Analysis and Risk Management in Software Development Using the Logical-Algebraic Model

[1] Khansaa Azeez Obayes Al-Husseini, [2] Ali Hamzah Obaid
[1] Babylon Technical Institute, Al-Furat Al-Awsat Technical University,51015 Babylon, Iraq.
inb.khansaa@atu.edu.iq, azeezkhansaa@gmail.com
[2] Babylon Technical Institute, Al-Furat Al-Awsat Technical University,51015 Babylon, Iraq.
inb.ali210@atu.edu.iq ,alimk_iq@yahoo.com

Abstract. In this paper, we construct a logical-algebraic model that covers all the parameters and operations of the risk process and create Specification of risk assessment processes based on logical-algebraic formalisms-sets, matches, relations, tuples, predicates, and Specification tools to automate the process of risk assessment. And addressing concerns in a change in the software development life cycle that changes requirements depending on the requirements of the development work.

Keywords: Risk Management, Software Intensive Systems (SIS), information technology (IT).

1 Introduction

Risk is an increased probability of loss, The risk is an increased probability at losses, a possible problem that is or not to occur. This, as a rule, arises from the loss of input, control in time. And the possibility of a loss in the process of software development is called the risk of software development. The loss can be any, for example, increasing the cost of production, reducing the quality of software development, the inability to complete the project on time. The risk of software development exists because the future remains uncertain, and there are many known and unknown things that cannot be included in the project plan. The risk of software development There are two types of risks that are either internal and controlled by the project manager or external risks that are outside the control of the project manager To automate the processes of risk assessment, first of all, it is necessary to build the most general model that reflects common entities: the totality of the parameters used and the operations used. Logical algebraic models are most often used to specify objects of modeling at the most general level. When modeling the risk assessment processes, this reveals a multitude of parameter values and a variety of operations of the risk assessment process. When

In: P. Sosnin, V. Maklaev, E. Sosnina (eds.): Proceedings of the IS-2019 Conference, Ulyanovsk, Russia, 24-27 September 2019, published at http://ceur-ws.org

creating appropriate automation tools in the logical-algebraic model, multiple specifications appear that are involved in risk assessment, as well as many software functions for processing these specifications and generating estimates.

2 Software development methodology

is the operation of dividing software development work into distinct stages And aiming to a good organization and to improve the design. Common methodologies include a prototyping, Development (RAD) , Rational Unified Process (RUP) ,waterfall, Most theories attempt to minimize risk by using developing software in little time, called restoration, which typically lasts one week to one month and includes all the tasks necessary " planning, requirements analysis, design, coding, testing, and documentation". Figure (1)

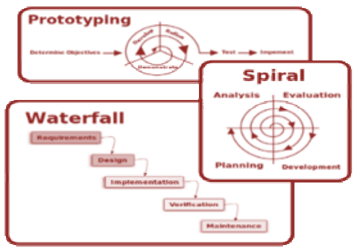

Fig. 1. The basic approaches

3 Software Development Process

A set of steps illustrating the stages of software development shows Figure (2)

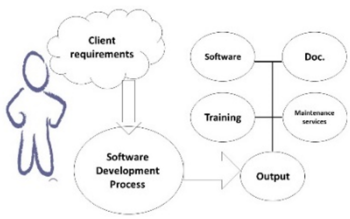

Fig. 2. Software development process.

Figure(2) above oversimplifies the process subtleties The first step is ideas in the process, extract and form these ideas of the requirements, the requirements that feed the process are ideas. Program objectives can be set at the requirements stage.

In the second step, the output of a development process, The user manual, software development documentation, testing, and maintenance is the product of the development process to help developers improve and develop the software in the future

4 Risk Management in Software Development

Risk is the probability of loss. This depends both on the likelihood of adverse events and the consequences of these events; The impact is manifested in a blend of money related misfortunes, time delays and loss of productivity. Hazard is an antecedent to the issue, the probability that at some random point in the product life cycle, Because of these and other factors, every software development project contains elements of uncertainty. the predicted goals will become unattainable within the accessible assets.

5 Empirical Data Sample Analysis

is a proof-based way to deal with the investigation and translation of data. The exact methodology depends on genuine information, measurements, and results as opposed to speculations and ideas. The data sample provided by project consists the following assumptions about the data sample were made:

- Duration of the project.
- Start and finish date.
- Development cost.
- Rate software development.

These findings suggest that the data sample provided by the project the specific features of risk data. Software Development projects have a high rate of failure. A model of software development using the Logical-Algebraic Model describes managerial processes that can be used to reduce software development difficulties from the managerial control perspective.

Our study examined the performance of projects Software Development in relation to the activities of maturity. activities associated with the Software Development control of development-related positively to project performance measures.

Measure of probability

Probability measures in Software Development risk analysis must relate to a specified level of exposure which can, for example, be the consumption of a particular quantity of Project time without work, or an individual exposure event.

. The probability measures are generally expressed in one of two forms:

• The probability of the risk event occurring with a specified exposure event (e.g. probability of illness),

• The average number of risk events that may occur within a specified period.

6 Description of the Risk Assessment Process

Software Development projects have a high rate of failure. Organizations have tried to reduce the rate through methodological approaches but with little perceived success. A model of software development maturity using the Logical-Algebraic Model describes managerial processes that can be used to attack software development difficulties from the managerial control perspective. Our study examined the performance of projects Software Development in relation to the activities of maturity. activities associated with the Software Development control of development-related positively to project performance measures.

Work on risk analysis and the construction of an adequate model of its assessment is very time-consuming. This is due, on the one hand, to the instability of the causes of risk factors, and on the other hand, to the complexity of formalizing the quantitative assessment of performance. Therefore, the justification and development of risk assessment models require a thorough analysis of the nature of the initial information on the causes and risk factors, as well as the purpose of the study.

Depending on the nature of the initial information available at the time of problem statement and the chosen method of uncertainty description, the following classes of mathematical models of risk impact assessment are most common: deterministic; stochastic; linguistic and non-stochastic.

In its most general form, parameterization of risk assessment processes covers the following entities:

where R is an assessment of the consequences of a risk event;

P-probability of occurrence of a risk event;

I-potential consequences of the risk factor.

7 Development of Logical-Algebraic Model

The above description shows that the risk assessment process involves different in nature and mechanisms of representation of the entity, which causes the multi-grade algebraic model. To formalize using the technique of multi-sorted algebras (algebras multistoried) you must first specify the underlying set. This specification is given below.

PSPEC ={ PSpec1, PSpec2, …} – project specification,

Where PSpeci =< attri ,vali> – significant specifier, attr i – attribute i-ro specifier's, vali – attribute value i-ro specifier's.

Dictionary report error RF={rf1, rf2…………} – multiple risk factors,

Where rfi=<rfnamei, pspeci>,

Where rfnamei – risk factor name;

pspeci ∈ PSREC – program specification, which is associated with the i-th risk factor;

PR – set of material values of risk probability;

SR – the set of values of financial losses associated with the fact of risk triggering;

PF – many types of prototyping used to Refine risk assessments;

EST = {est1, est2 ….} – many of the risk assessments,

Where

esti = <rfi, pri,sri> – vector of estimation-set of complex estimations risk the form of triples, where rfi ∈RF, pri ∈PR, sr ∈ SR.

EXP = {exp1, exp2, …} – a lot of experts submitted a spec to the following:

expi = <exp_namei, exp_wi, exp_speci >,

Where exp_namei, exp_wi, exp_speci – name, weight and specialization of the expert;

LOGS = {log1, log2, …} – multiple log entries that capture the results of the risk assessment process;

The functions used in the risk assessment process in the form adopted for the logical-algebraic specification are as follows:

Fest: PSPE × C × Exp × XRF → EST - risk assessment function;

Frf select: RF×SPEC→RF* – function of selection of a subset of risk factors important for the project;

Pspec_select: PSPEC × EST → PSPEC – allocation function specifiers, which have made the greatest impact on value pi и si;

Flog: EXP × EST × DT→ LOGS – a function of General process logging, where DT is a set of data-time values to be recorded in log records.

Pstop: RF × PR × SF → {true, false} – predicate of the iterative process stop " Risk Assessment-Prototyping-Modification PSPEC».

Pnp: RF × PR × SF × PF → {true, false} – the predicate initiate the prototyping process in order to clarify the risk factors.

Composing a system of supporting sets and functions involved in risk assessment, as well as a set of predicates that serve the process, we obtain the following logical-algebraic model in the format of a multi-grade algebra specification:

RfsckA =(RF, PSPEC, EST, EXP | Fest, Frf select, Pspec_select, Flog, P_stop, Prt,Perr,………)

Where Prt - the predicate of the decision about the need to create and execute a prototype of profiling .

Perr - The predicate of the decision about the need to create and perform a prototype evaluation of accuracy.

Strategy pessimistic assessment

$$P_{Stop}^{max} = \sum_{i=1}^{n} P_i^{max} \times S_i^{max} \le P_B$$

for an optimistic assessment

$$P_{Stop}^{min} = \sum_{i=1}^{n} P_i^{min} \times S_i^{min} \le P_B$$

average grade

$$P_{Stop}^{maen} = \sum_{i=1}^{n} P_i^{mean} \times S_i^{mean} \le P_B$$

8 Model of Iterative Process of Risk Minimization in The Design of Automated Systems

Risk assessment - a set of analytical measures to predict the possibility of obtaining additional business income or a certain amount of damage from the risk situation and late adoption of measures to prevent the risk.

The degree of risk is the probability of occurrence of a loss event, as well as the amount of possible damage from it. Risk can be:

acceptable-there is a threat of complete loss of profit from the implementation of the planned project;

critical-possible non-receipt of not only profits, but also revenue and loss coverage at the expense of the entrepreneur;

catastrophic-possible loss of capital, property, and bankruptcy of the entrepreneur.

The risk impact assessment model can be summarized as follows:

F = f(P, I)……

where: F - assessment of the consequences of a risk event;

P-probability of occurrence of a risk event;

I-potential consequences of a risk event.

In General, the construction of a risk assessment model is complicated by the instability of causes or risk factors and the complexity of formalizing the results of activities. Therefore, when justifying and developing risk assessment models, a thorough analysis of the nature of the initial information on the causes and risk factors is required.

From the above we come to an Algebraic model of the process Figure (3).

PPstop = Predicate that detects that all the possibilities to Refine the risk assessment have been exhausted

Pstop=($\sum_{i=1}^{N} P_{\eta} \times S_{\eta} < EST$ Bound

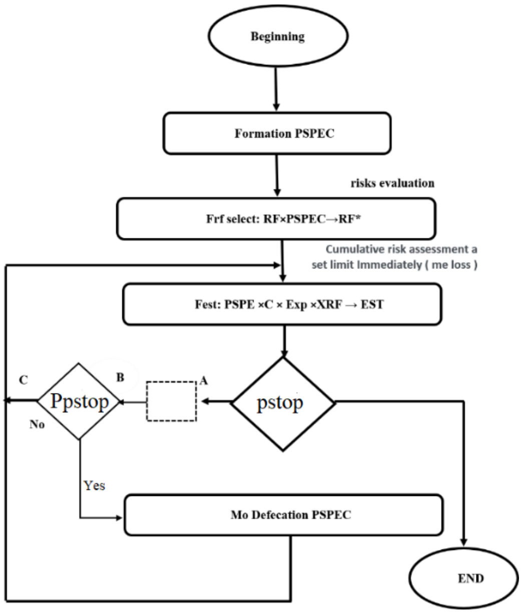

Fig. 3. Algebraic model of the process.

9 Conclusion

The practical importance of research is the ability to use theoretical and methodological results in practice to ensure risk calculation and thus the stability of the means of implementation and development of the project as there are many risks associated with creating high-quality programs on time and with a specific budget. Nevertheless, in order to make sense to take these risks, they should be compensated with sufficient remuneration. The developed tools help the project manager to provide for risk response measures, based on more reliable information. Reliable and more detailed information allows making more effective decisions, which in turn reduces the possibility of risk occurrence. The product of probability and impact determines the importance of risk - its value is an indicator that can be used in the decision-making process.

References

1. Sosnin p.i. pseudo-code control of work flows in the design of automated systems. 2012.
2. Negoda v.n. simulation programs generation based on decision tables translation technics. "interactive systems: the problem of iluman-computer interaction, proceedings of the international conference. ulianovsk, 2001. p.92-93.
3. Khansaa Azeez Obayes Al-Husseini, Information security in the field of technical development and information, Interactive Systems: Problems of Human-Computer Interaction Ulyanovsk: USTU, 71-80p.2015.
4. Sosnin p. question-answer processor for cooperative work in human-computer environment. proceeding the 2 international ieee conference intelligent system, 2004;452-456.
5. Allison robin. msf risk management discipline v.1.1, 2002.
6. Zaitseva, lv, methods and models of adaptation to students in computer training systems, educational technology & society.- no. 4.
7. Khansaa Azeez Obayes Al-Husseini , Ali Hamzah Obaid : development of risk management tools in question-answering based software design environment .international journal of computer science and mobile computing ijcsmc, vol. 7, issue. 6, june 2018, pg.165 – 174 . issn 2320–088x
8. Ali Hamzah Obaid , Khansaa Azeez Obayes Al-Husseini : tools for conceptual-algorithmic prototyping of project solutions in software intensive systems design . international journal of engineering technologies and management research, 5(10), 117-122. doi: 10.5281/zenodo.1491903 . issn: 2454-1907 . vol.5 (iss.10): october 2018 .
9. Khansaa Azeez Obayes Al-Husseini , Ali Hamzah Obaid : usage of prototyping in software testing .multi-knowledge electronic comprehensive journal for education and science publications . issue (14), nov (2018) .
10. Khansaa Azeez Obayes Al-Husseini : risk management tools in the design of automated systems . p 287 interactive systems: problems of human-computer interaction . ulyanovsk , russia: ustu, 2017. − 290 p. udc 681.518 (04). isbn 978-5-9795-1692-9
11. Ali hamzah obaid : tools for conceptual-algorithmic prototyping in solving design problems in the development of systems with software ,interactive systems: problems of human-computer interaction . ulyanovsk , russia: ustu, 2017. − 290 p. udc 681.518 (04) . isbn 978-5-9795-1692-9
12. Kotov v. ye. petri networks .- moscow: nauka, 2013.
13. kurgan gs, models, methods and technology of differentiated learning on the basis of the internet. m. williams, 2011.
14. OBAID, Ali Hamzah. Information hiding techniques for steganography and digital watermarking. UDC 681.518 (04) INTERACTIVE S<STEMS: Problems of Human-Computer Interaction.–Collection of scientific papers.- Ulyanovsk: USTU, 2015.- 306 p. 2015, pp. 63.
15. http://iso27000.ru/chitalnyi-zai/upravlenie-riskami-informacionnoi-bezopasnosti/kak-upravlyat-riskami-informacionnoi-bezopasnosti .
16. C.a. alexander, the timeless way of building, new york,, 1979.google scholar .
17. karlm, "software lifecycle models", kth,2006.