

**RPTSS 2018**  
**International Conference on Research Paradigms**  
**Transformation in Social Sciences**

**PECULIARITIES OF FUNCTIONAL REQUIREMENTS OF  
AUTOMATED ENTERPRISE RESOURCE MANAGEMENT  
SYSTEM**

R.D. Gutgarts (a)\*

\*Corresponding author

(a) Irkutsk National Research Technical University, 83 Lermontov St, Irkutsk, Russian Federation,  
gutgarc@gmail.com, 89643596493

*Abstract*

The article is devoted to the peculiarities of the development of functional requirements for automated enterprise resource management systems (ERP-system). An approach to the formulation of requirements for systems of this class, based on the use of classical management functions (planning, accounting, control, analysis, regulation) is considered. Particular attention is drawn to the universal nature of such formulations. In future, when developing ERP-systems, this creates good prerequisites for the correct algorithmization of the requirements of potential users. The article gives examples of incorrect task names and recommendations for their editing in the correct formulations, which are algorithmically justified. This helps to minimize the problems associated with incomplete and incorrect formulation of requirements. A set of tasks formulated on the basis of management functions determines the functionality of the system being designed. Functionality is understood as a set of algorithms implemented within the framework of one project to create an ERP system. The document proposes a two-level scheme for the formation of requirements. The first level: the formation of a list of tasks using the management functions in their name. This allows us to apply the principle of uniformity to the names of problems, taking into account their further computer processing and lead to a classical structural representation: initial data → algorithm for processing → result. Second level: a description of each task for programming. Thus, it is possible to take into account practically all the features of each problem under consideration..

**Keywords:** ERP-system, functional requirements, automated enterprise resource management, management functions.



## 1. Introduction

Enterprise resource management automation, integrating different management functions, has been carried out for more than 50 years. However, there is no agreement between IT-specialists who design ERP-systems and enterprise workers who use software in their professional activities. The problem arises during the design of the system when potential users generate ERP-system requirements. The major cause is differences in the terminology used to describe functions and business processes of the ERP-system. As a result, much time is spent on negotiation and specification which increase project costs. As far as it is difficult to find a software compromise, many IT-projects are discontinued or partly implemented.

Requirement specifications are crucial for developing ERP-systems as types of functional software. For example, Wieggers and Beatty (2014) presented a 736-page research on software requirements. Their first research on this issue was published in 1999.

Among the key challenges of ERP-system development are:

- Lack of source data from clients (13% of projects).
- Incomplete requirements and specifications (12% of projects).
- Changes in requirements and specifications (12% of projects).

At present, all ERP-system development methods do not impose hard constraints on project start requirements. They can be edited at any stage. However, any changes increase the project life cycle and its costs. Therefore, it is recommended that at least initial project start requirements are correctly laid down.

According to Hull, Ken, and Jeremy (2005) the two key causes of functional software designing project failures are:

- Incomplete requirements (13.1%).
- Insufficient involvement of users (12.4%).

Thus, a quarter of all failures result from these two causes. At the same time, there are two crucial factors which contribute to successful projects:

- Active involvement of users in requirements generation (15.9%).
- Clear statement of requirements (13%).

### 1.1. Participants of requirements generation

Any ERP-system is applied. Therefore, all ERP-system requirements should be developed by specialists who will interact during the design, testing and operation of the system. The full list of these specialists was suggested by Wieggers and Beatty (2014) who divided them into three categories:

1. Specialists working outside an organization, which designs ERP-systems, but using information generated within the system (purchasers, purchasing specialists, subject experts, consultants, etc.).
2. Designing organizations (design managers, marketing departments, IT-designers, training managers, etc.).
3. Project teams (project managers, business analysts, designers, business decision designers, quality control analysts, process analysts, etc.).

Involvement of all these participants depends on:

- the scale of projects (project functions);
- internal features of automation objects (manufacturing and management technologies applied);

- relations between automation objects and connected systems and organizations;
- features of project management processes;
- a professional level of an ERP-system designing team.

## 1.2. Requirements generation issues

Formulation of functional requirements is carried out at the earliest stage of the ERP-system creation. Therefore, this process is one of the main ones. From the quality of its implementation depends the entire subsequent process of design and development (Baronov, Kalyanov, Popov, & Titovsky, 2006).

Sometimes the requirements are presented in a generalized form and an additional procedure for their detailing is required for implementation in the form of a program. In addition, functional requirements are not separately identified and are considered together with technological requirements (for example, storage of information) (Usov, 2007).

Some authors consider not specific requirements, but characteristics of requirements for various business configurations (simple configuration, mechanical bureaucracy, professional bureaucracy, adochocracy). For each configuration, the requirements groups are taken into account: where will be applied, operational activities, personnel, functionality, system architecture. At the same time, functionality is formulated only in general terms, which can not be implemented algorithmically in the original form and require very detailed additional consideration (Astapchuk & Tereshchenko, 2015).

Often the requirements for ERP-systems are considered only as technological requirements (for example, modular design, reliability, security and others). However, such requirements are common to any system that will be implemented as an application program. Functional requirements are described as the names of the modules included in the ERP system, that is, in general terms. Such a description does not reflect the algorithmic nature of the solution (Oleinik, 2012).

Analysis of only a few approaches allows us to draw the following conclusions:

1. There are no uniform standards for the formulation of functional requirements.
2. Information on requirements is of an advisory nature.
3. The stylistics of the formulation of requirements presents them only in a general way.
4. All the options considered require further elaboration.
5. None of the approaches makes it possible to present the requirements as a task for programming

The generation of automated system requirements is a challenge of ERP-system designing and development which consists of the following problems:

1. For a number of reasons, ERP-system users cannot provide exhaustive information upon request.
2. As far as each party involved includes specific professional specifications, some requirements can be contradictory.
3. It is difficult to determine requirements detail levels which can increase their amount, make it difficult to unify the requirements and impossible to manage them correctly.
4. Insufficient involvement of users.
5. Not all potential users are surveyed when designing requirements. Therefore, their requirements will not be implemented.

6. There are redundant requirements which can be neglected since they do not improve decisions but need time and financial resources (Wiegers & Beatty, 2014).

Another problem is the lack of a single structural form, into which primary requirements can be recorded. Therefore, there are different options for describing them, for example, a text or a graphic model (Gusyatnikov & Bezrukov, 2010).

## **2. Problem Statement**

According to Leffingwell and Widrig (2002), identification of requirement errors increases ERP-system development project costs. The cost of elimination of requirement errors identified at the system implementation stage is 20 times as much as the cost of elimination of requirement errors identified at the system designing stage. E. Hull, K. Jackson and J. Dick also maintain that view.

Thus, high quality development of requirements contributes to:

- Implementation of projects on time and within allocated budgetary resources.
- Minimization of possible changes.
- Decrease in system modernization costs.
- A high level of implemented project validation.
- 

### **2.1. Choosing task names when generating requirements**

It is useful to choose correct names of tasks which have to be implemented when designing a new system or modernizing an existing one. Task names have to include management functions which predetermine management functions implementation algorithms under the automated data processing, i.e. they simplify decision algorithm programming (Gutgarts, 2015).

Technical, technological, information and software functions of modern information technologies are diverse. They can implement any incredible functional requirements of potential users. However, it is necessary to identify these requirements and cast them into the correct form in order to simplify a formalization process. It is an obligatory programming condition. The requirements which cannot be described using logical dependences and formal presentations (unable to be algorithmized) cannot be implemented.

Future ERP-system users have to know which functions the system implements. However, at the designing stage, manufacturers cannot explain and formalize their wishes and IT-specialists cannot understand manufacturers since they use different terms.

To understand requirements of future ERP-system users, IT specialists have to go into details of manufacturing and/or management processes. Otherwise, they cannot carry out pre-project analysis. This process is often iterative. When IT specialists and manufacturers reach understanding, the ERP-system designing and development process starts.

The issues of relations between future users and IT specialists have existed since the beginning of development of applied automated systems for manufacturing companies. In 1973, the Information Bulletin of London University Computer Center published a comical picture of these relations (Pavlovskaya, 2004).

The issue of misunderstanding between parties involved is still crucial. Therefore, if required functions are not implemented, the project is useless (Gutgarts, 2008).

### 3. Research Questions

The article aims to study approaches to generation of ERP-system requirements and suggests an author's two-level requirements description scheme.

#### 3.1. Functional requirements generation methods

A. Coburn suggested an approach based on the following concepts:

1. Actors: somebody or something that is behaving.
2. Participants: somebody or something interested in the system behavior.
3. Key actors: participants who can initiate relations with the system for achieving specific purposes.
4. Use cases: behavioral scenarios.
5. Scope: a system designed.
6. Initial conditions and guarantees: what should be true before and after the use case implementation.
7. Main scenarios: standard solutions (main calculation processes) excluding errors.
8. Extension use cases (possible branches of the main calculation process) occurring under certain conditions violating standard use cases (non-standard processing procedures have to be applied) (Kobern, 2002).

Dealing with software requirements, D. Leffingwell and D. Widrig use the following concepts:

1. Requirement is a function of a system.
2. Requirements management involves recording, analysis and implementation of variable system requirements.
3. Tasks of clients which have to be described using their terminology. It helps understand the needs of users and implement them in the system.
4. Functions are requirements implemented in the system. When a set of functions negotiated with future users are identified, requirements are specified.
5. Precedents are use cases for obtaining data processing results upon the user request (Leffingwell & Widrig, 2002).

The key feature of these approaches is the lack of a unified presentation structure and bulky description.

### 4. Purpose of the Study

The research aims to demonstrate relations between task names during the functional ERP-system requirements generation and implementation.

#### 4.1. The first requirements generation level

Tasks names are presented through main management functions. Let us analyze examples of each management function.

*Recording* means that recorded information reflects performance of transactions. The task involves data basing from primary media (documents, oral messages, devices, etc.).

*Planning* (calculation, prediction) involves calculation procedures. The task is performed using information in the data base. Information which is not stored in the data base can be used for solving certain tasks.

*Control* involves calculation of deviations of planned data from factual (recorded) ones. The results can be kept or not kept in the data base depending on their further application.

*Analysis* involves selection of information from the data base and delivery reports to a specialist for recording, planning and control.

*Control* involves editing of planned information.

Thus, the use of functions as task names simplifies development of algorithms for automated decisions.

#### **4.2. The second requirements generation level**

Part 1: General characteristics of the task

1.1. Name (when using a management function) and assignment (for which objects it is designed).

1.2. Conditions and restrictions (if the use of a mathematical model is suggested); possible optimality criteria (for optimal planning tasks); alternate solutions (logical branches of the algorithm); restrictions on the search space.

1.3. Chronological features.

1.4. Relations with other tasks (through input and output data).

1.5. Restrictions imposed by related tasks.

1.6. Source data collection management (data sources and acquisition methods).

1.7. Chronological restrictions on solution results.

1.8. Specific features (if there are any).

Part 2: Data support.

2.1. Description of input information (a presentation form, semantic characteristics of indices and other features).

2.4. Reference data.

2.5. Output (resultant or analytical) information (a presentation form, semantic characteristics of indices and other features). The list and structure of controlled output information. The availability and assignment of the query system. Purposes of information (solution results) utilization.

Part 3: Mathematical support (task solution algorithm describing a standard solution path and all possible non-standard situations occurring when solving the task.

Part 4: Description of a test case. Identification of the quality and amount of source data which will be used for testing the task solution correctness.

Depending on the task, some parts can be omitted or expanded (specified).

## **5. Research Methods**

The first column contains task names which were defined by surveyed specialists. The second column contains correct task names.

**Table 01.** Examples of task names based on management functions

<b>Initial (incorrect) task name</b>	<b>Correct task name</b>
Recording of working hours of each employee based on the visit log	Recording of working hours
Comments. The word “each” is redundant. As far as the record of working hours is based on the visit log, it is carried out using factual data in the document. Specification “according to the visit log” should be taken into account when describing tasks for input data programming when the log structure and details will be described.	
Recording of employees using private offices	Analysis of offices for one employee
Comments: 1. Incorrect recording (“offices” and “employees” are different objects, and information about them is kept in different tables; the tables are linked). 2. A lot of redundant words. 3. To identify employees having private offices, it is necessary to apply an analysis function (a filter to select information by the specified feature is used).	
Recording of personal data about users and competition results	1. Recording of competitors 2. Recording of competition results
Comments: 1. One task name includes two objects (users and competitions). 2. As a result, one task name includes two different tasks. 3. Recording of competitors involves recording of information about competitors. Quality and numerical representation of information is determined when describing a solution algorithm.	
Recording of results of all competitions	Recording of competition results
Comments. The word “all” is redundant since recording involves the fixation of factual information which means that invalid competitions will not be taken into account.	
Analysis of the number of wins, speed and time assessment for each competitor.	1. Analysis of competition results. 2. Analysis of results of winners
Comments: 1. There is no need to explain “what is being analyzed”. Any explanation refers to a solution algorithm. 2. The algorithm can be divided into two tasks. 3. For one tasks, winners’ results can be included in the analytical document (arranged by target parameters), since the win is determined by values of registered parameters. 4. For both tasks, only selected parameters can be included in the analytical document.	

Specialized economic researches show that task names are defined with regard to management functions. It can be used when generating ERP-system requirements. Let us give some examples. When planning, the following tasks are solved:

- Calculation of resource costs.
- Calculation of production costs.
- Calculation of prime costs for each good.

To solve analytical tasks, the following reports (analytical tasks) are generated by managers of different levels:

1. Consolidated reports (generated monthly):
  - Profit and loss reports.
  - Balance sheet reports.
  - Cash flow reports;
2. Current reports (generated monthly or weekly):
  - Resource and material reports.

- Manufactured products reports.
- Debit debt reports, etc.
- 3. Short-term reports (generated daily):
  - Sales reports.
  - Incomplete production reports.
  - Purchase reports.
  - Production cost report.
  - Commercial and management cost reports, etc. (Kuzmina & Akimova, 2015).

For the subject area “Management analysis”, the following tasks are described:

- Preparation of financial plans.
- Forecast financial planning.
- Current and short-term financial planning.
- Analysis of purchasers.
- Market segmentation.
- Analysis of the structure, dynamics and state of capital assets.
- Analysis of capital asset utilization performance indices (Nikiforova & Tafintseva, 2015).

Among management accounting tasks are:

- Analysis “costs – production output – profit”.
- Calculation of sales costs.
- Calculation of direct materials costs.
- Deviation analysis (when calculating standard production costs).
- Calculation of purchased and used materials.
- Labor performance control.
- Variable overhead cost control.
- Calculation of product life cycle costs (Druri, 2003).

## 6. Findings

To generate functional requirements for automated enterprise resource management systems, future users have to define task names correctly. Task solution algorithms rather than procedure algorithms are programmed. Task solving is transformation of input information (information query, report generation, source data for calculations, etc.) into output information (response, generated report, a data base fragment of calculations, etc.) according to the target algorithm under imposed restrictions (if there are any). Thus, during one procedure, one or several tasks can be implemented. The use of management functions in task names simplifies programming tasks. It helps reduce project costs and life.

## 7. Conclusion

The suggested two-level requirements description scheme can be used at ERP-system requirements generation and software implementation stages.

The scheme is universal and can be applied in any subject area.

## References

- Astapchuk, V.A., & Tereshchenko, P.V. (2015). *Corporate information systems: requirements for design: a textbook for universities*. Novosibirsk State Technical University - 2nd edition, revised and enlarged. Moscow: Uright, 36-37.
- Baronov, V.V., Kalyanov, G.N., Popov, Yu.N. & Titovsky, I.N. (2006). *Information technology and enterprise management: monograph*. Moscow: Akademia IT, 137-138.
- Gusyatnikov, V.N., & Bezrukov, A.I. (2010). *Standardization and development of software systems*. Moscow: Finance and statistics, INFRA-M. 132.
- Gutgarts, R. D. (2008). *Analysis of information in the enterprise. Problems of theory and practice of management, 2*, 62-68.
- Gutgarts, R. D. (2015). *The role of business processes in the formation of requirements for the ERP system. Problems of theory and practice of management, 1*. 118-127.
- Druri, K. (2003). *Management accounting for business decisions*. Moscow: UNITI-Dana, 85,343,345,431,435,438,440,532.
- Hull, E., Ken, J. & Jeremy, D. (2005). *Requirements development and management*. Telelogic, 2, 8-9.
- Kobern, A. (2002). Modern methods for describing functional requirements for systems. Moscow: Lori, 1-3.
- Kuzmina, M.S., & Akimova, B.Z. (2015). *Enterprise (organization) cost management*. Moscow: KNORUS, 48,191-193.
- Leffingwell, D., & Widrig, D. (2002). *Principles of work with software requirements*. Moscow: Williams, 35, 37-38, 41-42.
- Nikiforova, N.A., & Tafintseva, V.N. (2015). *Managerial Analysis*. Moscow: Uright, 26,180-181, 222-223.
- Oleinik, P.P. (2012). *Corporate information systems: a textbook for bachelors and specialists: in the direction 080800 "Applied Informatics" and other economic specialties*. St. Petersburg: Peter. 10-17, 99-104.
- Pavlovskaya, T.A. (2004). *C/C++. Programming in a high-level language*. Saint Petersburg: Piter, 111.
- Usov, A.B. (2007). *Requirements for Information and Analytical Control Systems for Industrial Enterprises. Automation and Modern Technologies, 7*, 43-46.
- Wieggers, K., & Beatty, J. (2014). *Development of software requirements*. Moscow: Rus. Red: BHV, 5, 29.