed**: It seems not seldom that vendors of package systems lack in having a clear understanding of structured ways to proceed during customization. Since they and not the user organization are expected to direct the process a lack in this knowledge has also severe consequences for the process.

- **Handling system versions**: Not seldom problems arise when new package system versions clash with already worked out customization in the old version or special extensions for one organization.

The partnership between the user and vendor organization

Customization and system introduction of an organizational information system often last over years. The cooperation is close and needs good dealings and settled rules. The more a system is introduced into the organization the more the organization is dependent on the vendor/system. On the other hand the vendor is interested in organizations being good reference customers for the system. Problems spoiling this sensible and important partnership might undermine the success of the whole project.

- **Cost factor and responsibilities**: Problems arise if there are unclear rules about costs and responsibilities during customization. The questions which need to have established ways of negotiation are: Which tasks have to be accomplished and paid by whom? What kind of customization will raise additional costs? Who will pay for delays which might arise either way (e.g. user to vendor: some customization information is missing in time, vendor to user: required system changes are delayed).

- **System support of a standard system**. A difficulty arises if users are not satisfied by the adaptable system's functionality and usage. This can have different reasons, the system might simply have errors, functionality is missing, the system is not as flexible as expected or the users want to have additional or different support.

In case of missing functionality the view points can differ extremely. The vendor providing the additional functionality wants to get paid for it. The user organization assesses the missing functionality being standard and, by providing a requirements document, having supported the vendor in building this system portion and might want to get paid for this.

Another problem lies in the fact that users often have the opinion that the bought system should support the tasks in a standardized way, meaning by 'standardized' the way the own organization is doing it.

- **System extensions**: In the presence of package systems user requirements for changes and extensions might not be taken up. Within the vendor organization a product manager has to gather requirements from different customers and make decisions. Maybe the required changes from one customer won't win at all.

The Package System

Neither for package systems nor for organizational information systems with cooperation support do there exist established ways to develop, to design or to construct the systems [8]. Therefore with the pressure on the market optimal solutions should not be expected.

- **Interfaces**: Problems arise if the interfaces for customization are difficult to use or even missing. The design of customization interfaces varies. The easiness for users to do parts of the customization on their own highly depends on the quality of these interfaces.

- **Degree of flexibility**: Systems can differ in their degree of flexibility to be adapted to the needs of the users. If adaptability is missing it means that the vendor has to change the system code for this customer. This has consequences on the handling of future system versions from the vendor.

- **Software architecture**: For certain kinds of customization the software architecture seems to be easy e.g. reading tables from a database. Even in this context questions arise if tables should be changeable while the system is running. Flexibility in providing different cooperation means for articulation work (e.g. providing different ways to handle allocations of patient examinations) [12] are much more difficult to support and need additional research. Integration issues with triggering events and data across system borders using a communication server are not at all standard. The distribution of new package software versions in the context of specialized extensions at different customer sites requires new approaches in versioning.

3 CUSTOMIZATION TASKS

After facing so many problems it becomes clear that the customization process is highly complex and can be extremely tedious for users. We feel that a contribution to participatory design during customization should start by categorizing and listing the usual customization tasks, s. also [2][10]. This makes the effort transparent and can help clearing up management expectations being willed to follow blurring from the vendors side (hardly spoken). It can prevent the future system users from suffering the lack of necessary resources for customization. And it shows how many of these tasks have to be performed by the future system users.

We build the following categories:

- **Provision of organizational knowledge in catalogs.**

Normally a lot of information which is organization or region dependent needs to be given in catalogs. It can belong to

- the structure of the organization,

- the specialty of its services,

- organization dependent descriptions of legally required catalogs,

- information about cooperation partners of the organization, etc.

Examples are the list of different wards, X-ray investigations, standard patient processes for certain diagnoses, descriptions for diagnosis keys, catalogs of health insurance organizations or family doctors, etc. For accomplishing these tasks it needs to be determined and distributed:

- Which catalogs have to be filled out?

- Which interfaces are provided for these tasks, does a documentation about their usage and availability exit, or do users need to be trained?

- Who has to give the information (The design of some catalogs e.g. investigations or standard patient processes need a lot of specialized knowledge determined in careful and work-intensive cooperation)?

- Do useful models from the vendor exist?

- **Setting-up work places**

  - For each work place kind (with the same tasks) a detailed check of the system functionality has to be made. This includes the available system functions, the information which will be produced or used and the printouts to be made. Examples are: Can the choices a patient wants to make for his/her stay be calculated in the system (single or twin room, choice of physicians), is it possible that a patient has more than one family physician, are certain statistics available, etc?

  - The user interface needs to be tested. Should the system provide pre-filled standard values in data fields, should a navigation through the fields be imposed, do possibilities to change window attributes exist, etc?

  - If there exist temporal orders among functions should this be enforced by the system?

  - For each of the workplaces the access rights (who can start which module or functions, who is allowed to see which data, etc.) needs to be determined, documented and accordingly installed.

For each of these checks which need to be performed together with the future user at the corresponding

workplace a change protocol needs to be written. It is not easy to decide at which times these checks should be made, after customization with catalogs or before. Before has the advantage of having time for changes, after has the advantage that all of the functionality can be tested.

All this work is a big effort. Not doing it means to have hard times after the system introduction.

- **Arranging cooperation facilities**

  - In the context of cross-workplace or cross-departmental workflows the kind of listing of open work tasks for each different workplace kind has to be determined.

  - Additionally the information flow has to be worked out. Which information is sent to whom and who is informed about changes, e.g. who has to sign when orders for a certain investigation?

  - Another check concentrates on the cooperation in cross-departmental processes. In case of mismatch between the existing handling and the provided system support one needs to decide if the organization should adapt to the system solution or the system should get changed.

- **Testing the adaptation and extensions**

  - After customization with catalogs and setting up of the workplaces the future users again have to test its success, i.e. customization as system development does need to be done in cycles [3] [13].

- **Training**

  - The training of users usually involving many employees has to be planned carefully. It should take place after the customization is finished. However, because of the sometimes unexpectedly arising cycles during customization, changes of time schedules for training might happen and disturb in many the trust into the project's success.

- **Planning the process and cooperation with the vendor**

  - During these customization tasks and system checks users will find errors or missing functionality and will define change requirements.

  - For both the planning of the process and the cooperation with the vendor (mainly in the sense of negotiation of prices) it is of utter importance to build categories of arising tasks the vendor has to carry out. E.g. each required extension needs time, often the usual development cycles and can delay the whole introduction of the system. Additionally, users seldom are able to make requirements specification on their own, so it must be planned who will assist them in doing so.

  - We therefore distinguish three categories: errors, usual customization, extensions or changes, s. also [11]. Errors should be eradicated by the vendor without additional costs, for bigger changes and extensions new contracts need to be set up.

Summing up and looking at the manifold tasks we want to repeat intentionally that - from the users side - the customization effort is nearly as task-intensive as a development of a system is. It often takes years. And it seems that the user organization has even more responsibility. Lots of the tasks are tasks the users have to perform - providing information and performing the customization.

For overcoming the complexity of these tasks and processes we propose that a structuring on different levels is necessary: the task load, the users performing the tasks and the vender-user cooperation. This will be introduced in the next three sections.

- Task load: It is necessary to structure the whole process with the manifold tasks in the large. What has to be done when in which sequence. We therefore introduce the notion of system stages being based on a kernel system.

- The responsibilities for the customization work within the user organization need to be settled. We introduce a project leader group and project groups describing their tasks and cooperation.

- The close cooperation between the user and the vendor organization needs to be structured. We present a periodically written progress report document for this reason.

4 SYSTEM STAGES FOR STRUCTURING THE CUSTOMIZATION PROCESS

Organizational package information systems for large organizations like hospitals should provide support for many different departments of an organization. Normally, the required system support is not provided by one vendor alone. Instead integration of several systems is required.

During the selection process we therefore propose to subdivide the entire system into a system kernel and separate it from specialized subsystems not belonging to the kernel.

As pointed out in [8] the kernel should support tasks of key units or departments which show a high cooperation profile.

It has to satisfy urgent needs of the organization or tasks which are often performed. Additionally, it must support cooperation by providing main shared data and should supply a basic and uniform set of cooperation means. The application kernel has to be designed in a way which supports the integration of specialized systems. Out of these domain-oriented requirements, several technological requirements concerning the architecture and the openness of the system follow.

The package system should be selected according to the domain-oriented and technological requirements of the kernel. This needs a careful selection since for many application domains there still doesn't exist an agreement about the content and the architecture of a kernel system.
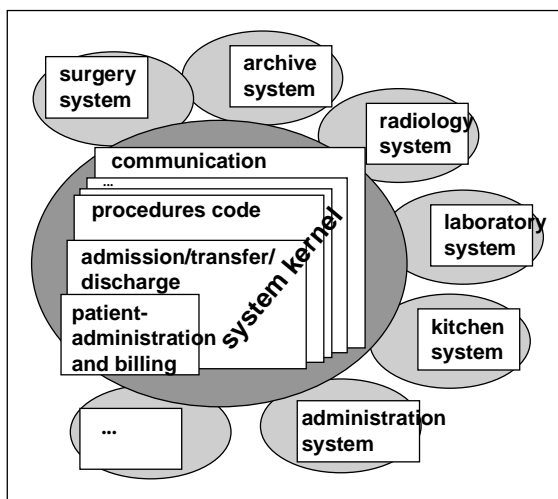

Figure1: System kernel with subsystems for a hospital of small size

Nevertheless, for the customization process it is now important to further subdivide the still large kernel system into stages since it can't be introduced in one step. We must therefore plan and proceed in system stages. System stages determine the logical dependencies of the functionality of the application kernel. They describe within each stage the system support for tasks which are quite independent.

The resulting structure supports a careful planning of each of the stages concerning different aspects, e.g., additional hardware requirements, implications in the work task organization, determination which workplaces are affected, changes in the environment, etc.).

System stages also allow to plan and design necessary interim solutions on different levels of detail. Their careful design can be very important in the acceptance of the total system. An example for an interim solution is the cross-departmental workflow between the patient administration and the physicians at different wards for a patient's admission. During the first three days of a patient's stay physicians have to deliver a diagnosis key to the patient

administration which transfers this key to the health insurance companies for getting a cost guarantee for the patient. Since the system starts to be used in the patient administration without being introduced at the same time at the physicians' workplaces the administration needs interim system support for typing in the diagnosis keys themselves still receiving them by paper. This interim support has to be planned, trained and integrated into the system and disappears as soon as all of the physicians work with the system.

Additionally, system stages even ease the planning of integrating specialized systems relating their introduction to certain stages.
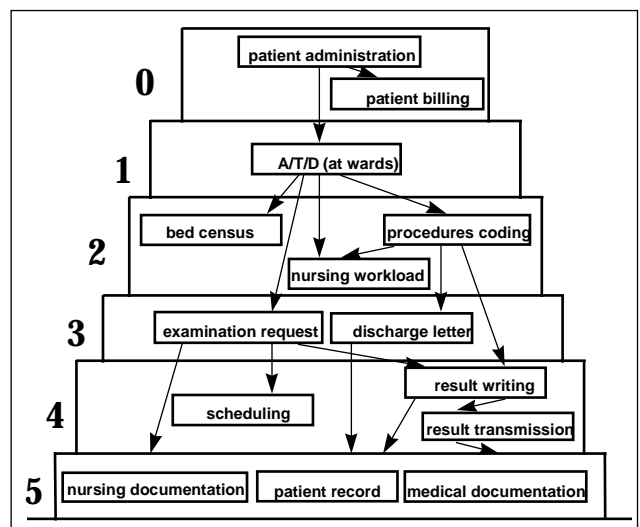
An example is given in figure 2.

Figure 2: System stages for the kernel system

From our project experience we only can emphasize the usefulness of system stages. We introduced concrete (named) stages during a workshop at the end of the market analysis and decision phase for the system. During this workshop they made necessary sequences for system introduction transparent to the different user groups and the management. We were astonished seeing one day after the workshop an information sheet covering the system stages being distributed across the whole hospital. Since then the concrete system stages form a part of our common project language.

5 IMPLEMENTATION OF COMMITTEES IN THE USER ORGANIZATION

Addressing the complexity of the many customization tasks, we above introduced means for structuring these tasks in steps or system stages. Another source of complexity lies in the fact who has to work on these tasks and be responsible for them. In our project, we had to implement new organizational groups within the hospital. A project leading group was founded running over the whole time of the

customization process and several project groups were introduced for certain tasks during customization. In the following we describe the tasks of the committees, their meeting frequency, budget, to whom they have to report and which members they should incorporate.

The project leading group (PLG)

The PLG sits periodically, depending on the work load between every two or five weeks.

- *Tasks:* The PLG has to prepare, to direct and to control the customization process and the use of the running system. It gives priorities during system introduction and sets up project groups, determines their tasks in a broad manner, coordinates their work and supervises their results. The PLG is responsible for cross-departmental organizational decisions arising with the new system and evaluates the project.

The PLG is responsible for agreements and correspondence with the vendor in case of problems and difficulties (see also section 2).

- *Budget:* The PLG has its own budget and is responsible for spending this money.

- *Report:* The PLG reports to the hospital management about the project progress, points out decisions to be made and presents recommendations.

- *Members*: Employees of different units of the hospital are members of the PLG, ideally it comprises one representative of each of the (running) project groups and at least the head of the computer department.

The project groups (PGs)

Projects groups are introduced for each unit/profession using a system module of the system kernel, are devoted to the choice and integration of a specialized system or belong to cross-departmental tasks normally being related to one system stage. Examples are project groups for patient administration, physicians, nurses etc. Groups for specialized systems are laboratory, radiology, kitchen etc. Overlapping groups are concerned about patient labels, controlling statistics, data protection, implementing a physical network in the hospital, etc.

Normally, project groups have a long life because of the system stages (system introduction at a certain unit often lasts over different stages) but are being flexibly active according to the actual needs.

- *Tasks*: PGs have to perform the actual detailed tasks as listed above in section 3 for the portion of the system being used at the corresponding workplaces they are responsible for.

- *Budget*: PGs give cost requirements to the PLG.

- *Report*: PGs have to provide detailed plans for activity to the PLG and report about results in a certain frequency. The PGs gather and provide an atmospheric picture about the customization, staff training and the running system.

- *Members*: PGs should have a stable but small set of members. They involve other staff as necessary.

In case of having no computer department the PLG needs to fight for resources and implement such a group (as happened in our project). The computer department also has to be organizationally settled in the organization and works in close cooperation with the PLG and PGs.

6 PROGRESS REPORTS AS A COMMUNICATION MEANS BETWEEN THE USER AND THE VENDOR ORGANIZATION

For a documentation of the joint activities and close cooperative efforts between the vendor and the user organization a progress report has to be written. It should be in a useful periodicity, we propose to provide it monthly.

It should include at least three different parts:

- an easy to grasp *graphical overview* about the main activities in given categories,

- a more detailed description of the activities in each category with *planned and actual work* to evaluate the deviations and report on reasons for them,

- a brief description of each *appointment with the vendor.*

The progress report should include pointers to other sources of documentation if available, as to software documentation, version numbers, protocols, installation guides etc.

Graphical overview

The graphical overview provides an easily comprehensible document for the progress report over months and years. It can be used by the PLG or the management of the user organization but also in case of dissent and problems concerning the whole project progress.

It shows major and minor activities (indicated by the thickness of the bars) and their duration. The activities are very broadly summed up in categories which need to be determined from representatives from both the vendor and the user organization. These categories might change over time according to actual work performance. In the start of a project they include activities on buying hardware and installing a network throughout the organization. Integration issues normally arise for long periods because they seem to be a never ending story or arise anew with each specialized

system which need to be chosen and integrated. A big topic is at each time the software, different versions, error removal, software extensions and customization. This can be further split according to the work items of the different PGs.

Additionally, the number and the date of the appointments with the vendor are documented, noted as circles on the time scale.

Figure 3 gives an example for this kind of document, having one axis for the categories and one for the time scale.
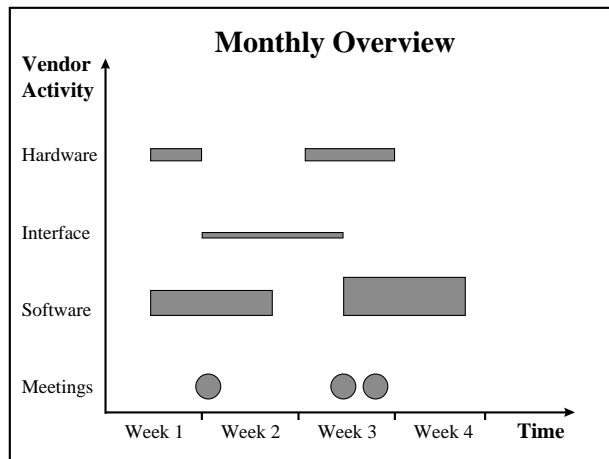
Monthly Overview

Figure 3: Example for a monthly graphical overview

Planned and actual work

The more detailed description of the planned and actually performed tasks is provided for each of the categories from the graphical overview and shows the more detailed items of work.

The overall intention is to document the planning and agreed cooperation as well as the status of the actually performed work. It captures reasons for deviations between planned and performed actions and is used to evaluate the reasons for better future planning.

In detail it captures for each task its subject, a short description, the subject of tasks, responsibilities, person months, conditions, negotiated dates for exchange of prerequisites (like documents, requirements) from one partner to the other and reasons for delay.

Some examples are for exchanged documents might be:

- purchase documents for buying a server,
- plans for the network,
- configuration concepts for clients, installation guides, access rights, documentation,
- documentation of software versions because of removal of errors or providing extensions,

- requirements documents for extensions,
- strategic documents for integration of specialized systems.

Figure 4 presents a table for documentation.

Task		
	Planned	Actual
Task description		
Estimate (person months)		
Input from user/vendor what when who		
Output from user/vendor what when who		
Reasons for delay	-	
Overhang to the next month		

Figure 4: Table for documenting planned and actual tasks

Appointments with the Vendor

Normally because of budget reasons, a documentation of the appointments with the vendor are necessary. They include

- the agreed goal for the day and what actually was achieved,
- a reference to protocols or arrangements.

During the project it arose that it is quite helpful to provide such frames for documentation. They ease the documentation work and unify the whole resulting documentation.

7 SUMMARY

In this article we address the customization process as being a challenge for participatory design. We claim that in the context of organizational package information systems the customization work in the user organization is comparable with the necessary participation during system development projects. Even in some aspects the user has to cope with work he didn't have to accomplish in development projects such as structuring the process, providing requirements specification (for extensions or changes), filling catalogs with organizational knowledge.

The tasks are manifold and often overseen to be that work

intensive. In order to underline the complexity and the importance of the subject we listed often arising problems during customization. Sources are wrong expectations from the user organization concerning package software, missing involvement of the management in customization projects, competing heterogeneous user groups with different expectations, competition of resources, unclear customization tasks and organizational changes which need to be decided and handled. Further problems take roots in unclear politics on the vendors' side and a lack of established ways to proceed. Dissent arises conc. cost factors, responsibilities, what functionality a package system should cope and inflexibility conc. system extension because they need to fit into the product management and how the system is used in several organizations. Furthermore, the quality of a package system may vary in the provided interfaces for customization and degree of adaptability.

As a contribution towards overcoming these problems we listed and categorized the different tasks arising during customization. They include the provisions of organizational knowledge in system catalogues, setting up workplaces through testing, determining adjustments of system modules, writing change or extension requirements documents, arranging cooperation facilities, testing adaptations and extensions, planning the training of the staff and last but not least planning and structuring the process and the cooperation with the vendor.

For carrying out projects and coping with their intrinsic complexity we proposed three structures found helpful in our own project experience. We introduced a temporal structure in accordance to system stages. They make the process transparent to the user organization and manageable through ordering the manifold tasks. We pointed out the importance of committees which need to be settled in the user organization. They are responsible for carrying out the strategic planning and leading the cooperation among the user groups and with the vendor as well as performing the detailed customization work. Since the partnership with the vendor is of high importance during the whole project we presented a progress report document for planning and documenting the cooperation.

REFERENCES

1. Egger, E., Wagner, I.: Negotiating Temporal Orders. The Case of Collaborative Time Mangement in a Surgery Clinic, In: *Computer Supported Cooperative Work*, vol 1, 1993 pp 225-275.

2. Fischer, G., Girgensohn, A.: End-User Modifiability in Design Environments. In: *Proceedings of CHI '90* (Seattle 1990), ACM Press.

3. Floyd, C. Outline of a paradigm change in software engineering. In Bjerknes, G., Ehn, P., Kyng, M. (eds.) Computers and democracy – a Scandinavian challenge. Aldershot, England, Avebury 1987, 191–210.

4. Grudin, J.: Computer-Supported Cooperative Work: History and Focus, In: *IEEE Computer*, Vol. 27, No. 5, 1994, pp. 19-26.

5. Grudin, J.: Groupware and Social Dynamics: Eight Challenges for developers. In: *Communictations of the ACM*, Vol. 37, No.1, 1994, pp 92-105.

6. Krabbel, I. Wetzel, I., Ratuski, S.: Participation of Heterogeneous User Groups: Providing an Integrated Hospital Information System. In: J. Blomberg, F. Kensing, E. Dykstra-Erickson (Eds.): *PDC'96* Proceedings of the Participatory Design Conference, Cambridge, Massachusetts, November 1996, pp. 241-249

7. Krabbel, A., Ratuski, S. Wetzel, I.: Requirements Analysis of Joint Tasks in Hospitals. In: B. Dahlbom et al. (eds.): IRIS 19 „The Future", *Proceedings of the 19th Information systems Research seminar* In Scandinavia, August 1996 at Lökeberg, Sweden. Gothenburg Studies in Informatics, Report 8, June 1996, pp. 733-749

8. Krabbel, A. Wetzel, I. Züllighoven, H.: On the Inevitable Intertwining of Analysis and Design: Developing Systems for Complex Cooperations. In: G. van der Veer, A. Henderson, S. Coles (eds.): *DIS'97* Designing Interactive Systems: Processes, Practices, Methods, and Techniques, Conference Proceedings, Amsterdam, The Netherlands, August 1997, pp.205-213

9. Mackay, W.E. Triggers and Barriers to Customizing Software. In: *Proceedings of CHI '90* (Seattle 1990), ACM Press, 153-160.

10. MacLean, A. et al.: User-Tailorable Systems: Pressing the Issues with Buttons. In: *Proceedings of CHI '91* (New Orleans 1991), ACM Press, 175-182.

11. Mørch, A.: Three Levels of End-user Tailoring: Customization, Integration, and Extension. In: *Proceedings of the 3rd Decennial Conference: Computer in Context: Joining Forces in Design* (Aarhus 1995).

12. Schmidt, K., Bennon, L.: Taking CSCW Seriouly: Supporting articulation work. In: *Computer Supported Cooperative Work*, vol.1., nos 1-2., 1992, pp 7-40.

13. Stiemerling, O. et al.: How to Make Software Softer - Designing Tailorable Applications. In: *Proceedings of DIS '97* (Amsterdam 1997), 365-376.