

A HOLISTIC APPROACH FOR MANAGING REQUIREMENTS OF DATA WAREHOUSE SYSTEMS

Josef Schiefer

IBM Watson Research Center
josef.schiefer@us.ibm.com

Beate List

Vienna University of Technology
list@ifs.tuwien.ac.at

Robert M. Bruckner

Vienna University of Technology
bruckner@ifs.tuwien.ac.at

Abstract

Quantitative evaluations show that many data warehouse projects fail at an alarming rate, wasting all the time, money, and effort spent on them. Cost overruns and schedule delays are attributed in large part, to problems associated with requirements. Therefore, a crucial success factor for the implementation of data warehouse systems is efficient requirements management. In this paper we introduce the easyREMOTE^{DWH} (easy Requirements Modeling Technique for Data WareHouses) requirements management approach, which facilitates an incremental and iterative development of complete, correct, feasible, necessary, prioritized, unambiguous, and verifiable data warehouse requirements. As a foundation we present a requirements specification template with consideration of different stakeholder perspectives and the typical characteristics of data warehouse systems. We discuss traditional requirements management activities (change control, version control, status tracking, requirements tracing) in the context of data warehouse environments and extend these traditional activities to support more efficient requirements documentation and visibility.

Keywords: Data warehousing, requirements engineering, requirements management

Introduction

Data warehouses are high-maintenance systems, because reorganizations, product introductions, new pricing schemes, new customers, changes in production systems, and so forth are going to affect the data warehouse. If the data warehouse is going to stay current (and being current is absolutely indispensable for user acceptance), changes to the data warehouse have to be made promptly. Therefore, the data warehouse system has to evolve with the business trends.

The development team of a data warehouse system often struggles with this evolutionary behavior, because requirements are always changing. For a better control of this problem, an organization has to define the steps that the data warehouse team will perform to manage their requirements. Documenting these steps enables the members of the organization to perform essential project activities consistently and effectively. Therefore, the following topics should be addressed:

- A model of documenting and capturing data warehouse requirements
- Tools, techniques, and conventions for controlling versions of requirements,
- Procedures, processes and standards, which are used to manage requirements changes
- The steps to follow for analyzing the impact of proposed changes to the data warehouse project
- The ways that new requirements and changes to existing ones are proposed, processed, negotiated, and communicated to all affected functional areas
- Which requirements statuses will be used, as well as the permissions to change them
- Requirements status tracking and reporting procedures

This paper discusses requirements management activities that address these topics. The remainder of this paper is organized as follows. In section 2, we discuss the contribution of this paper and related work. In section 3, we give an overview of data warehouse requirements and introduce a model for the requirements capture. In section 4, we give an overview of the requirements management activities for data warehouse systems. In the following sections 5 - 9 we discuss the requirements management activities in detail. Finally, in section 10 we provide our conclusion.

Contribution and Related Work

Building a data warehouse system is different from developing transaction systems, whereby the requirements analysis process for the latter is supported by numerous methods. Many authors, like Bickerton (1992), Davis and Allen (1993), or Sommerville and Kotonya (1998), have written publications that survey requirements management methods and requirements tools. But little has been done to compile the best practices of requirements specification independent of tools or methods and in the context of the business needs that requirements must fill. Up to now, the data warehouse design process has not been supported by a formal requirements analysis method, although, there are some approaches for requirements gathering.

Inmon (1996) argues that a data warehouse environment is data driven, in comparison to classical systems, which are requirement driven and the requirements are understood after it is populated with data and being used by the decision support analyst. He derives the data model by transferring the corporate data model into a data warehouse schema and by adding performance factors. List and Schiefer (2000) introduced an iterative and incremental requirements development process for data warehouse systems, but they do not include an approach for managing the requirements.

A process driven approach is applied by Kimball et al. (1998), whereby the fundamental step of the design process is based on choosing a business process to model. As this approach has proven its success in various projects, and as enterprises in general have shifted to process-centered organizing, we adopted the process-oriented approach.

Suzanne and James Robertson (1998) introduced the Volere requirements management approach. We are using the Volere requirements process as a basis for the easyREMOTE^{DWH} requirements process and extend it with additional process activities, which are necessary for continually evolving, large-sized projects like data warehouse projects.

Our contribution is the introduction of a proper requirements specification model for data warehouse systems. We are using abstraction levels for the data warehouse requirements to represent the different perspectives of stakeholders. Furthermore, we discuss requirement management activities and give guidance in providing tool support for these activities.

Data Warehouse Requirements

Figure 1 shows the impacts of data warehouse requirements. It demonstrates that data warehouse requirements directly affect technical and design aspects of a data warehouse system as well as project management issues.

The data warehouse requirements are the foundation for all future project activities and have a major impact on the success of the data warehouse project. Studies have shown that 40% of all data warehouse projects never develop, and 85% fail to meet business objectives (Wong et al. 1999). On average, data warehouses usually fail as a result of poor communication between IT and business professionals, as well as developers who possess poor project management skills and procedures. In order to achieve a successful data warehousing implementation, a great deal of requirements engineering effort and planning is required.

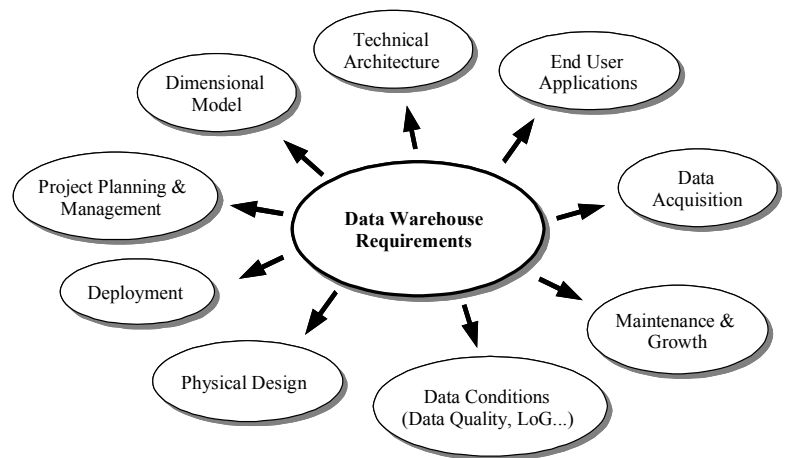


Figure 1. Impact of Data Warehouse Requirements (adapted from Kimball, 1998)

Requirements of an enterprise-wide data warehouse system determine its functional behavior and its available information, for example what data must be accessible, how it is transformed and organized, as well as how it is aggregated or calculated. The requirements enable the stakeholders to communicate the purpose, establish the direction and set the expectations of information goals for the enterprise.

Stakeholders often express their needs in general expectations of the data warehouse system to improve their business. This business view describes the goals and expectation of stakeholders, which is the foundation of the data warehouse requirements. On the other hand, the development team of a data warehouse system expects a complete, correct and unambiguous specification of the system it has to build, which means a further refinement of the business requirements from the stakeholders. Therefore, it is necessary to transform the business requirements to a detailed, testable, and complete specification for the data warehouse team (Wiegiers 1999).

For that reason, data warehouse requirements have different abstraction levels. Each abstraction level has its own group of stakeholders and shows the data warehouse system from a different perspective. Figure 2 shows the different perspectives used in the easyREMOTE^{DWH} approach.

Requirements from business perspective represent high-level objectives of the organization for the data warehouse system. They are primarily captured in a document describing the project’s vision and scope. The business requirements identify the primary benefits that the data warehouse system will provide to the organization and its

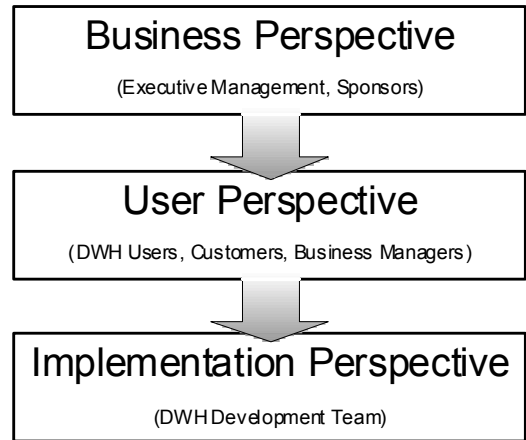


Figure 2. Perspectives for Data Warehouse Requirements

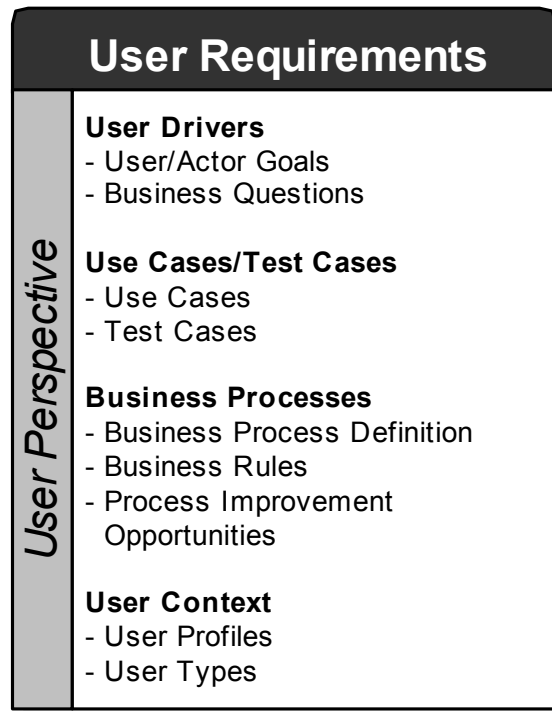


Figure 4. Requirements Template for User Perspective

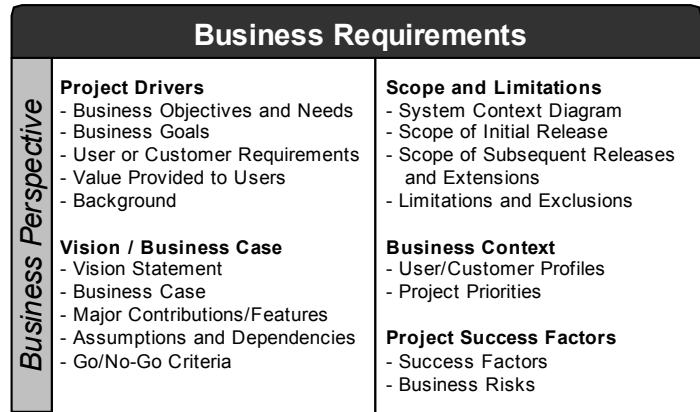


Figure 3. Requirements Template for Business Perspective

users. They represent the top level of abstraction in the requirements chain. They express business opportunities, business objectives and describe the typical users and organizations requirements and their provided value of the system at a high level. Figure 3 shows a detailed template for capturing business requirements.

Requirements from user perspective describe the tasks that the users must be able to accomplish with the help of the data warehouse system. These requirements must be collected from people who will actually use and work with the data warehouse system. The user requirements must align with the context and objectives established by the business requirements. They can be captured in use cases or scenario descriptions; they focus on what the users need to do with the data warehouse system and are therefore much more powerful than the traditional requirements elicitation approach of asking users what they want the system to do. Figure 4 shows a template for user requirements.

Requirements from an implementation perspective represent the data warehouse requirements on a very detailed level. The high level of detail facilitates the complete, fine-grained specification of the requirements, which are an important input for the data warehouse development team. They must align with the user and business requirements. Functional requirements define the functionality that the development team must build into the data warehouse system to enable users to accomplish their tasks, thereby satisfying the business requirements. Information requirements define the information needs of the organization. They describe the information and data, which the data warehouse should deliver or should have access to.

Furthermore, requirement attributes and interface requirements augment the description of the functional and information requirements by describing the characteristics in various dimensions that are important either to users or to the data warehouse team. They are properties, or qualities that the data warehouse system should have. For instance, when the functional requirements are known, it can be determined how they are to behave, what qualities they are to have, and how big or fast they should be. When the information requirements are known, its attributes can be determined, like data quality or data granularity. Figure 5 shows a template for specifying functional and information requirements including their attributes.

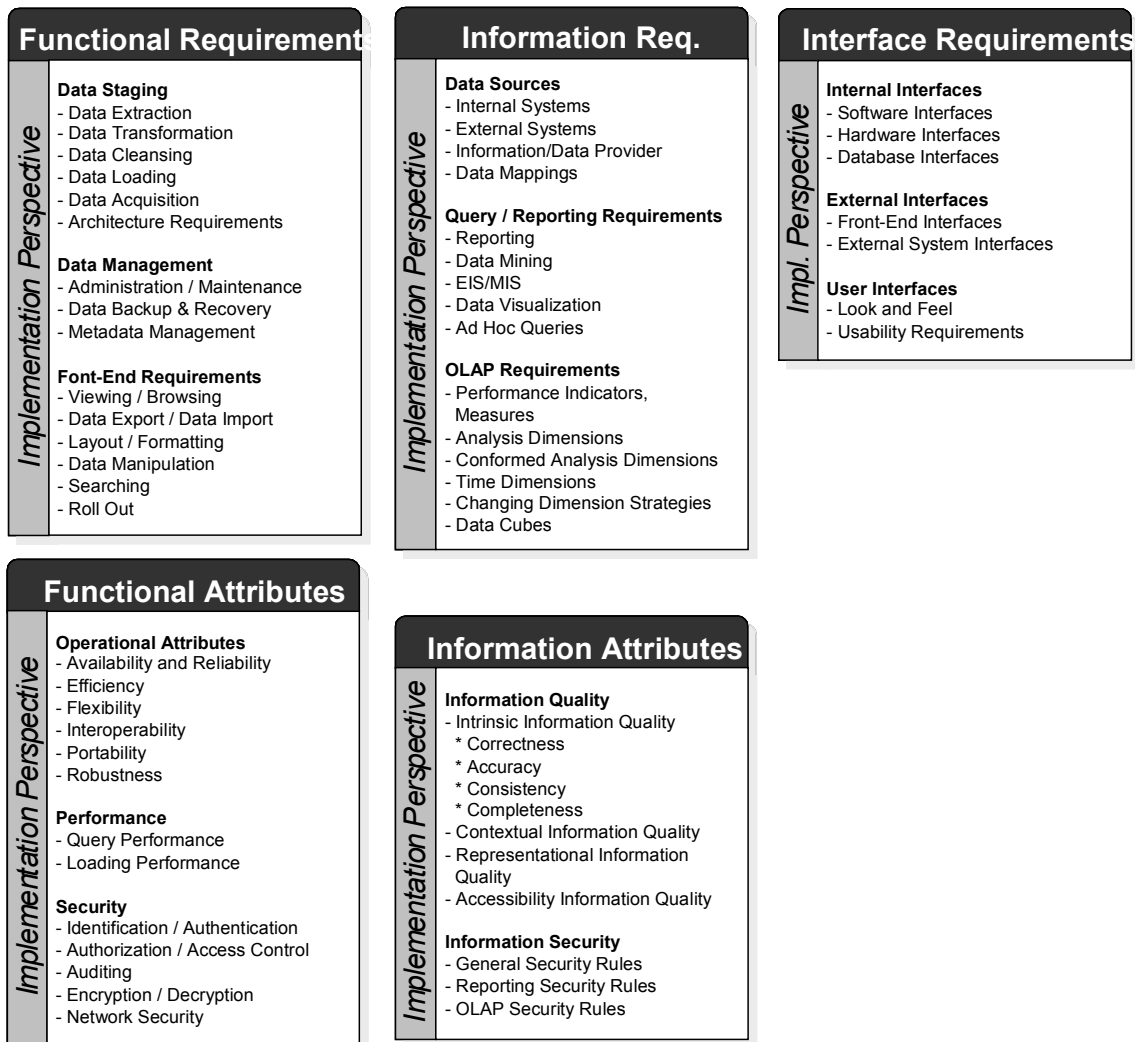


Figure 5. Requirements Template for Implementation Perspective

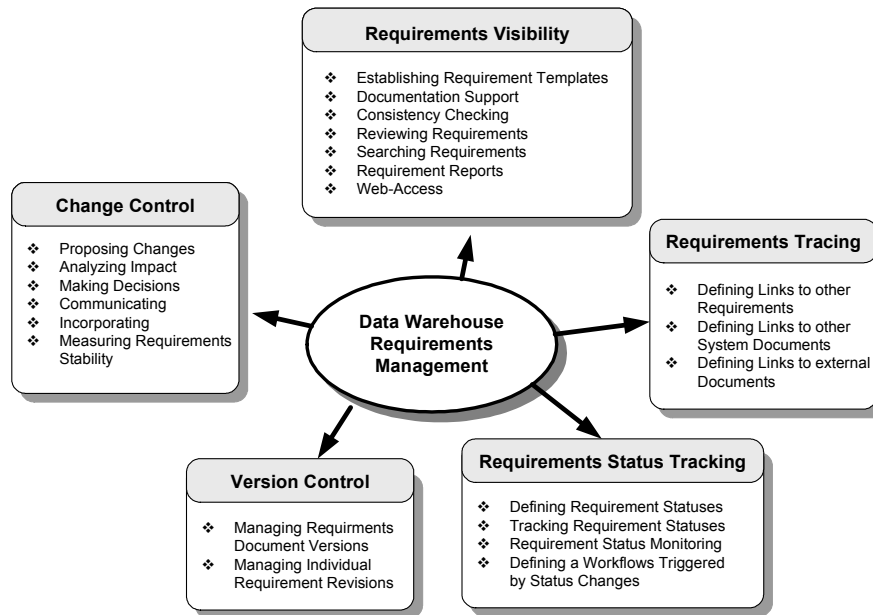


Figure 6. Major Requirements Management Activities for Data Warehouse Systems

Requirements Management Activities

The entire domain of requirements engineering can be subdivided into *requirements development* and *requirements management*. Requirements development involves eliciting, analyzing, documenting, and validating of requirements. Once data warehouse requirements are reviewed and approved, they define the requirements baseline for the implementation. The baseline constitutes an agreement between the stakeholders and the data warehouse team about the requirements for the data warehouse system.

The requirements agreement is the bridge between requirements development and requirements management. Requirements management includes all activities for establishing and maintaining the integrity and accuracy of the requirements agreements as the data warehouse project evolves. Requirements management includes the following activities (see also Figure 6):

- Controlling changes to the requirements baseline
- Keeping project plans current with the requirements
- Controlling versions of individual requirements and requirement documents
- Managing the relationships between requirements, and links or dependencies between individual requirements and other deliverables
- Tracking the status of the requirements in the baseline
- Support for documenting the requirements
- Facilitating the requirements visibility by providing browsing, searching and reporting functionalities

Change Control

Change control is concerned with the procedures, processes and standards, which are used to manage changes to data warehouse requirements. Change control ensures that similar information is collected for each proposed change and that overall judgments are made about the costs and benefits of the proposed changes (Sommerville et al. 1998). Without formal change control, it is impossible to ensure that proposed changes to the requirements support the fundamental business goals of the data warehouse system.

In order to ensure a consistent approach to change control, organizations may define a set of change control policies. These cover:

- The process for the change request and the information required to carry out each change request

- The process used to analyze the impact and costs of change and the associated traceability information.
- The formation of a change control board, which formally considers the change requests and decides which changes to implement.
- The software support for the change control process.

Change Control Process

The process of requirements change control consists of a set of activities for documenting, reporting, analyzing, costing and implementing changes to a set of data warehouse requirements. The change control process can be seen as a three-stage process. First, some problems with data warehouse requirements are identified. This could come from an analysis of the requirements, new stakeholder needs, or operational problems with the data warehouse system. The requirements are analyzed using problem information and requirements changes are proposed. In the second stage, the proposed changes are analyzed to see how many other data warehouse requirements are affected by the change and roughly how much it would cost (in both time and money) to implement the change. In the final third stage, the change is implemented. A set of amendments to the requirements specification or a new version of the requirements specification is produced.

A well-defined change control process provides stakeholders with a formal mechanism for proposing changes in requirements. The change control process allows tracking of the status of all proposed changes in the requirements specification, and it helps to ensure that no suggested changes are lost or overlooked. Once data warehouse requirements are baselined, they must follow the change control process for all proposed changes to them.

A change control process should not become an obstacle to making changes in the requirements specification. Instead, it's a funneling and filtering mechanism to ensure that the most appropriate changes are adopted and that the negative impacts on the data warehouse project are minimized. The change control process should be well-documented, as simple as possible, but nevertheless effective.

Change Control Board

The change control board is the body of people, be it one individual or a diverse group that makes binding decisions about which proposed requirement changes and suggested new data warehouse extensions to approve. Data warehouse projects can have several levels of control boards, some of which are responsible for making business decisions (such as requirements changes) and some of which make technical decisions. Some change control boards are empowered to make decisions and simply inform management about them, while others can only make recommendations for management decisions. A higher-level change control board will likely have the authority to approve changes with greater impact on the data warehouse project than a lower-level board.

An effective change control board considers all proposed changes at regular intervals and makes timely decisions based on analysis of the potential impacts and benefits of each proposed requirement change. The change control board membership should represent all groups that could be affected by a requirement change that lies within the scope of its authority. The change control board might include representatives from following areas:

- Project management
- Database management
- Data staging
- Development
- Testing
- Deployment
- User representatives

Change Control Policy

The data warehouse team should clearly communicate a policy that states its expectations of how proposed requirements changes will be handled. Following are change control policies proved to be practical (Wiegers 1999):

- All requirements changes must follow the change control process. If a change request is not submitted in accordance with this process, it won't be considered.

- No design or implementation work other than feasibility exploration will be performed on unapproved changes.
- Simply requesting a change doesn't guarantee that it will be implemented. The change control board will decide which changes to implement.
- The contents of the change requests must be visible to all stakeholders.
- If change requests are modified, the original text must be maintained (version management).
- Every requirement change must be traceable to an approved change request.

Tool Support for Change Control

Change management involves handling large amounts of information and passing it between individuals in the organization. It is necessary to keep track of which changes have been proposed, which have been approved by the change control board, which have been implemented, which are still under consideration, etc. The whole change control process can benefit enormously from effective tool support. The capabilities of such a tool support may include:

- Electronic change request forms, which are filled in by different participants in the change control process.
- A database to store and manage these forms.
- A change model which may be instantiated so that people responsible for one stage of the change control process know who is responsible for the next process activity.
- Electronic transfer of forms between people with different responsibilities and electronic mail notification when activities have been completed.
- Direct links to the requirements, which are affected by the change request.
- Measuring requirement change activities.

Version Control

Version control is an essential aspect of managing data warehouse requirements. Every version of the requirements document must be uniquely identified. Stakeholders and the data warehouse team must be able to access the current version of the requirements, and changes must be clearly documented and communicated to everyone affected. To minimize confusion, conflicts, and miscommunication, only designated individuals should have the permission to update requirements. Each published version of the requirements documents should include a revision history that identifies the changes made, the date each change was made, the individual who made the change, and the reasons for each change.

Tool Support for Change Control

The most powerful approach to version control is to store the requirements in a document database. This way, requirement management tools can easily track and report the complete history of changes made to every requirement, which is valuable when an earlier version of a requirement needs to be reverted. The easyREMOTE^{DWH} prototype stores requirements and project documents in a full-text database (Lotus DOMINO® server) that includes a version-tracking feature to maintain a history of changes to a document.

Requirements Status Tracking

Tracking the status of each requirement throughout the development of the data warehouse system is an important aspect of requirements management. Overall project monitoring is improved if periodical reports can be delivered regarding the percentage of the entire requirements set that exists in each possible status category. Classifying requirements in status categories is more useful than trying to monitor the percent completion of each requirement (Wieggers 1999).

Tool Support for Requirements Status Tracking

The easyREMOTE^{DWH} prototype allows monitoring the requirements by a requirement development status and several requirement management statuses. Table 1 shows the states of the requirement development status.

Table 1. States for the Requirement Development

State	Description
<i>Proposed</i>	The requirement has been requested.
<i>Approved</i>	The requirement has been analyzed, its impact on the rest of the data warehouse project has been estimated, and it has been baselined. The data warehouse team has committed to implement the requirement.
<i>Designed</i>	The design elements that address the requirement have been created and reviewed.
<i>Implemented</i>	The code for the implementation of the requirements has been designed, written and tested
<i>Deleted</i>	The requirement has been deleted from the baseline. An explanation of why and by whom the decision was made to delete the requirement should be included.

Table 2 lists the requirement management statuses, which allow a more detailed monitoring of the requirements. Requirement management statuses help stakeholders and the data warehouse team to assess better requirement characteristics, which are represented by requirement management statuses.

Table 2. Requirements Management Statuses

Status	Description
<i>Approval Status</i>	The current status of the requirement approval. Possible states are: Not Approved, Proposed for Approval, Approved
<i>Completion Status</i>	Determines whether the requirement is complete. Possible states are: Complete, TBD (To Be Determined)
<i>Inspection Status</i>	This status indicates whether the requirement was inspected. Possible states are: Not Inspected, Proposed for Inspection, Inspected
<i>Change Status</i>	If a requirement is to be changed, this status is set. Possible states are: Not Changed, Proposed for Change, Changed
<i>Rework Status</i>	Requirements, which have to be reworked, are indicated by the rework status. Possible states are: Not Reworked Proposed for Rework, Reworked
<i>Release Status</i>	This status shows when the requirement has to be implemented. Possible states are: Next Release, Release 1, Release 2, ..., Future, Deleted
<i>Prototyping Status</i>	If a requirement is to be prototyped, this status is set. Possible states are: No Prototyping, Propose for Prototyping, Prototyping in Progress, Prototyping Successful, Prototyping Not Successful

Requirements Tracing

A critical part of the requirements management is the assessment of the impact of a change on the rest of the data warehouse system. If the change is proposed while the requirements are being developed, the change affects on other requirements have to be assessed. If the change is proposed during the implementation phase, the impact assessment involves assessing how the change affects the requirements, the data warehouse design and its implementation.

Traceability links allow you to follow the life of a requirement both forwards and backwards, from origin through implementation (Gotel and Finkelstein, 1994). Figure 7 illustrates the four types of requirements traceability links. Stakeholder needs are *traced forward* to requirements, so that it is possible to identify the affected requirements, if they need changes. Conversely, you can

trace *backward* from requirements to stakeholders needs to identify the origin of each data warehouse requirement. If business requirements and use cases represent the stakeholders' needs, the left half of Figure 7 illustrates tracing between business requirements/use cases and detailed system requirements. The right half of Figure 7 shows that you can trace also *forward from* requirements by defining links between individual requirements and specific data warehouse downstream products (i.e. design, code or other artifacts). The fourth type of link traces data warehouse downstream products backward to requirements so that you know why you created each data warehouse element. These traceability links avoid including data warehouse elements that don't relate to user-specified requirements.

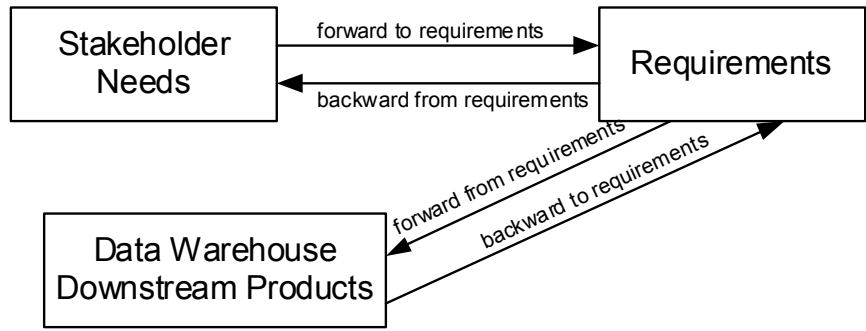


Figure 7. Types of Requirements Traceability

Tool Support for Requirements Tracing

Requirements traceability cannot be fully automated because the knowledge of the links originates in the minds of the stakeholders and the data warehouse team. However, once traceability links are identified, they should be documented with the requirements.

The easyREMOTE^{DWH} prototype distinguishes two types of requirements dependencies between requirements: 1) top-down dependencies and 2) cross dependencies. The top-down traceability links reference requirements of a higher abstraction level. On the other hand, requirements, which have an effect on requirements of the same abstraction level or on other documents, are cross-referenced.

Requirements Visibility

Visibility techniques are vital for the requirements development and management. An easy access to the requirements facilitates the communication between stakeholders and the data warehouse team. In the following we list issues, which should be considered for improving requirements visibility to enhancing the communication between all involved parties.

Tailored and Modifiable Requirements Specification Template. The requirements specification template defines the overall structure for the requirements of the data warehouse system. The requirements specification template should be modifiable during the requirements process. The requirements analyst should have the possibility to begin at the project blastoff with a requirements specification template, which is fine-tuned over time according to the needs of the data warehouse project. That means, the requirements specification is evolving during the requirements process, the more information about the data warehouse system is discovered and collected. A general template for data warehouse requirements was introduced in section 2.

Customizable Requirement Forms. Requirements can be grouped into categories, which share the same requirement structure. For each category a requirement form must be developed. For instance, because of their different requirements structures, use cases and business requirements use separate requirement forms. Furthermore, individual requirement forms ensure consistency for the requirements, and allow an automatic checking of the completeness of each requirement. The requirements forms of easyREMOTE^{DWH} prototype are dynamic forms, which automatically adjust the form fields according to entered data. For instance, if a requirements analyst decides to specify a use case, automatically the fields for use cases are displayed and other fields are hidden.

Establishment of a Glossary. The glossary is a shared repository for the definition of general terms, which are used in the requirements specification. The glossary clarifies the meaning of the defined terms and can solve ambiguity problems in advance. The easyREMOTE^{DWH} prototype provides a glossary as an alphabetized collection of specialized terms with their meaning, which occur during the requirement development.

Reviewing Process. An integrated reviewing process helps the data warehouse team and the stakeholders to effectively communicate requirement proposals. A reviewing process assures a common agreement on the requirements that satisfy all stakeholders of the data warehouse system.

Access Control. For creating and changing requirements, individual groups of users have a different level of access to the requirements specification. Certain requirements activities can only be executed by users, which belong to a certain user group. For instance, the approval or change of requirements can be only executed by users who have the permissions for this action. The easyREMOTE^{DWH} prototype manages the document access by an access control list. Furthermore, the prototype supports electronic signing for requirement documents, which is an important feature for the approval of requirements.

Communication with Stakeholders. The data warehouse team and the stakeholders should be able to discuss requirements issues electronically through threaded conversations. Therefore, web-enabled discussion databases and news servers can be very useful. The easyREMOTE^{DWH} prototype provides a discussion database for the members of the data warehouse project. The discussion database is an informal meeting place, where the stakeholders and the data warehouse team can discuss requirements and share their thoughts and ideas.

Subscription/Newsletters. Automatically triggered notifications inform users when a new requirements is created or when an existing requirement is modified. This feature allows users to define trigger events for requirements they want to be informed about. Users of the easyREMOTE^{DWH} prototype can subscribe to newsletters, which contain a summary of requirement activities and links to the affected requirement documents. The user can define preferences in the newsletter profile (see Figure 8), which events should trigger a newsletter notification. The newsletters are delivered by e-mail.

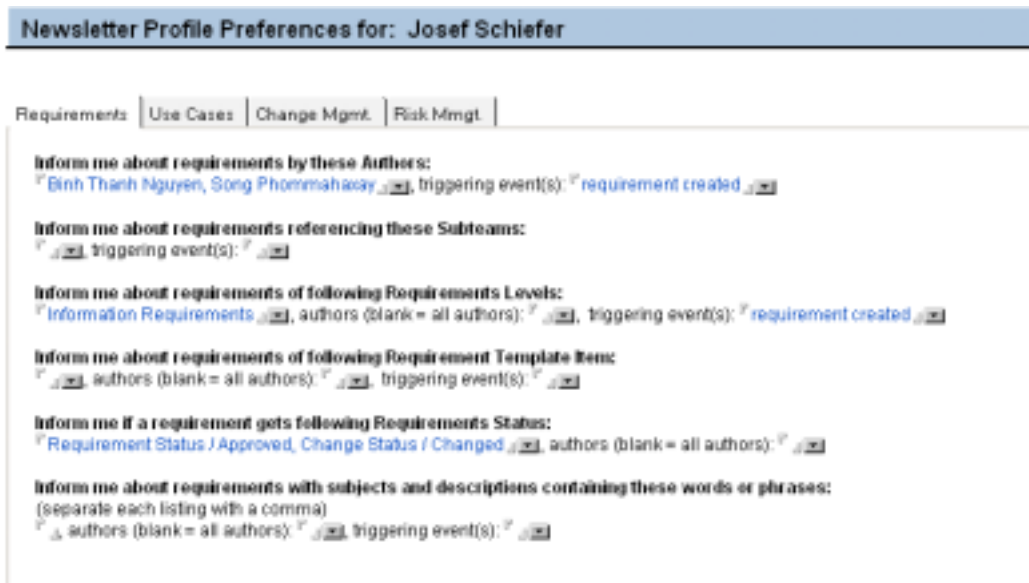


Figure 8. easyREMOTE^{DWH} Prototype: Newsletter Profile

Requirements Search. Because the requirements specification of a data warehouse system can become very comprehensive, it is important that stakeholders and members of the data warehouse team can effectively search for requirements. The search capabilities should include at least a field and full text search. Because Lotus Notes® is a document management system and offers a wide range on document search capabilities, the easyREMOTE^{DWH} prototype allows flexible ways for searching requirement documents. The prototype supports four kinds of document search: 1) simple full text search, which searches for requirement documents that contain the entered search string somewhere in the requirement, 2) field search, which looks for the entered search string in specific requirement fields, 3) word variants search and 4) fuzzy search, which widens the simple full text search, by making it more comprehensive.

Requirement Reports. Requirement reports provide insight into the requirements specification and the data warehouse project. The measurements in the requirement reports should be motivated by the questions of the stakeholders and the data warehouse team, and the goals that should be achieved. The easyREMOTE^{DWH} prototype provides several requirement measures, which are

available in document views. For instance, the requirement measurements allow the data warehouse team to assess the stability and completion of the requirements specification.

Navigation, Document Linking. Users should be able to easily follow traceability links or cross-references to other documents. Direct document links facilitate the navigation within the requirements specification. Document links and hyperlinks facilitate the implementation of traceability links and cross-references.

Web Access. Stakeholders and the data warehouse team should be able to access the requirements via web interface. A simple web browser should be sufficient for browsing and writing requirements.

Conclusion and Future Work

The requirements management process for data warehouse systems is a large and complex job. Lack of control and transparency can result in project overrunning costs and slipping schedules. Automation and integration are essential to improving the effectiveness of both developing requirements and managing the process. In this paper we introduce the easyREMOTE^{DWH} approach, which is a tailored requirements management approach for data warehouse systems. The presented approach represents “good practices” for requirements management. It focuses on the communication between the stakeholder and users of a new data warehouse system and those who are building it. Furthermore, it helps to apply an effective requirements engineering method by the use of different perspectives for the requirements capture. We discussed five major requirements management activities in detail and listed requirements for a proper tool support of these activities. Further problems we want to consider in the future include the validation and verification of information requirements, the estimation of implementation costs based on requirements, and workflow management technology for the requirements management.

References

- Bickerton, M., *A practitioner's handbook of requirements engineering methods*, Oxford University Press, 1992.
- Coad, P.; Yourdon, E., *Object-Oriented Analysis, Englewood Cliffs, N.J.*: Prentice Hall 1991
- Davenport, Th., *Process Innovation: Reengineering Work through Information Technology*, Harvard Business School Press, 1993
- Davis, Alan M., *Software Requirements: Objects, Functions, and States*, Prentice Hall, 1993.
- Gotel, O., Finkelstein, A., *An Analysis of the Requirements Traceability Problem, First International Conference on Requirements Engineering*, Los Alamitos, IEEE 1994.
- List, B., Schiefer, J., Tjoa, A M., *Use Case Driven Requirements Analysis for Data Warehouse Systems, Data Warehousing 2000*, Friedrichshafen.
- Robertson, S., Robertson, J., *Mastering the Requirements Process*, Addison-Wesley, 1999.
- Sommerville, I.; Kotonya, G., *Requirements Engineering: Processes and Techniques*, Horizon Pubs & Distributors Inc., 1998.
- Wieggers, K., *Software Requirements*, Microsoft Press, 1999.
- Inmon, William H., *Building the Data Warehouse*, John Wiley & Sons, 1996.
- Kimball, R., Reeves, L. Ross, M., Thornthwaite W., *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses*, John Wiley & Sons, 1998.
- Schneider, A., Wong, B., Wong, K., Maloney, G., *Data Warehousing*, Bentley College Community, 1999.