CHAPTER 5

The task of quantitative assessment of changes in the long-term IT project management system

Maksym Yevlanov
Nataliia Vasyltsova
Iryna Panforova
Anastasiia Popova

Abstract

The object of the study is the process of change management in an IT project.

The study addressed the problem of quantitative assessment of changes that arise during the implementation of a long-term IT project. Existing common methods of change management do not allow for quantitative assessment of the main parameters of changes. Modern research is mainly aimed at solving the problem of quantitative assessment and change management within the entire IT project. The issue of quantitative assessment of changes that arise in the process of work of individual teams of IT project performers remains practically unexplored.

In the course of the study, it was proposed to improve the existing method for quantitative assessment of changes, based on the Beckhard and Harris's model. To improve this method, it was decided to use a descriptor approach. An improved method was developed, which, unlike the existing one, allows for quantitative assessment of changes based on the values of descriptors. These descriptors are formed in the process of performing sprints of a long-term IT project by its performers.

Based on the improved method, elements of information technology for automated solution of the problem of quantitative assessment of changes in the management system of a long-term IT project have been developed. A description of the architecture of information technology has been proposed, its technological stack has been defined, and the results of the development of software elements have been presented.

Experimental verification of the obtained research results was carried out within the framework of the long-term IT project "Web Constructor". The verification was carried out on the results of the work of one team of project developers.

 DOI: 10.21303/978-9908-9706-8-4.ch5

The calculation was based on data obtained during the team's execution of seven project sprints. The verification results showed that the predicted values of the indicators quite accurately coincide with the values of the time actually spent on implementing changes (0.97 and 0.81 for the sixth and seventh sprints of the IT project, respectively).

Overall, the application of the improved method improved the change implementation rates during the IT project implementation and improved the team's attitude to the change management process by 14.5%. This allows to consider the improved method and implemented IT elements as a usable tool for change management of individual teams of IT project performers.

Keywords

IT project, change, Beckhard and Harris's model, least squares method, Huber method, descriptor, sprint.

5.1 Introduction

The last decades are characterized by the widespread use of long-term projects, the implementation period of which is three or more years. Such projects are typical for various branches of the IT sphere: development and implementation of information technologies (IT), information systems (IS), IT infrastructure, scientific research in this field, etc. [1, 2].

Analysis of the characteristics of modern long-term IT projects allows to highlight their main features [2]:

– complexity of description, caused by a large number of functions, processes, elements, data and connections between them;

– the presence of a set of closely interconnected subsystems, each of which has its own local goals and objectives;

– the need to integrate existing and re-developed applications;

– heterogeneity of individual groups of developers in terms of the level of classification and established traditions of using tools;

– a constant flow of changes in modern IS.

A significant number of changes in the implementation of such projects occur at the operational level of tasks, which makes high-quality change management at this level necessary. For example, poor quality of time-to-market time-variance assessment of individual project operations can lead to project overruns, resource overstrain, and unforeseen delays in product launch. On the contrary, systematic time-to-market time-variance assessment allows project management to make informed decisions regarding resource planning, task allocation, and priorities in the event of changes in project requirements or conditions. Implementing such an assessment as a separate functional task of the long-term IT project management system helps reduce the risks of project delays and ensures its successful completion within the established deadlines.

The modern point of view considers change management as one of the tasks of change management. The term "change management" should be understood as a comprehensive, cyclical, and structured approach to changing individuals, groups, and organizations from their current state to a future state with expected business benefits [2, 3]. This approach is now described in the form of a framework – conventions, principles and methods of change management in the field of program management, project portfolios and individual projects. But, unfortunately, the existing change management framework [3] and standards and body of knowledge on IT project management [2, 4] do not provide specific recommendations for the selection of change management methods and their implementation in IT project management systems. Therefore, conducting research on solving change management problems as elements of the IT project management system is relevant from a theoretical and applied point of view.

5.2 Analysis of the approach and modern methods of project change management

The general features of the implementation of processes, subprocesses and individual works of the existing change management framework are determined by the change management life cycle adopted in [3]. According to this life cycle, each of the change management subprocesses is proposed to be considered as a set of individual works, each of which is performed in the order established by the framework (but not necessarily strictly sequentially one after the other). The entire change management process and its subprocesses in [4] are proposed to be described using an iterative life cycle model, taking into account the possibility of constant occurrence of adaptive changes in response to changing circumstances. Such a representation of the change management life cycle allows minimizing the connections between individual subprocesses and works of the existing change management framework and considering them as separate independent objects of scientific research. At the technological level, this means the possibility of creating and using for automation of the corresponding subprocesses and works of individual IT, which interact with each other according to the service-oriented paradigm.

Since the existing regulatory documents do not provide recommendations for choosing a specific change management method in an IT project, it was decided to consider the features of the change management methods that are considered the most common. These methods include [5]:

– Prosci ADKAR model [6, 7];
– the Accelerating Implementation Methodology (AIM) method [8, 9];
– Beckhard and Harris's model [10];
– Bridges Transition model [11];
– Kotter's 8-Step Change model [12];
– Kübler-Ross model [13];
– a method based on the Kurt Lewin model [14].

ADKAR is an abbreviation that represents five key stages necessary for the successful implementation of changes: "Awareness", "Desire", "Knowledge", "Ability", "Reinforcement" [7]. The Prosci ADKAR model was created by J. Hyatt in the late 1990s. This method offers a deep approach to change management, focused on personnel. It is assumed that in the process of implementing changes, it is necessary to clearly explain to personnel the reasons for the changes and their importance, which contributes to the involvement of personnel in the process. Next, it is important to train each employee in the methods of implementing changes, which allows to show the level of their professional knowledge and skills in the process of introducing changes. The final stage involves consolidating the changes, ensuring the sustainability of innovations in the organization's activities [6].

The Prosci ADKAR method is part of a broader approach to change management developed by PROSCI. It is based on the method using the ADKAR model, but includes additional tools, methods and practices aimed at effectively implementing changes in the organization [7].

AIM is a powerful and disciplined method for managing organizational change, including transformational change, until the full return on investment. AIM can be applied to any type of initiative or project, but most organizations direct the main resources and energy to the technical and business process components [8, 9]. One of the advantages of this method is the ability to systematically analyze and anticipate possible difficulties in the change process, which allows organizations to prepare and respond to them more effectively. However, AIM can require significant resources and time spent on using the method, and can also become difficult in the event of adverse circumstances or unforeseen events during the change implementation process [9]. The method, based on the Beckhard and Harris's model, provides five stages of change management aimed at identifying the need for change, developing a strategy for its implementation, forming an action plan and identifying responsible executors. The main advantage of the method, based on the Beckhard and Harris's model, is a systematic approach to the change process, which allows to structure it and manage it effectively. Praxie has developed software applications for change management using a method based on the Beckhard and Harris's model and provides training [10, 11].

The method, based on the Bridges Transition model, enables organizations and individuals to better understand the human and organizational aspects of change and manage them effectively. The main advantage of this method is its focus on internal transition, which can provide a deeper and more sustainable change in the organizational environment. However, it may be less effective when it is necessary to respond quickly to external and unpredictable changes, as it focuses on the internal aspect of the transition [11].

The method, based on the Kotter's 8-Step Change model, is designed to increase staff involvement in change management and ensure its acceptance by all employees. The method includes the following eight stages [12]: "Creating the emotional need for change", "Building a coalition", "Building a vision", "Communicating the vision", "Implementing actions", "Formation of short-term achievements", "Consolidation of achievements", "Implementation of changes into the culture". Skipping one of these stages can lead to problems in making changes and complicate the change process.

The main advantage of the method is its systematic approach to change management and emphasis on involving personnel in the process.

The following can be noted as disadvantages of the method based on the Kotter's 8-Step Change model:

– implementation of all eight stages of the method may require significant effort and time, especially in large organizations or with a large scale of changes;

– the method focuses mainly on the organizational aspects of changes, leaving aside individual emotions and needs of employees;

– the method does not always take into account external influences, such as economic or political factors, which may affect the success of changes.

The method, based on the Kübler-Ross model, describes the stages of personnel behavior change, including: personnel resistance to change, lack of awareness of the consequences of change, personnel adaptation to new working conditions, positive attitude of employees to change and their acceptance. This method is important for understanding and predicting personnel reactions to changes in the organization. However, it is worth considering that the reaction to change can be individual and does not always correspond to this sequence of stages. From a practical point of view, the method can be used to identify any obstacles in the early stages of change projects and develop appropriate strategies. The main advantages of the method include the fact that it is simple but effective for managing organizational changes. The disadvantage of the method is that since all employees react at different speeds, they are at different stages marked on the change curve. This often leads to problems when planning changes [13]. The method, based on the model of Kurt Lewin and developed in the middle of the twentieth century, remains one of the most popular at the current stage of change management development. This method is notable for its systematicity and practicality, which makes it effective for implementing changes in various types of organizations. However, it can require significant resources and time to implement, and may be less effective in the case of complex organizational structures or indiscriminate application of change management methods [14].

To compare the considered methods of change management in projects, the study proposed basic criteria (indicators). The results of the comparison of change management methods by criteria are given in **Table 5.1** [5].

Table 5.1 Comparison of existing methods of change management in projects

Indicator	Method No. 1	Method No. 2	Method No. 3	Method No. 4	Method No. 5	Method No. 6	Method No. 7
Ease of use	–	–	+	–	–	+	+
Complexity of implementation	–	–	+	–	+	+	+
Time required for use	+	+	–	+	+	+	–
Availability of quantitative assessment of changes	–	–	–	–	–	–	–
Flexibility of the method	–	–	+	+	–	+	–
Use in rapid changes	–	+	+	–	–	–	–
Possibility of certification	+	+	–	+	+	–	–
Possibility of training	+	+	+	+	+	+	+
Availability of an informative website	+	+	+	–	+	–	–
Availability of free training	–	–	+	–	–	+	+
Availability of use in complex projects	+	+	+	–	+	–	–
Cost of training, $	2000	1790	–	–	6000 (850)	–	–

Source: [5]

Table 5.1 uses the following notations [5]:
– method No. 1 – Prosci ADKAR model;
– method No. 2 – AIM method;
– method No. 3 – Beckhard and Harris's model;
– method No. 4 – Bridges Transition model;
– method No. 5 – Kotter's 8-Step Change model;
– method No. 6 – Kübler-Ross model;
– method No. 7 – method based on the Kurt Lewin model.

The main drawback of the considered change management methods is their lack of opportunities for quantitative assessment of changes at the level of individual project tasks. Therefore, an analysis of modern research aimed at eliminating this drawback was conducted.

A significant amount of modern work in the field of change management in projects and, in particular, in IT projects is based on the idea of change management to the tasks of general (organizational) project management as a whole. For example, in [15] a new hybrid model of the IT project life cycle is proposed, which combines predictive planning with iterative implementation. This model emphasizes defined requirements, short sprints and early feedback, focusing on the interaction of people and customers, while limiting changes during the project life cycle [15]. But this model does not take into account the features of long-term IT projects. In addition, this model, like the change management methods discussed above, is focused on solving the problems of project personnel management, and not on quantitative assessment and change management. In [16] mathematical models and methods of change management in megaprojects caused by integrative actions of stakeholders under complex external conditions are investigated. The term "megaprojects" in [16] defines large-scale investment programs with complex organizational structures that unite many stakeholders, the interaction of which leads to the redistribution of power and the creation of temporary control centers. To describe the management of megaprojects, vector-matrix models of a dynamic system with feedback on the results of changes were used. To identify recurring patterns of negative events, the method of event-based analysis was used. As a result of the study, in [16] a prototype of IS was proposed, which is based on:

– a mathematical model of change management in megaprojects;

– a methodology of neural network analysis based on the use of the large language model Qwen 2.5-Plus for processing text information and forming quantitative estimates;

– a software interface for uploading documents, automated data processing and visualization of results.

This prototype provides users with the ability to analyze stakeholder interactions, assess the intensity of change, and predict potential risks based on historical data [16].

The disadvantage of this IS prototype is its focus on processing textual information, through which stakeholders can express their own attitude to the megaproject. This means that quantitative assessments of changes are formed in this prototype on the basis of individual statements, which can be influenced by the economic or political environment of stakeholders. In addition, this prototype is also focused on managing changes and risks of the entire megaproject as a whole. Solving the problem of project evaluation by using neural networks is one of the current research directions. Thus, in [17] it was proposed to use a neural network of radial basis functions to predict project efficiency, which illustrates changes in efficiency levels during the phases of failures and project recovery. To train this network in [17], data from 64 completed construction projects were used. The results show that the discrepancy between the predicted and actual values of the stability of the assessed project is less than 10%. But this approach has quite significant disadvantages. Among these shortcomings, it is necessary to highlight, in particular: the orientation of the proposed IT to the evaluation of the project as a whole; the need to train the neural network on a large volume of historical data; the impossibility of quantitatively assessing changes in individual project tasks.

A separate issue is the analysis of resistance to changes by project employees. Existing methods for assessing resistance are based on closed questionnaires and binary classifications. However, such methods, as indicated in [18], limit the expression of opinions and do not provide a nuanced segmentation of employees' positions on changes. Therefore, in [18] it is proposed to use an innovative automated methodology for analyzing resistance to changes by project employees, which combines specialized Large Language Models (LLM) with a zero result (in particular, DeBERTa-v3-large-zeroshot) and rapid engineering methods. However, the disadvantage of this methodology is that it, like existing methods, is based on employees' responses to pre-prepared questionnaires. Therefore, this issue requires additional scientific research that goes beyond the scope of this study.

In general, the limitations and shortcomings inherent in modern research in the field of change management in projects and, in particular, in IT projects, can be formulated as follows:

– the absence in the overwhelming majority of widespread applied change management methods of a tool for quantitatively assessing these changes;

– the orientation of the overwhelming majority of methods and IT change management precisely on managing project personnel and stakeholders, as well as their relationships with each other within the project life cycle;

– the use of neural networks for various purposes in the proposed modern IS and IT change management and assessment studies, which significantly complicates the design and implementation of these IS and IT;

– the use of large language models in IS and IT change management and assessment, which leads to the formation of change assessments based not on specific actions of project stakeholders, but on their statements on this issue.

Therefore, the main direction of work on automation of change management is the expansion of existing project management systems and, in particular, IT projects by developing separate analytical services. These services should be aimed at solving the problems of assessing changes in those phases, processes and activities of an IT project that are relevant for stakeholders in specific periods of time. A feature of these services should be their readiness for operation in the conditions of the so-called "cold start" (i.e., in the absence of an array of historical data or a small amount of such data for a specific project).

An example of research in this direction is the work [19], devoted to the analysis of the impact of changes together with differences in the code during the verification of the software code of an IT project. To carry out this analysis, in [19] it was proposed to combine methods of dependency analysis based on call graphs and methods of intelligent history analysis. Using this combination of methods made it possible to calculate a set of file metrics and an overall risk score for each change request. The obtained estimates were not very accurate, but their accuracy generally satisfied stakeholders and IT project personnel. In addition, the time of analytical calculations during the experimental verification of the obtained combination of methods ranged from 7.4 to 22.43 seconds [19]. This makes it possible to apply the corresponding IT to change management in the management systems of any IT projects based on almost any Agile or hybrid methodologies and frameworks.

Therefore, the purpose of this study is to develop a service for automated solution of the problem of quantitative assessment of changes for the long-term IT project management system. The operation of this service will reduce the costs of implementing long-term IT projects by reducing unplanned time costs for performing individual tasks of these projects.

To achieve this goal, the following tasks were solved in the study:

– to improve the method based on the Beckhard and Harris's model for quantitative assessment of changes in a long-term IT project;

– to develop IT elements of quantitative assessment of changes based on the improved method;

– to carry out experimental verification of the obtained research results.

5.3 Materials and methods

The object of the study is the change management process in an IT project. This process is not included in the standard processes of the IT product life cycle as a system [4, 20], but is recognized as a typical process of the current change management framework [3]. This process can be added to any process model of an IT project without significant changes in this model.

The main hypothesis of the study is the hypothesis of the possibility of improving the quality of assessment of changes in the execution time of individual tasks of long-term IT projects by developing and implementing a service that will allow automating the solution of the change assessment problem.

The developed service is based on a method based on the Beckhard and Harris's model [10].

This method consists of the following stages:

– "Internal organizational analysis";

– "Determination of the need for changes";

– "Analysis of differences between the current state and the desired one";

– "Action planning";

– "Transition management".

"Internal organizational analysis" stage. The purpose of this stage is to determine the general attitude towards change in the organization. At this stage, it is necessary to identify employees who may resist change. In particular, it is necessary to identify any external factors that may hinder the change process.

"Determination of the need for changes" stage. This stage is necessary in order to create a basis for implementing change. Key change participants must agree that change is necessary for the success of the organization. This agreement assumes that change participants can clearly articulate where they want to take the organization and why implementing change will help bring the organization closer to the desired state. It is also necessary to have an idea of the consequences associated with refusing to implement change [10].

"Analysis of the differences between the current state and the desired state" stage. Before implementing change, it is first necessary to determine what deviations exist between the current state of the organization and what it should be. Determining these deviations is important in order to clearly formulate an understanding of the future of the organization.

"Action planning" stage. A change plan is formed and submitted for implementation. It is necessary to identify the key participants in the change process and the responsibilities of everyone who makes the changes.

"Transition management" stage. After the change is implemented, the employees implementing the changes are responsible for continuously monitoring the progress of the changes and making adjustments.

The main disadvantages of this method are:

– the lack of the ability to quantitatively assess the changes that arise during the implementation of the IT project;

– the subjectivity of making decisions on change management, which depend on the individual competence of decision-makers during the implementation of the stages of the method under consideration.

To develop a service that will automate the solution of the problem of quantitatively assessing changes in long-term IT projects, it is proposed to apply a descriptive approach. This approach is based on the following concepts: SCRUM, sprint, task, descriptor, change indicator.

SCRUM is a project management methodology commonly used in software development. It is based on iterative and incremental approaches to product development, where the team works in short cycles (sprints) [21].

A sprint is a short period of time (usually two to four weeks) in which a project team works on a specific set of tasks. This practice is implemented when using the Agile software development methodology to increase the efficiency and transparency of the development process [21].

A task is a separate task with a unique identification number that is performed within a sprint by a specific project team to achieve a set goal or result. A task is most often evaluated by its complexity and execution time. It has the following characteristics: description, creation date, priority, estimated execution time, responsible executor, execution status, acceptance criteria, deadline, comments, and additional materials.

A descriptor is a quantitative characteristic of a task that provides descriptive information about the task in a numerical format. A descriptor can describe certain aspects of tasks, such as: information about the use of a certain technology, information about the team that will perform the task, etc.

Change indicator – a numerical indicator that can be used to assess changes when they are introduced. The change indicator can be the time required to complete a task, or the difference between the planned time and the time spent on completing the task.

The descriptor approach, which is proposed for assessing changes at the level of IT project tasks, includes the following processes:

– collecting and storing in a database information that will describe the characteristics of the tasks being performed in the form of descriptors;

– building statistical models to find the dependence of the change indicator on the descriptor in order to quantitatively assess the changes.

A condition for using the descriptor approach is the availability of sufficient information about the tasks being performed. For long-term IT projects (a distinctive feature of which is the characteristic accumulation of information on completed tasks in a significant amount, which allows this information to be analyzed using statistical methods), this condition is met.

5.4 Results of improving the method of quantitative assessment of changes in a long-term IT project

5.4.1 Classification of descriptors that describe tasks and changes in an IT project

To use the descriptor approach within the framework of the implementation of a long-term IT project, the following classification of descriptors is proposed:

– technical descriptors;
– task feature descriptors;
– team description descriptors;
– testing process descriptors;
– web page accessibility descriptors;
– other descriptors.

The number of types of descriptors by which the classification is carried out may vary depending on the requirements of the IT project.

A descriptor may have a binary value: "Yes" (or "1" or "True") or "No" (or "0" or "False"). For example, the descriptor of the change of the library for testing modules for a certain software component can have the value "1" (if the task requires changing the library for testing the component) or "0" (if not).

Examples of a descriptor with a simple numerical value can be a descriptor of the number of lines of code for which it is necessary to implement functional testing, or a descriptor of the number of errors in the code that must be fixed during the task.

When using the descriptor approach, it is recommended to provide the ability to enter descriptors into the database to all members of the project team.

It is proposed to carry out the process of updating the descriptor database within each sprint and add information about the presence of this process to the readiness requirements of tasks that exist in the project (definition of done, DoD) [22].

If there is a database of descriptors and change indicators, it is possible to build statistical models of the dependence of change indicators on descriptors.

It is proposed to build a unidimensional model, based on the following reasons:

– simplicity of calculations;

– the need for a smaller amount of data for model development;

– the absence of a problem associated with multicollinearity of the proposed descriptors [23].

A unidimensional model may be sufficient to identify the main trends and dependencies in the data, especially at the initial stages of the study.

5.4.2 Development of an improved method for quantitative assessment of changes

Using a descriptive approach, the basic method based on the Beckhard and Harris's model model [10] was adapted to change management at the level of individual IT project tasks. As a result of this adaptation, an improved method for quantitative assessment of changes was obtained, which is presented as a sequence of stages and individual activities performed within specific stages. The scheme of implementation of the stages of the method and their most important activities, taking into account the cyclical transitions between activities and stages, is shown in **Fig. 5.1**.

The stage "Internal project analysis" (Stage 1) is intended to form a description of a long-term IT project as a basis for further quantitative assessment of changes. This stage is proposed to be considered as a set of the following activities:

– assessment of current processes and change management systems at the task level;

– identification of employees who may have significant resistance to change;

– identification of external factors that may hinder the change process;

– analysis of resources available for implementing changes;

– determination, description and classification of descriptors;

– determination of a place to store descriptors (database);

– determination and description of change indicators and their storage location;

– appointment of those responsible for forming the descriptor database and its updating.

These activities within Stage 1 can be performed sequentially or in parallel.

The determination, description and classification of descriptors for current tasks should be performed at each iteration of the project.

The stage "Collection of information in the form of descriptors at each iteration of the project" (Stage 2) is necessary for the formation of a descriptor database. This stage consists of operations for collecting and storing descriptor values determined in the results of Stage 1. These operations can be implemented both by means of the existing IS for managing a long-term IT project and by means of the service being created.

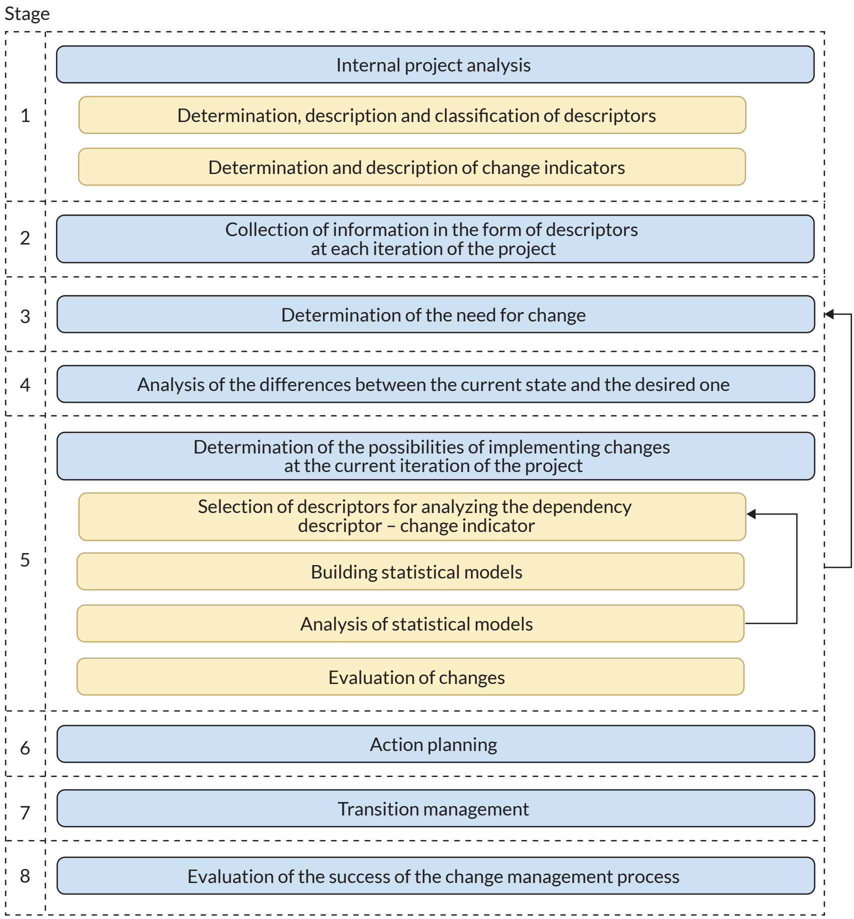

Fig. 5.1 Stages and key activities of the improved quantitative change assessment method

The stage "Determination of the need for changes" (Stage 3) is necessary to create a basis for implementing changes. At this stage, most of the work can be performed by a business analyst. It is proposed to consider this stage as a set of the following activities:

– identifying specific areas where changes are needed at the task level;

– assessing the needs of users and customers in making changes to tasks;

– setting goals and expected results from implementing changes.

The stage "Analysis of differences between the current state and the desired one" (Stage 4) is necessary to identify the impact of changes on the IT project. This stage is proposed to be considered as a set of the following activities:

– identification of a set of significant deviations that exist between the current state of the analyzed task and its state after making changes;

– assessment of possible risks and the impact of changes on the project;

– comparison of technical characteristics of the current and desired state of the project.

The stage "Determination of the possibilities of implementing changes in the current iteration of project execution" (Stage 5) is necessary for quantitative assessment of the changes identified in Stage 3 and analyzed in Stage 4. At this stage, most of the work should be performed using the tools of the created service. This stage is proposed to be considered as a set of the following activities:

– selection of descriptors for analyzing the dependence of the change indicator on the descriptor;

– construction of statistical models;

– analysis of statistical models;

– assessment of changes at the task level.

A set of descriptors can be analyzed by the mean value of the descriptors, but this analysis can have significant drawbacks. Analyzing data using the mean (average) is a simple way to describe the central tendency of the data. However, this approach may not be predictive in cases where the data have high variability or when they are unevenly distributed. Especially in cases where the data have a large number of outliers or asymmetry, using the mean can lead to incorrect or incomplete understanding of the data.

Taking into account the features of the descriptor approach, within the framework of Stage 5, it is proposed to investigate the relationship between the indicators of change (the desired indicators) and the values of the descriptors. This relationship can be represented by the formula

$$y_i = \beta_0 + \beta_1 \cdot x_i, \tag{5.1}$$

where y_i – the dependent variable (change indicator); x_i – the independent variable (descriptor); β_0, β_1 – regression parameters.

After that, the task of finding the values of the regression parameters is solved.

There are many methods for building regression models. However, the most commonly used method is the Ordinary Least Squares (OLS).

Before using the OLS method, it is necessary to check the fulfillment of the prerequisites for using regression analysis. As a result of such analysis, outliers may be detected that will need to be processed.

In the case of outliers in the data collected during the implementation of the project tasks, the developed combined method proposes to use Huber regression, because it has properties related to robustness [24].

After building statistical models, it is proposed to check their performance using the following indicators [24]:

– coefficient of determination (R^2);

– forecasting coefficient (Q^2);

– forecasting coefficient calculated using the Leave-one-out cross-validation procedure (Q^2_{LOOCV});

– standard deviation (σ);

– assessment of the significance of the statistical model (F-criterion).

If, during the verification of statistical models, these indicators acquire unsatisfactory values, another descriptor is selected and the statistical models are recalculated.

If the values of these indicators are satisfactory, the changes are assessed using the constructed model.

When using the obtained assessment, an assumption is made about the possibility of implementing changes at the task level in the current iteration of the IT project. If the assessment of the required task completion time corresponds to this possibility, the changes are approved. If the assessment does not correspond to the possibility of implementing changes, the results of Stage 3 are analyzed, after which, if necessary, the possibilities of either attracting additional human resources from other teams or overtime work of the current team members are analyzed. If it is determined that the changes cannot be implemented in the current project iteration, a new task is created. This task is included in the list of tasks when planning subsequent project iterations.

The stage "Action Planning" (Stage 6) is necessary for carrying out operations to re-plan the IT project taking into account the introduced and evaluated changes and implementing this plan. It is proposed to consider this stage as a set of the following activities:

– identification of key participants in the change process and responsibilities of each person making the changes;

– development of a specific change implementation plan, including resources and deadlines;

– determination of the sequence of steps for implementing the changes;

– preparation of a communications plan to inform customers about the changes and their impact on the project.

The stage "Transition Management" (Stage 7) is necessary for operational management of work on the direct implementation of the IT project iteration. It is proposed to consider this stage as a set of the following activities:

– implementation of the change plan for the task;

– monitoring the impact of changes on the project and timely identification of problems;

– providing support for users during the transition to a new state of the project.

The stage "Evaluation of the success of the change management process" (Stage 8) is necessary to analyze the progress of the IT project iterations and assess the success of the planned changes to the IT project. This stage is recommended to be carried out once every several (two or more) iterations of the project or at the request of the project stakeholders.

In order to assess the success of the change management process, it was proposed to use the following indicators at Stage 8 [5]:

– the share (in percent) Ch_{Cp} of time changes canceled due to the impossibility of their implementation, excluding changes that became unnecessary for a certain period;

– the share (in percent) of time changes that were accepted by the customer and approved as successfully implemented for a certain period of time, Ch_{Cp};

– the difference between the time spent on work and the estimated time for a certain period of time, δCh_T;

– the share (in percent) of changes that were completed on time for a certain period of time, Ch_{iTp}.

The share (in percent) of changes canceled over a certain period of time due to the impossibility of their implementation Ch_{Cp}, excluding changes that have become irrelevant, can be calculated by the formula [5]

$$Ch_{Cp} = \frac{Ch_C}{Ch} \cdot 100\%, \tag{5.2}$$

where Ch_C – the number of canceled changes for a certain period of time due to the impossibility of their implementation, excluding changes that have become irrelevant; Ch – the total number of changes for a certain period of time.

Large values Ch_{Cp} indicate poorly planned changes.

The share (in percent) of changes that were accepted by the customer and approved as successfully implemented for a certain period of time Ch_{Sp} can be calculated by the formula [5]

$$Ch_{Sp} = \frac{Ch_S}{Ch} \cdot 100\%, \tag{5.3}$$

where Ch_S – the number of changes that were accepted by the customer and approved as successfully implemented for a certain period of time.

A large value Ch_{Sp} indicates a better change management process.

The difference between the time spent on performing tasks of a long-term IT project and the estimated time for a certain period of time δCh_T can be calculated by the formula [5]

$$\delta Ch_T = \frac{\sum_{i=0} (t_i^{(p)} - t_i^{(a)})}{t \cdot i}, \tag{5.4}$$

where $t_i^{(p)}$ – the time planned for implementing the change, days; $t_i^{(a)}$ – the time spent on implementing the change, days; t – the time period for evaluation, days; i – the number of changes for the time period t.

The indicator δCh_T indicates whether changes are performed on time and in accordance with the change plan. The lower the indicator, the better organized the change management.

The share (in percent) of changes that were completed on time for a certain period of time Ch_{iTp} can be calculated by the formula [5]

$$Ch_{iTp} = \frac{Ch_{iT}}{Ch} \cdot 100\%, \tag{5.5}$$

where Ch_{iT} – the number of changes that were completed on time for a certain period of time.

A high value Ch_{iTp} indicates a better change management process and adherence to the planned schedule.

The use of the proposed improved method for solving the problem of quantitative assessment of changes made it possible to formulate the main requirements for the service that should provide an automated solution to this problem. To ensure the possibility of multiple use of the obtained solutions, this service during design and implementation (design & development) is proposed to be considered as a separate IT for automated solution of the problem of quantitative assessment of changes.

5.5 Elements of information technology for automated solution of the problem of quantitative assessment of changes

IT for automated solution of the problem of quantitative assessment of changes (hereinafter referred to as IT for quantitative assessment of changes) is proposed to be developed for automated implementation of the proposed improved method. Therefore, it was decided to use this IT to automate only those stages of the improved method that are directly related to the collection, processing and storage of data and information on changes that arise during the implementation of a long-term IT project.

These stages include [5]:

– Stage 1 "Internal project analysis";

– Stage 4 "Analysis of differences between the current state and the desired one";

– Stage 7 "Transition management".

Based on this decision, it was proposed to present the developed IT as a sequence of the following stages and steps [5].

Stage 1. Survey of the team of performers on the current perception of the change management process.

Step 1.1. Forming a questionnaire and conducting a survey of all employees who form the IT project teams regarding their current attitude to the change management process.

Step 1.2. Processing the survey results and forming current assessments of the level of employee satisfaction with the change management process.

Stage 2. Forming and storing descriptors of individual IT project work.

Step 2.1. Determining the set of descriptors of individual IT project work.

Step 2.2. Forming sets of values of the defined descriptors.

Step 2.3. Storing the formed sets of values of the defined descriptors.

Stage 3. Statistical analysis of IT project descriptors.

Step 3.1. Determining indicators of IT project changes.

Step 3.2. Selecting the defined descriptors to create a change model.

Step 3.3. Analysis of the sets of values of the selected descriptors for the presence of outliers.

Step 3.4. If the analysis results obtained as a result of Step 3.3 indicate the absence of outliers, then build a change model using OLS. Otherwise, build a change model using the Huber regression method.

Step 3.5. Calculate the performance indicators of the built change model.

Stage 4. Survey of the team of performers on the final perception of the change management process.

Step 4.1. Form a questionnaire and conduct a survey of all employees who form the teams of performers of the IT project on the final attitude to the change management process.

Step 4.2. Process the survey results and form final assessments of the level of employee satisfaction with the change management process.

A description of the IT architecture of quantitative change assessment in the form of a data flow diagram is given in **Fig. 5.2** [5]. When creating this diagram, the Yordon-DeMarco notation was used. **Fig. 5.2** does not indicate the names of the flows that are directly related to the data warehouses, because these names coincide with the names of the data warehouses themselves.

For the further implementation of IT, the features of its technology stack were determined. The technology stack will be understood here and now as a set of technologies that are used together to develop and support software [25].

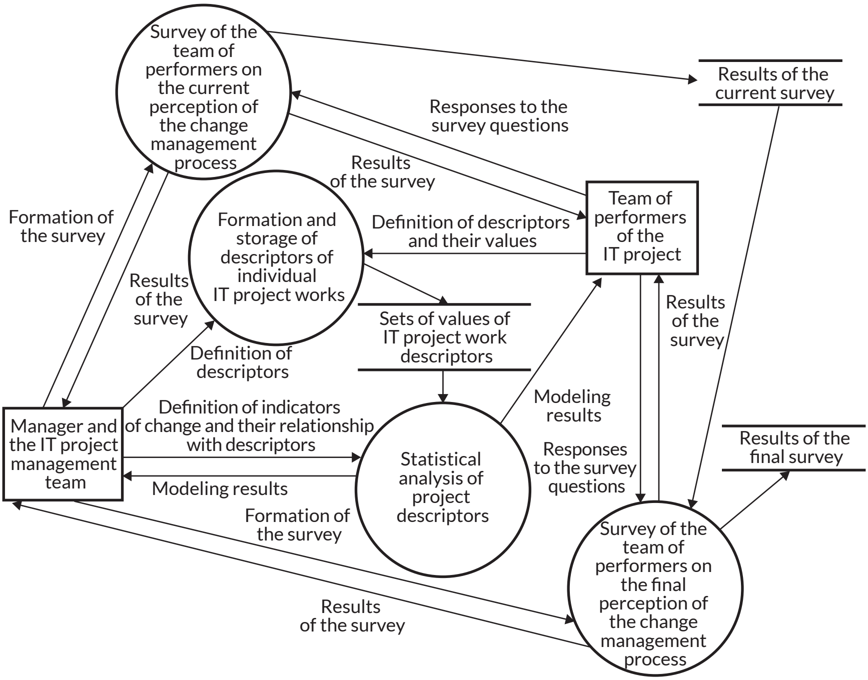

Fig. 5.2 Description of the architecture of the information technology of quantitative change assessment
Source: [5]

The technological stack of IT quantitative change assessment has the following components:

– for the development of the elements "Survey of the team of performers on the current perception of the change management process" and "Survey of the team of performers on the final perception of the change management process", it is proposed to use the Google Forms service, which is included in the free Google Docs editor package from Google;

– for the development of the element "Formation and storage of descriptors of individual IT project works", it is proposed to use existing IT project management systems (for example, Jira [26]) and tools for storing descriptor descriptions and their numerical values (for example, Microsoft Excel, or its analogues, or database management systems Firebase, MongoDB, PostgressSQL, etc.);

– for the development of the element "Statistical analysis of IT project descriptors", it was proposed to develop a specialized service using the Python programming language.

Microsoft Excel was used to implement data warehouses where it was planned to store descriptor descriptions and their numerical values.

A fragment of the program code for calculating the values of the Gaussian density function is shown in **Fig. 5.3**. A fragment of the program code for plotting the calculated Gaussian density function and checking the results obtained using the three-sigma method is shown in **Fig. 5.4** [5].

A fragment of the program code for creating a model using the OLS method is shown in **Fig. 5.5**. A fragment of the program code for creating a model using the Huber regression method is shown in **Fig. 5.6** [5].

A fragment of the program code for calculating the performance indicator of the model "Coefficient of determination" is shown in **Fig. 5.7**. A fragment of the program code for calculating the performance indicator of the model "Forecasting coefficient" is shown in **Fig. 5.8**. A fragment of the program code for calculating the performance indicator of the model "Forecasting coefficient Q^2_{LOOCV}" is shown in **Fig. 5.9** [5].

A fragment of the program code for calculating the performance indicator of the model "Standard deviation (σ)" is shown in **Fig. 5.10**. A fragment of the program code for calculating the performance indicator of the model "*F*-criterion" is shown in **Fig. 5.11** [5].

```
// Функція для обчислення значень гауссівської функції щільності (PDF)
function gaussianPDF(x, mean, variance) {
  const stdDev = Math.sqrt(variance);
  return (1 / (stdDev * Math.sqrt(2 * Math.PI))) * Math.exp(-((x - mean) ** 2) / (2 * variance));
}
```

Fig. 5.3 A fragment of the program code for calculating the values of the Gaussian density function
Source: [5]

```
// Функція для побудови графіку розподілу Гаусса у вигляді гістограми та перевірки методом трьох сигм
export function plotGaussianDistribution(x, chartRef2) {
  // Підготовка даних для побудови
  const mean = math.mean(x); // Середнє значення x
  const variance = math.variance(x); // Дисперсія x
  const stdDev = Math.sqrt(variance); // Стандартне відхилення

  // Обчислення значень гауссівської функції для кожного x
  const data = x.map(value => ({
    x: value,
    y: gaussianPDF(value, mean, variance)
  }));

  // Вивід корисної інформації в консоль
  console.log(`Середнє значення (Mean): ${mean}`);
  console.log(`Дисперсія (Variance): ${variance}`);
  console.log(`Стандартне відхилення (Standard Deviation): ${stdDev}`);

  // Перевірка за методом трьох сигм
  const lowerBound = mean - 3 * stdDev;
  const upperBound = mean + 3 * stdDev;
  const outliers = x.filter(value => value < lowerBound || value > upperBound);

  console.log(`Аномальні значення (за межами 3σ): ${outliers}`);

  new Chart(chartRef2.current.getContext('2d'), {
    type: 'bar', // Використання гістограми
    data: {
      labels: x,
      datasets: [{
        label: 'Гауссівський розподіл',
        data: data.map(point => point.y),
        borderColor: 'rgba(75, 192, 192, 1)',
        backgroundColor: 'rgba(75, 192, 192, 0.2)',
        borderWidth: 1
      }]
    },
    options: {
      scales: {
        x: {
          type: 'linear',
          position: 'bottom',
          title: {
            display: true,
            text: 'Значення X'
          }
        },
        y: {
          type: 'linear',
          position: 'left',
          title: {
            display: true,
            text: 'Густина ймовірності (Probability Density Function)'
          }
        }
      },
      plugins: {
        annotation: {
          annotations: {
            box1: {
              type: 'box',
              xMin: lowerBound,
              xMax: upperBound,
              yMin: 0,
              yMax: Math.max(...data.map(point => point.y)),
              backgroundColor: 'rgba(0, 255, 0, 0.1)',
              borderColor: 'rgba(0, 255, 0, 0.5)',
              borderWidth: 1,
              label: {
                content: '3σ Range',
                enabled: true,
                position: 'center'
              }
            }
          }
        }
      }
    }
  });
}
```

Fig. 5.4 A fragment of the program code for plotting the calculated Gaussian density function and checking the results obtained using the three-sigma method
Source: [5]

```
// Простий метод МНК (метод найменших квадратів)
export function leastSquaresRegression(x, y) {
    const n = x.length;
    const X = math.concat(math.reshape(x, [n, 1]), math.ones([n, 1]), 1);
    const theta = math.multiply(math.inv(math.multiply(math.transpose(X), X)), math.multiply(math.transpose(X), y));
    return theta;
}
```

Fig. 5.5 A fragment of the program code for creating a model using the least squares method
Source: [5]

```
  while (iteration < maxIterations) {
    let updatedTheta = [0, 0];
    let totalLoss = 0;

    for (let i = 0; i < n; i++) {
      const r = y[i] - (thetaHR[0] * x[i] + thetaHR[1]);
      const loss = huberLoss(r, delta);
      const weight = Math.sqrt(delta / (r * r + delta));

      // Construct weighted matrices
      const weightedX = [X[i][0], X[i][1]]; // Select columns [0, 1] (corresponding to x and the intercept)
      const weightedY = yMatrix[i][0]; // Access element from yMatrix directly using array indexing

      // Update coefficients using weighted least squares
      updatedTheta = math.add(updatedTheta, math.multiply(weight, weightedX, weightedY));

      totalLoss += loss;
    }

    // Перевірка на збіжність за зміною загальних втрат
    if (Math.abs(totalLoss - previousLoss) < tolerance) {
      break; // Зупиняємо ітерації, якщо втрати майже не змінюються
    }

    // Оновлення коефіцієнтів регресії на основі вагованих МНК
    thetaHR = updatedTheta;
    previousLoss = totalLoss; // Оновлюємо попередні втрати для порівняння
    iteration++;
  }

  return thetaHR;
}
```

Fig. 5.6 A fragment of the program code for creating a model using the Huber regression method
Source: [5]

```
function calculateR2(actual, predicted) {
  const meanActual = math.mean(actual);
  const totalSumOfSquares = math.sum(actual.map(val => Math.pow(val - meanActual, 2)));
  const residualSumOfSquares = math.sum(predicted.map((val, i) => Math.pow(actual[i] - val, 2)));
  const r2 = 1 - (residualSumOfSquares / totalSumOfSquares);

  return r2;
}
```

Fig. 5.7 A fragment of the program code for calculating the performance indicator of the model "Determination coefficient R^2"
Source: [5]

```
// Розрахунок простого Q^2 для регресії
function calculateSimpleQ2(y, yPred) {
  const meanY = math.mean(y);
  const totalSumOfSquares = math.sum(y.map(val => Math.pow(val - meanY, 2)));
  const sumSquaredErrors = math.sum(y.map((val, i) => Math.pow(val - yPred[i], 2)));
  const simpleQ2 = 1 - (sumSquaredErrors / totalSumOfSquares);

  return simpleQ2;
}
```

Fig. 5.8 A fragment of the program code for calculating the performance indicator of the model "Forecasting coefficient Q^2"
Source: [5]

```
// Розрахунок Q^2 за LOOCV для регресії
function calculateQ2LOOCV(x, y, regressionFunction, addParam) {
  const n = x.length;
  let sumSquaredErrors = 0;
  let totalSumOfSquares = 0;

  for (let i = 0; i < n; i++) {
    const xTrain = [...x.slice(0, i), ...x.slice(i + 1)];
    const yTrain = [...y.slice(0, i), ...y.slice(i + 1)];
    const xTest = x[i];
    const yTest = y[i];

    const theta = regressionFunction(xTrain, yTrain, addParam);

    // Перевірка, чи отримано коректні значення коефіцієнтів theta
    if (theta !== null && typeof theta === 'object' && theta.length >= 2) {
      const yPred = theta[0] * xTest + theta[1];
      const squaredError = Math.pow(yTest - yPred, 2);
      sumSquaredErrors += squaredError;

      const meanY = math.mean(yTrain);
      const totalSquare = Math.pow(yTest - meanY, 2);
      totalSumOfSquares += totalSquare;
    } else {
      console.error('Regression function did not return valid coefficients.');
      // Опціонально можна повернути null або інше значення у випадку помилки
      return null;
    }
  }

  const Q2 = 1 - (sumSquaredErrors / totalSumOfSquares);
  return Q2;
}
```

Fig. 5.9 A fragment of the program code for calculating the performance indicator of the model "Forecasting coefficient Q^2_{LOOCV}"
Source: [5]

```
// StandardDeviation
function calculateCalculatedStandardDeviation(actual, calculated, numParams) {
  const n = actual.length;
  const residuals = actual.map((val, i) => val - calculated[i]);
  const squaredResiduals = residuals.map(residual => Math.pow(residual, 2));
  const sumSquaredResiduals = math.sum(squaredResiduals);

  const sigmaCalc = Math.sqrt(sumSquaredResiduals / (n - numParams - 1));
  return sigmaCalc;
}
```

Fig. 5.10 A fragment of the program code for calculating the performance indicator of the model "Standard deviation (σ)"
Source: [5]

```
function calculateFStatistic(R2, numParams, numDataPoints) {
  const F = (R2 / (1 - R2)) * ((numDataPoints - numParams - 1) / numParams);

  return F;
}
```

Fig. 5.11 A fragment of the program code for calculating the performance indicator of the model "F-criterion"
Source: [5]

5.6 Description of an example of solving the problem of quantitative assessment of changes during the implementation of a long-term IT project

5.6.1 Description of the features of the long-term IT project "Web Constructor"

Experimental verification of the obtained research results was proposed to be carried out during the implementation of the long-term "Web Constructor" IT project. The duration of the implementation of this IT project is 6 years.

The main result of this IT project is the "Construct" system. This system implements an administrative portal, where customers can use the library of components to create sites. The "Construct" system supports more than 20 languages and allows to create web pages for most countries of the world. The results obtained from the implementation of the long-term "Web Constructor" IT project are already in operation and continue to expand.

The main goal of the long-term "Web Constructor" IT project is the development and support of the created "Construct" system.

According to the classification of the main groups of projects by interests of the Project Management Institute (PMI), this project belongs to IS development projects [2, 3].

Table 5.2 provides an additional classification of the "Web Constructor" project. This classification was carried out according to classification criteria according to the materials of sources [27, 28].

Table 5.2 Additional classification of the "Web Constructor" project

Classification feature	Project type
By scale	Large
By complexity	Technically complex
By implementation time	Megaproject
By resource constraints of a set of projects	Program
By the nature of the project and the level of participants	International
By the nature of the project's target task	Marketing
By the main reason for the project	Need for structural and functional transformations
By the location of the customer	External
Degree of customer participation in the project	Average

The "Web Constructor" project is divided into subprojects depending on the requirements. For example, during the project implementation, subprojects took place to update designs, transition to more progressive and competitive technologies, add new functionality, and others. The project continues to scale, so new teams and subprojects are currently being created to perform individual business tasks.

During the implementation of the "Web Constructor" IT project and its individual subprojects, a lot of work is performed, the main of which are:

– creation of new system components;
– updating existing system components;
– creation of new web page templates;
– improving the performance of existing web pages;
– collection of analytical information on system usage;
– system support/updating;
– solving problems with content accessibility;
– transition to new technologies;
– implementation of the possibility of using the system in new countries;
– updating the system design.

To implement the "Web Constructor" IT project, the resources of the following IT company departments were used:

– Management department;
– Inclusion and accessibility department;
– Business analysis department (BA);
– Strategic architecture and consulting department (Solution Architects, SA);
– Back end (BE) development department;
– Front end (FE) development department;
– System efficiency and performance department;
– Software development and implementation automation and optimization department (DevOps);
– User interface design department;
– Software testing experts department (Quality Assurance, QA).

Several teams participate in the implementation of the IT project. Most teams are development teams, each of which has 6–10 employees. Such a team usually includes several BE developers, FE developers, and QA specialists. One of the project participants plays the leading role of the team leader and reports to other teams and divisions of the IT company on the success of the work, maintains communication with them.

The "Web Constructor" IT project is international and includes employees from different countries of the world.

When planning the work of the teams of the "Web Constructor" IT project, the provisions of the Agile SCRUM methodology were used [21]. The main type of iteration of this IT project is a sprint lasting 3 weeks.

When planning a specific sprint of the "Web Constructor" IT project, the tasks that are allocated in the backlog of this sprint can be attributed to one of the following classes:

– tasks that must be fully completed during the current sprint;
– tasks that must be completed during the next sprint;
– tasks that will be performed during several sprints.

If the task is selected with the intention of obtaining a finished version of the program (i.e. as a release task), the team should spend no more than 2 weeks on its implementation, because it must be checked by test engineers and customers during acceptance testing (UAT) in the software product environment. If the task is planned without taking into account obtaining a finished version of the program (i.e. as a non-release task), development can take place within the entire sprint. Therefore, testing the task will be planned within the next sprint.

The main technologies used in the implementation of the "Web Constructor" IT project and forming its technological stack are: JS, React, JAVA, SQL, HTML, CSS, Redux, TS, LESS, AWS.

5.6.2 Overview of the causes of changes in the long-term "Web Constructor" IT project

The risks and changes that occur during the implementation of the "Web Constructor" project are closely related to its features and classification. Therefore, before starting the experimental verification of the obtained research results, it is necessary to consider the current problems that may lead to changes in the "Web Constructor" IT project and the "Construct" system.

The most relevant problem encountered during the implementation of the long-term "Web Constructor" IT project is the problem of changing the characteristics of tasks during their execution. Changing these characteristics in the process of developing a project task after assessing the time required for its execution is a very common phenomenon [29].

The characteristic of the task is a description of what must be performed during the implementation of the task. The characteristics are described and divided by BA specialists into characteristics on the BE side and characteristics of the FE. Sometimes the same characteristic may require work from two parts (BE and FE).

In this case, for the requirements descriptors, it is proposed to include such a requirement in both descriptors.

When performing the "Web Constructor" IT project, the characteristics of the tasks are described in a table format and each of them is given a universal identifier. If the task is technical in nature, then the characteristics can be explained and described by the system architect. At the time of discussing the task before selecting it for execution during the work sprint, the characteristics should already be described and the team should familiarize themselves with them in advance. This provides an opportunity to resolve misunderstandings about the task before its final discussion (grooming, SCRUM stage) [21].

Depending on the characteristics, the tasks are divided into technical and business tasks. Within the framework of business tasks, customer requirements are fulfilled, technical tasks are solved in order to improve the technical performance and quality of the created product.

Another group of relevant problems of changes in the long-term "Web Constructor" IT project are problems that arise during the project implementation.

Firstly, such problems include restrictions on direct communication between individual departments. As a result of such restrictions, communications become more complicated and sometimes take place as a dialogue in the form of electronic correspondence. Such complications, in turn, can slow down the editing of elements of the Construct system and lead to changes that should be made already at the time of work on the relevant tasks.

Secondly, such problems include situations when the design department initiates changes in designs during the implementation of a certain functionality, that is, when the task has already passed the stage of evaluation by the development team. Examples of the most frequent changes that arise due to this reason include the following: replacing images, changing colors, changing fonts, changing sizes, introducing new designs for individual cases or screen sizes, etc.

Thirdly, such problems include situations in which designers do not always inform BA specialists about changes to graphic designs after making them. Direct communication between the development team and designers in the "Web Constructor" IT project is absent and occurs through communication with BA specialists. Such changes can make the characteristics described by BA specialists irrelevant and create the need to make changes at the task level [26].

The problems considered generate a set of changes that arise directly during the implementation of the IT project. If such changes are insignificant and the team has the opportunity to perform them within the task it is working on, these changes are made immediately. If the changes are critical and complex, they are made as a separate task and given a higher priority. A task of this type is called a change request (Change Request, CR) [30]. In this case, another task planned within the current sprint will be replaced by this CR, if the team has the opportunity to complete it, or the requirements for the tasks in progress are simplified.

Another group of urgent problems of changes in the long-term "Web Constructor" IT project is associated with the need for interaction of individual subprojects with external organizations or internal teams. The results of the activities of these organizations and teams can significantly affect the progress of these subprojects and the entire IT project. For example, during the development of the server part, the JAVA programming language and the Adobe Experience Manager (AEM) platform are used. But the use of this platform in some cases, unfortunately, leads to difficulties and limitations in expanding the functionality. This leads to the fact that some tasks cannot be completed on time. In some cases, such tasks are postponed and a etter is written to the AEM developers with a request to fix the problems found.

Similar problems also arise during the development of the user interface of the "Construct" system. There is a separate internal subproject that provides a common library with components to be used at the "Web Constructor" IT project level. This leads to problems integrating individual system components with the library being created.

Some similar characteristics of tasks in parallel subprojects are implemented differently, which also leads to problems and complicates the integration process with related components and external systems.

Similar problems arise when using external libraries (for example, node package manager). There have been situations when updating the library version led to significant system defects and required replacing the used library or making a decision to use an old version of the current library. For such a decision, it was necessary to take the risk that refusing updates could cause system security problems. In such cases, it is possible to decide to wait for updates to the necessary library, because the development team of this library may know about the existing problem and plan to solve it in future releases.

A separate group of problems arises in the process of solving tasks of estimating the time required to perform a certain task and implement changes at the task level. There are cases when the team cannot estimate the time required to perform a task or implement changes due to insufficient information on the technical features. In such cases, it is proposed to perform the task for the purpose of more detailed research and estimation of the time required to perform the task or implement the change. Such a task is called a proof of concept (POC). However, the team may not always be aware of the risks and propose conducting such a study. In such cases, during development, it becomes necessary to make changes to the characteristics or to calculate the amount of time required to perform the task.

Estimation of the time required to perform tasks in different teams of the "Web Constructor" IT project occurs differently. In development teams, time estimation is carried out by planning using Poker Planning technology. Scores are given in the range from 1 to 13 by all team members. The meeting at which the assessment process takes place usually takes place a week after the meeting to discuss the task with the BA specialists. This, in turn, leads to the fact that the assessments may not be accurate and relevant, because the team may not remember all aspects of the assessed task.

Among other problems that are the causes of changes in the long-term "Web Constructor" IT project, the following types of problems should be noted:

a) problems arising from the work of the project teams under stressful conditions;

b) problems arising from the need to compensate for work;

c) problems arising from the complexity of the existing organizational structure of the IT company;

d) problems arising from the low level of experience of individual IT company specialists.

Type of problem (a) is caused by the fact that due to high competition, IT project tasks must be performed quickly and efficiently. This requirement sometimes leads to overtime work by employees or work on weekends. Frequent cases of such work can lead to emotional fatigue of employees and worsen the effectiveness of the team and individual project participants.

Type of problem (b) is caused by the international crisis of the IT market. The consequence of this crisis is a situation in which customers prefer budget solutions and refuse to implement the latest technologies due to financial resources. This leads to increased competition in the market and a decrease in the technological level of the created "Construct" system.

Type of problem (c) is caused by the incomplete knowledge and skills of members of the project teams regarding the rights and obligations of managers of individual divisions of the IT company. Team members are not personally familiar with higher-level managers and may not know which link in the organizational structure to contact with various issues. This can lead to the problem of a slow response when an employee contacts its manager, and the manager is forced to go further through the links of the organizational structure of the IT company's management.

A type of problem (d) is caused by the low level of experience of a certain group of employees in working with parts of the developed system. This is due to the long duration of the implementation of the IT project "Web Constructor" – a significant part of the system components was implemented by employees, some of whom are no longer working on the project. Among the problems of this group, one can single out the problem of poor-quality process of transferring information about the project and poor-quality process of adapting employees of the IT company.

According to the results of the review of the problems that occur during the implementation of the "Web Constructor" IT project, it was proposed to classify these problems for experimental verification of the obtained research results as follows:

– problems of quality of estimating the time required to complete tasks and make changes;

– problems of medium-term forecasting;

– problems of changing task characteristics;

– problems of an emotional nature;

– problems of the level of compensation;

– problems of the need for overtime work;

– problems of a technical nature;

– problems of communication;

– problems of the level of knowledge of employees;

– problems of organizational structure complexity;

– problems caused by the use of outdated documentation and the lack of a quality adaptation process for employees performing the project.

5.6.3 Description of the progress and results of experimental verification of the obtained results

It was decided to conduct an experimental verification of the obtained research results within the framework of the tasks and iterations of the "Web Constructor" IT project by one development team (the description of this team is given in **Subsection 5.6.1**). The IT project manager was appointed responsible for the implementation of the stages of the improved method.

Before the implementation of the improved method, the method based on the Beckhard and Harris's model was used during the project implementation [10].

During the implementation of Stage 1 of the improved method, the manager conducted a survey of the team regarding the perception of the change management process operating in the team at the task level. An anonymous questionnaire (Google Forms) was created for the survey and sent to each project participant. The questions and answers of the project participants are given in **Table 5.3**.

Table 5.3 Assessment of employees' attitude to the method based on the Beckhard and Harris's model

Questions for evaluation on a scale from 0 to 10	Employee								Average score per question
	1	2	3	4	5	6	7	8	
Speed of change implementation (10 – fast)	6	5	4	6	7	4	5	3	5
Comfort of change management processes (10 – comfortable)	5	4	5	6	7	6	8	7	6
Presence of stressful situations during change implementation (10 – absent)	7	8	6	7	7	6	8	6	6.875
Need for overtime work (10 – absent)	8	9	8	10	9	10	9	9	9
Difficulty of change assessment (10 – not difficult)	6	7	5	6	7	8	5	4	6
Average employee rating	6.4	6.6	5.6	7	7.4	6.8	7	5.8	6.575

According to the survey results, the average score of the current change management method was 6.575.

Then, with the participation of all project team members, an analysis of the use of descriptors was conducted.

A conditional name was chosen for each descriptor for use in the descriptor databases. The main descriptors at the time of discussion were the following:

– number of task characteristics (C_Amount);

– number of task characteristics to be implemented on the client side (C_FE_Amount);

– number of task characteristics to be implemented on the server side (C_BE_Amount);

– planned number of days to complete the task (W_Days_Planned);

– number of days spent to complete the task (W_Days_General);

– number of task characteristics at the time of completion of work on it (C_Amount_Final);

– difference between the number of task characteristics at the time of taking the task into work and the number of characteristics that occurs at the time of completion of work on the task (C_Delta);

– the difference between the number of days planned for the task at the time of its initiation and the number of days spent on its implementation at the time of its completion (W_Days_Delta).

During Stage 2, information was collected in the form of descriptors by entering them into the descriptor database.

The tasks that were performed during the last five sprints were analyzed. The description of the descriptors for these tasks is given in **Table 5.4**.

Table 5.4 Descriptors for tasks performed during the last five sprints

Task ID	C_Amount, pieces	C_Amount_ Final, pieces	C_FE_ Amount, pieces	C_BE_ Amount, pieces	W_Days-Planned, days	W_Days-General, days
WEBCO_2342	80	92	30	50	15	20
WEBCO_2343	80	94	65	15	23	31
WEBCO_2344	20	25	10	10	8	10
WEBCO_2345	35	38	15	20	7	8
WEBCO_2346	42	47	12	30	12	15
WEBCO_2347	76	85	26	50	9	14
WEBCO_2348	12	12	12	0	5	5
WEBCO_2349	56	61	26	30	16	18
WEBCO_2350	15	20	10	5	5	7
WEBCO_2351	71	75	25	46	14	16
WEBCO_2352	12	12	0	12	3	3
WEBCO_2353	7	10	7	0	3	4
WEBCO_2354	2	2	2	0	1	1
WEBCO_2355	1	8	1	0	2	6
WEBCO_2356	35	40	32	3	18	20
WEBCO_2357	8	10	1	7	4	5
WEBCO_2358	54	65	53	1	50	55
WEBCO_2359	34	53	25	9	74	84
WEBCO_2375	50	43	30	20	15	12
WEBCO_2376	40	33	25	15	12	9
WEBCO_2377	35	28	20	15	10	8
WEBCO_2380	20	15	12	8	7	5
WEBCO_2384	70	62	40	30	20	16
WEBCO_2385	50	42	28	22	15	12
WEBCO_2386	45	32	25	20	13	8
WEBCO_2387	40	30	22	18	12	8
WEBCO_2388	35	32	20	15	11	10
WEBCO_2342	80	92	30	50	15	20

During the next (sixth) sprint, descriptors were also entered into the descriptor database. The values of these descriptors are shown in **Table 5.5**.

Table 5.5 Descriptors for tasks performed during the sixth sprint

Task ID	C_Amount, pieces	C_Amount_ Final, pieces	C_FE_ Amount, pieces	C_BE_ Amount, pieces	W_Days-Planned, days	W_Days-General, days
WEBCO_2389	30	26	18	12	10	9
WEBCO_2395	20	20	12	8	8	8
WEBCO_2396	25	21	15	10	9	7
WEBCO_2397	30	35	18	12	10	13
WEBCO_2398	22	24	14	8	8	9
WEBCO_2399	35	35	20	15	11	11
WEBCO_2360	65	75	40	25	20	25
WEBCO_2361	25	59	15	10	8	23

During Stage 3 of the sixth sprint, when performing task WEBCO_2360, it became necessary to make changes at the task level. The task included updating the designs of the bottom menu component of the page (footer). During the task, it was discovered that changes to the graphical interface designs were necessary. The designer updated the graphical interface, and the business analyst updated the component characteristics in accordance with the design updates.

During Stage 4, the difference between the task states without changes and with changes was studied in detail. The team was explained the changes necessary for implementation.

The development team requested Stage 5 of the improved method. To build statistical models of changes, a descriptor was selected that determines the difference between the number of task characteristics at the time the task was started and the number of characteristics that occur at the time the task was completed (C_Delta). The change indicator was chosen as the difference between the number of days planned for the task at the time of its start and the number of days spent on its implementation at the time of completion of work on it (W_Days_Delta).

First, the prerequisite for using regression analysis was checked. It was carried out for data collected on the basis of the first five sprints. Using the developed software, a Gaussian distribution was constructed, presented in **Fig. 5.12** [5].

The mean, variance, and standard deviation of the descriptor values collected for the first five sprints were calculated. The mean was 1.52. The variance was 62.72. The standard deviation was 7.92. There were no outliers outside 3σ. The check showed that there were no outliers in the data, so it was decided to build a model using the OLS method. The model calculations were performed for the descriptors collected from the first five sprints.

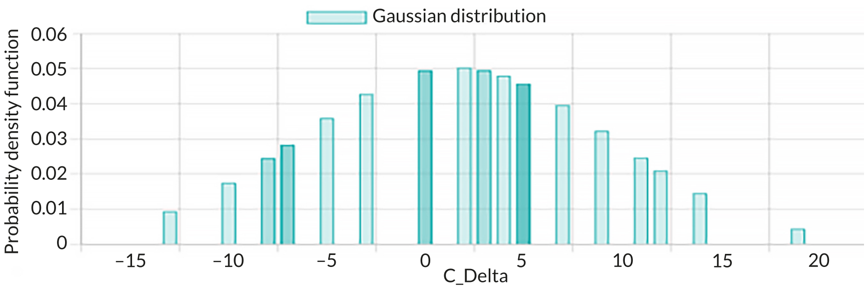

Fig. 5.12 Gaussian distribution for the data on the number of changed characteristics collected for the first five sprints
Source: [5]

The model obtained as a result of the OLS calculations is represented by the formula

$$y_i = 0.46 \cdot x_i + 0.26, \quad (5.6)$$

where y_i – the value of the change indicator W_Days_Delta (system property); x_i – the value of the descriptor C_Delta.

The change model, built using the OLS method for the data of the first five sprints, is shown in the graph (**Fig. 5.13**) [5].

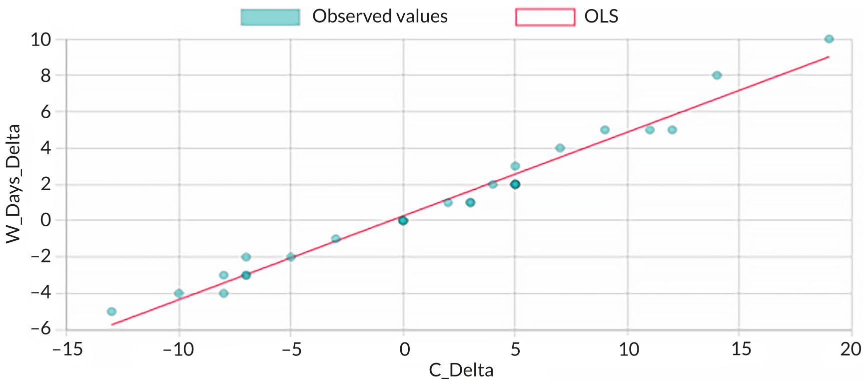

Fig. 5.13 Change model built using the OLS method for data from the first five sprints
Source: [5]

After building the model, its performance was checked using the following indicators: R^2, Q^2, Q^2_{LOOCV}, F-criterion and standard deviation σ. The calculated performance indicators for the change model, built on the basis of descriptors for the tasks of the first five sprints, are given in **Table 5.6**.

Table 5.6 Performance indicators for the change model

Method	Regression equation	R^2	Q^2	Q^2_{LOOCV}	F	σ
OLS	$y_i = 0.46 \cdot x_i + 0.26$	0.98	0.98	0.97	481.61	0.59

The values of the calculated indicators showed that the model is workable and can be used to estimate the change indicator.

During the sixth sprint, issues were resolved regarding updating the list of characteristics of the WEBCO_2360 task after it was put into operation. The initial number of characteristics was 65 units. The team was tasked with choosing one of the proposed options:

– add 20 new characteristics to the current task;

– add 10 priority new characteristics to the current task and create a task with 10 other characteristics to be performed during the next sprint.

The calculation of the change implementation time estimates under the conditions of adding 10 and 20 new characteristics is given in **Table 5.7**.

Table 5.7 Calculation of the change implementation time required for the WEBCO_2360 task due to changes in the number of characteristics

Method	Regression equation	Time of change (20 characteristics)	Time of change (10 characteristics)
OLS	$y_i = 0.46 \cdot x_i + 0.26$	9.46	4.86

Using the resulting change estimate, it was decided to add 10 new features during the current sprint. The project manager assigned people responsible for making changes and checking them.

The changes took 5 days, which was slightly longer than planned. This fact showed that using the improved method, an estimate was obtained that was close enough to the actual value. The calculation of the ratio of the predicted value to the actual value is given in **Table 5.8**.

During the seventh sprint, issues were addressed regarding the reduction of the list of characteristics of the WEBCO_2361 task after it was taken into operation. The initial number of characteristics was 55 units. After taking the task into operation, it was found that 7 characteristics were no longer relevant. The development team requested Stage 5 of the improved method.

Table 5.8 Calculation of the ratio of the predicted value of time to the actual value of time

Method	Forecast change, days	Actual change, days	Ratio of forecasted and actual results
OLS	4.86	5	0.97

To build statistical models of changes, it was decided to use the descriptor C_Delta and the change indicator W_Days_Delta, as for previous calculations. The data collected on the basis of the first six sprints were checked for outliers. The Gaussian distribution for the values of these descriptors, which were collected during the previous six sprints, is shown in **Fig. 5.14** [5].

The calculated mean was 2.4. The variance was 82.6. The standard deviation was 9.09. The three-sigma test showed the presence of an outlier in the data with a value of 34.

Since the test showed the presence of outliers, models were built using Huber regression. The models were calculated on the descriptor values of the first six sprints using the following values of the parameter δ: 1.345, 0.8, 0.1, 0.02, and 1.5. A fitness check was performed for the found change models. The results of the calculations are given in **Table 5.9**.

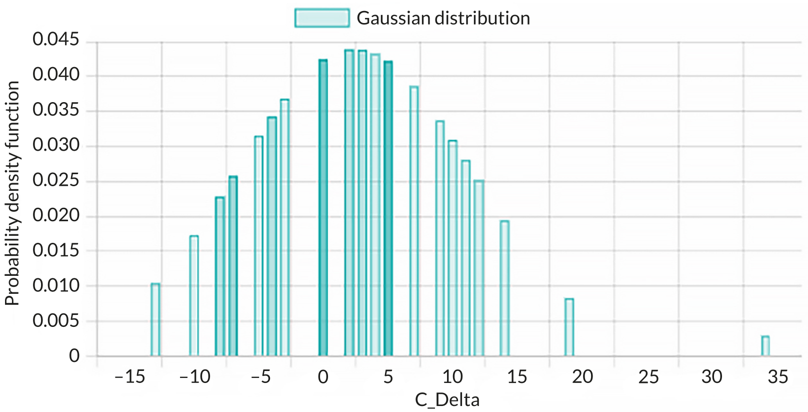

Fig. 5.14 Gaussian distribution for descriptor values collected during the previous six sprints
Source: [5]

Table 5.9 Models calculated using Huber regression and their fitness indicators

Parameter δ	Regression equation	R^2	Q^2	Q^2_{LOOCV}	F	σ
1.345	$y_i = 0.25 \cdot x_i + 0.01$	0.76	0.76	0.78	51	2.09
0.8	$y_i = 0.24 \cdot x_i + 0.01$	0.74	0.74	0.76	46	2.16
0.1	$y_i = 0.37 \cdot x_i + 0.01$	0.94	0.94	0.93	259	1.02
0.02	$y_i = 0.41 \cdot x_i + 0.02$	0.97	0.97	0.97	517	0.74
1.5	$y_i = 0.25 \cdot x_i + 0.01$	0.76	0.76	0.78	51	2.07

According to the calculated performance indicators, the value $\delta = 0.02$ was chosen for further calculations.

The model built using Huber regression for the data of the first six sprints is shown in the graph in **Fig. 5.15** [5].

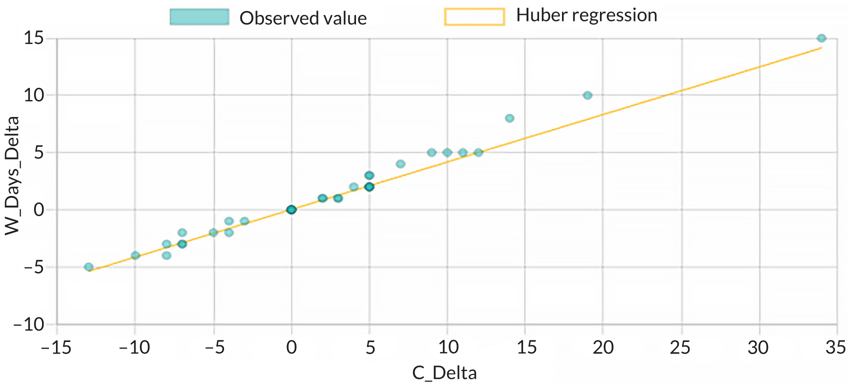

Fig. 5.15 Model built using Huber regression, for data from the first six sprints
Source: [5]

The calculated performance indicators of the model showed that the model is workable and can be used to estimate the indicator of changes in task execution time. The calculation of the estimate of released days under the condition of changing the number of characteristics of the task WEBCO_2361 is given in **Table 5.10**.

For task WEBCO_2361, it was decided to remove 7 characteristics from the task description in the current sprint. 4 days less were spent on solving the changed task. Calculations of the ratio of the predicted value to the actual value are given in **Table 5.11**.

Table 5.10 Calculation of the estimate of released days under the condition of changing the number of characteristics of the task WEBCO_2361

Method	Regression equation	Change indicator
Huber regression	$y_i = 0.41 \cdot x_i + 0.02$	2.85

Table 5.11 Calculation of the ratio of the predicted value to the actual value for task WEBCO_2361

Method	Forecasted value, days	Actual value, days	Ratio of forecasted and actual results
Huber regression	2.85	3.5	0.81

Thus, the calculated change estimates turned out to be close to real changes. Therefore, it was decided that the results of the experimental verification of the improved method and the elements of the corresponding IT are adequate to the real processes of implementing the long-term "Web Constructor" IT project.

5.7 Discussion of the research results

Improving the method based on the Beckhard and Harris's model using the descriptor approach made it possible to solve the following tasks:

– conduct a quantitative assessment of changes based on the values of descriptors that are formed in the process of implementing sprints of a long-term IT project by its performers;

– automate the solution of the problem of quantitative assessment of changes by forming and analyzing statistical models of the dependence of change indicators on descriptors.

The developed IT elements of quantitative assessment of changes allow to improve the quality of change management in a long-term IT project. This possibility was achieved through the use of the descriptor approach and statistical models of quantitative assessment of changes. The use of software implementation developed by IT greatly simplifies the work on assessing changes in the time parameters of individual tasks of a long-term IT project [5].

Unlike the derived method based on the Beckhard and Harris's model [10], the use of the improved method allowed to improve the performance indicators of the change assessment process. The results of comparing the values of these indicators calculated for the derived method based on the Beckhard and Harris's model and after the implementation of the improved method are given in **Table 5.12**.

Table 5.12 Results of comparing the values of indicators of the change assessment process

Indicator	Application of the derivative method based on the Beckhard and Harris's model	Application of the developed information technology for quantitative assessment of changes	Change in the indicator
Proportion (in percent) of time changes canceled due to impossibility of their implementation Ch_{Cp}	7	5	Decreased by 2
Proportion (in percent) of time changes that were accepted by the customer and approved as successfully implemented Ch_{Sp}	92	95	Increased by 3
Difference (in hours) between the time spent on work and the estimated time, δCh_T	7	2	Decreased by 5
Proportion (in percent) of changes that were completed on time Ch_{iTp}	79	86	Increased by 7
Employee attitude assessment towards change management processes (from 1 to 10)	6.575	8.025	Improved by 14.5%

The developed improved method has the following limitations:

– the duration of the project must be large enough to allow collecting the necessary amount of information for analysis;

– there is a need for additional storage of information for analysis;

– models built using this approach are relevant only for specific projects on the basis of which data was collected for building models (however, exceptions are possible).

The main limitation of the use of the developed IT elements is the need to operate IS and IT management of the work of project teams, which could provide the formation of derived data arrays for further calculation of the values of the selected descriptors of the tasks of a long-term IT project. In addition, a significant drawback of the developed IT is a significant increase in the duration of model calculations with an increase in the number of descriptors taken for analysis. To overcome this drawback, it is necessary to conduct additional research to reduce the time and computational complexity of the algorithms for implementing the developed IT [5].

Research on the development of an improved method for quantitative assessment of changes is proposed to focus on determining the possibility of its application for assessing and predicting other (except time) parameters of changes that arise during the implementation of a long-term IT project. To conduct such research, it will also be necessary to conduct additional research to determine the best indicators that characterize changes and the success of the change assessment process in IT projects of various types.

5.8 Conclusions

In the process of research, a method based on the Beckhard and Harris's model was improved for quantitative assessment of changes in a long-term IT project. To improve the method, it was proposed to use a descriptor approach. The improved method, unlike the existing one, allows for quantitative assessment of changes based on the values of descriptors that are formed during the implementation of sprints of a long-term IT project by its performers.

The results of improving the change assessment method were implemented in the form of IT elements of an automated solution to the problem of quantitative assessment of changes in the long-term IT project management system. A service architecture was selected for implementation, and the developed IT was considered as a means of implementing the corresponding service. The IT architecture was defined, a technological stack was proposed, and elements of the software implementation of the service were developed.

To experimentally verify the obtained research results, it was decided to test these results during the implementation of the long-term "Web Constructor" IT project. The test was performed on the results of the work of one project development team. The calculation used data obtained during the team's execution of seven project sprints.

The calculations were performed for two cases of changes that occurred during the sixth and seventh sprints. In the first case, the ratio of the predicted time estimate to the actual time spent on implementing the change was 0.97, in the second case – 0.81. These results showed that the predicted values of the indicators quite accurately coincide with the values of the time actually spent on implementing the changes. In general, the use of the improved method improved the indicators of change implementation during the implementation of the IT project (**Table 5.12**) and improved the team's attitude to the change management process by 14.5%.

Conflict of interest statement

The authors declare that there is no conflict of interest in relation to this paper, as well as the published research results, including the financial aspects of conducting the research, obtaining and using its results, as well as any non-financial personal relationships.

Use of artificial intelligence statement

The authors declare that they did not use artificial intelligence tools in preparing this manuscript.

References

1. Sakhno, Ye. Yu., Sidin, E. P., Korniiets, K. Ye. (2015). Modeling the effectiveness of realization the long-term projects. Scientific Bulletin of Polissia, 2 (2), 87–94.
2. Nastanova do zvodu znan z upravlinnia proiektamy. Nastanova PMBOK (2021). Project Management Institute, Inc. Available at: https://learn.ztu.edu.ua/pluginfile.php/274061/mod_resource/content/1/PMBOK7_Ukr_ForPersonalUseOnly.pdf
3. Managing change in organizations: a practice guide (2013). Project Management Institute, Inc., 127.
4. ISO/IEC/IEEE 12207:2017. Systems and software engineering – Software life cycle processes (2017). IEEE. https://doi.org/10.1109/ieeestd.2017.8100771
5. Vasyltsova, N., Popova, A. (2025). Information technology for quantitative assessment of changes in a long-term IT project. Management Information System and Devises, 184, 5–21. https://doi.org/10.30837/0135-1710.2025.184.005
6. Zosym, M. (2025). Adkar model. Maxym Zosym. Available at: https://www.maxzosim.com/adkar-model/
7. The Prosci ADKAR Model. A powerful yet simple model for facilitating individual change. Prosci. Available at: https://www.prosci.com/methodology/adkar
8. AIM Accelerating Implementation Methodology. Peacock Hill Consulting. Available at: https://imaworldwide.com/the-aim-methodology/
9. Change Management – Embrace Evolve Thrive. AIM Seize The Future. Available at: https://www.aim.com.au/leadership-strategy/courses/change-management-embrace-evolve-thrive

10. Use a Beckhard & Harris Change Process to Create Your Team's Template to Success. Praxie. Available at: https://praxie.com/beckhard-harris-change-process-online-tools-templates-web-software/
11. Bridges Transition Model. William Bridges Assosiates. Available at: https://wmbridges.com/about/what-is-transition/
12. The 8 Steps for Leading Change. Kotter. Available at: https://www.kotterinc.com/methodology/8-steps/
13. Malik, P. (2025). The Kübler Ross Change Curve in the Workplace. The Whatfix Blog. Available at: https://whatfix.com/blog/kubler-ross-change-curve/
14. Malik, P. (2025). Lewin's 3-Stage Model of Change Theory: Overview. The Whatfix Blog. Available at: https://whatfix.com/blog/lewins-change-model/
15. Madumo, G., Marnewick, C., Nyandongo, K. M. (2025). A hybrid approach to manage IT projects. International Journal of Agile Systems and Management, 18 (2), 172–199. https://doi.org/10.1504/ijasm.2025.145451
16. Mikhnenko, P. A. (2025). Mathematical model and intelligent system for analyzing the intensity of megaproject changes: the role of temporary change management hubs. Business Informatics, 19 (2), 54–76. https://doi.org/10.17323/2587-814x.2025.2.54.76
17. Chen, S., Wang, C., Yan, K. (2024). Assessing Project Resilience Through Reference Class Forecasting and Radial Basis Function Neural Network. Applied Sciences, 14 (22), 10433. https://doi.org/10.3390/app142210433
18. Karam, B. A. E., Fissaa, T., Marghoubi, R. (2025). AI-Powered Assessment of Resistance to Change in the Context of Digital Transformation. International Journal of Advanced Computer Science and Applications, 16 (6). https://doi.org/10.14569/ijacsa.2025.0160653
19. Göçmen, I. S., Cezayir, A. S., Tüzün, E. (2025). Enhanced code reviews using pull request based change impact analysis. Empirical Software Engineering, 30 (3). https://doi.org/10.1007/s10664-024-10600-2
20. ISO/IEC/IEEE Standard No 15288:2015 (2015). Systems and software engineering – System life cycle processes. ISO/IEC/IEEE International Standard. https://doi.org/10.1109/IEEESTD.2015.7106435
21. Malhotra, V. (2020). Single Reference Guide for Scrum Certification (Professional Scrum Master I (PSM I) and Professional Scrum Product Owner I (PSPO I) Certification). [Print Replica] Kindle Edition.
22. Larman, C., Vodde, B. (2016). Large-Scale Scrum: More with LeSS. Pearson Education.
23. Van Huffel, S. (Ed.) (1997). Recent Advances in Total Least Squares Techniques and Errors-in-variables Modeling. Society for Industrial & Applied, 377.

24. Diego, O., Essam, H. H., Salvador, H. (2021). Metaheuristics in Machine Learning Theory and Applications. Cham: Springer. https://doi.org/10.1007/978-3-030-70542-8
25. Yak vybraty pravylnyi tekhnolohichnyi stek dlia vashoho proektu. Redstone. Available at: https://redstone.agency/blog/yak-vybraty-pravylnyi-tekhnolohichnyi-stek-dlia-vashoho-proektu/
26. Doar, M. (2011). Practical JIRA Administration. O'Reilly Media, 140.
27. Kobylianskyi, L. S. (2002). Upravlinnia proektamy. Kyiv, 200.
28. Katrenko, A. V. (2011). Upravlinnia IT-proiektamy. Lviv: Novyi svit-2000, 552.
29. Guizzi, G., Fujita, H. (Ed.) (2023). New Trends in Software Methodologies, Tools and Techniques. IOS Press, 371. https://doi.org/10.3233/faia371
30. Roberts P. (2020). The Economist Guide to Change and Project Management. Economist Books, 448.