International Journal of Security and Its Applications
Vol.9, No.2 (2015), pp.49-60
http://dx.doi.org/10.14257/ijsia.2015.9.2.06

Analysis on Chaotic Characteristics of Software Project Risk System

[1] Ying Qu and [1]He Wang,

[1]*School of economic and management, Hebei University of Science and Technology*
quying1973@126.com

Abstract

Chaos theory has provided a new pattern for understanding the interacting process of risk factors in system analysis. Software project risk system is a nonlinear system and so it also can be seen as a chaos system. This paper introduces chaos theory into the analysis on software project risk systems to find the evolution law of risk system From the point of chaotic dynamics a dynamic model based on three main parameters for software project risk system on the basis of rational assumption is established. According to judging principles of Lyapunov exponent on chaotic characteristics and the calculation of Lyapunov exponent of software project risk system model, this paper proves that the software project risk system is a chaotic one, which lays the foundations for the related risk research based on the chaos theory. Then it discusses chaotic characteristics such as initial sensibility, inner randomness and strange attractor in software project risk system, which gives guidance for risk control. It also brings a new thought for the study of software project risk management from the point of system.

Keywords: *Software project, chaotic characteristics, risk system, Lyapunov exponent*

1. Introduction

With rapid development of information technology (IT), software product is playing more and more important role in modern society and the software industry has already been a significant component in national economy. However, with the growth of the software industry, more and more problems come along with the increasing quantity and scale in the development of software projects. Compared with any other project, the successful rate of the software project largely reduces which forces many enterprises to suffer big lose [1-3]. Therefore, it is necessary to analyze and control the software project risk aiming to complete the software project as planned.

Software project risk management, running through the whole course of software project life circle, is a set of systems of strategy, method, technology and tool which guarantees software projects to go as planned. As the component of regular project management, it obviously contains such activities as risk identification, evaluation, sorting, schedule, supervision, control, etc. In recent years, the study on software project risk management, attracting wide attention at home and abroad, mainly focuses on risk management theoretical system and supporting technology for risk management. The former pays more attention on the research of aspects like risk management theoretical system and frame, while the latter emphasizes to discuss how to realize management model mentioned above with the proper method. At present, the latter researches concentrate on supporting technology of risk analysis. The focus of theses researches lies in some micro-aspects such as happening probability, risk level and the loses due to risk. While profound macroscopic researches on risk occurrence mechanism, evolution and so on are not illustrated enough yet. Therefore, it is difficult to explain the project adversity, crisis phenomenon, inner course and strategy

ISSN: 1738-9976 IJSIA

mechanism. Meanwhile, other limitations of the current management theory and method are also exposed, for example, there are little researches on risk control or risk planning, while these aspects are main purpose for risk management.

As high uncertainty exists in software project, regular methods generally simplify the system through assumption or simulation, and so it can't reflect the full view of system. The disturbance of nonlinear effect among each unit and element makes more and more management theorists inclined to understand the risk system as non-linear and complex one. Through applying non-linear system theory and complex system theory, they can study the developing course of the system, and then solve and illustrate the problem and phenomenon occurring in risk system. From the perspective of system the paper, will make a quantitative study on software project risk management system by the use of chaos theory. It manages to seize characteristics and status evolution of the system in a macroscopic way.

The study has two specific contributions. Firstly, The paper propose a concept of a risk system leading us to study risk management in a system way. It proves that the software project risk system is a chaos system so that it lay a foundation for the further study based on chaos theory. Secondly, the chaotic characteristics of software project risk system are identified to help the risk system realize the self-organization, and thus open up a new thought for software project risk control.

The rest of this paper organized as follows. The second section summarizes the software project risk system. The third section proves the chaotic characters of software project risk system. The fourth section explains the chaotic characteristics of software project risk system and the last section makes a conclusion.

2. Literature Review

2.1. Theoretical Frame of Risk Management

Researches on this area is mainly focused on the problem how to define the complete process of risk management, detailed operation steps of risk management, and design guidelines on the implementation of risk management, etc. This aspect of researches begins earlier, so some classic theoretical frameworks appear.

Software project risk management starts from the study of prediction for software weakness which appears after software crisis in the 1970s. Boehm discussed, for the first time, software project risk management in his book - Software Risk Management and built a theoretical foundation in this field. He pointed out that the software project risk management was composed of risk assessment and risk control. It clearly defined risk management tasks and functions [4].

Another typical representative is, Carnegie Mellon University in America established the biggest base - SEI making researches and practice on software project risk management. SEI created a relative perfect system of software risk management method (Software Risk Management, SRM). Methods of SRM went through the life circle of software system, including the purchase, development and maintenance of software system [5]. Their outstanding contribution is fully aware of the dynamics in the technology process.

Hall (1998) proposed six subjects model [6], which considers the combination between risk management and project management and emphasizes risk evaluating and risk control. It turns out to be a beneficial trial of the combination of theory and practice.

Some other scholars like Walsh believe that methods are not enough for software project risk management. Through achievements of behavioristics and agency theory, they illustrated the big influence of decision maker's motivation and behavior on projects [7]. Samer brought the process of knowledge management to the risk management, and establish the risk management frame based on knowledge [8].

Up to now, the studies on the theory framework for software project risk management tend to be systematic. It mainly includes several classical systems, like Boehm and Charette risk management frame, Higuera and Hainies risk management frame, Leavitt risk management model, CRM model of SEI, etc. It outline the risk management processes that are risk identification, risk analysis and risk estimate, risk planning, risk control and risk tracing.

2.2. Practice on Risk Management

In recent years, scholars have made continuous exploration in the aspect of risk management practice.

(1) Risk identification. Boehm gives top 10 risk list to guide risk management [4]. Zhong Chen, focusing on risk identification and monitoring, discusses how to conduct the risk management of software project from the personal understanding of the concept of software project management, particularity of software project and method of risk management [9]. The literatures on risk identification are fewer than the ones on other aspects of risk management. It conducts with little the guidance of technologies and tools. The result seems subjective and lack of systematic.

(2) Risk analysis and evaluation. This is the hot research topic in the field of risk management. It focuses on using the qualitative analysis method and quantitative analysis method to analyze the risk influences. Edgar presented to evaluate increasing budget resulting from project risk with the method of probability and statistics [10]. Garvey and Powell applied three mathematical techniques (variable stochastic simulation, Taylor series theorem and Monte Carlo stochastic simulation) to evaluate the workload and project scale. Sherer raised to estimate the possibility of project failure with decision tree and Bayesian theory [11]. In recent years, many new methods such as fuzzy comprehensive evaluation, Petri network, influence diagram, and so on [12-16] are taken into the area to give more continuous exploration to risk management practice.

(3) Risk planning. As the systematic research has not gone into this field, literatures about this aspect are relative scarce.

(4) Risk control. Deping Li thinks it is necessary to identify the risk and evaluate its probability and influence in software project risk management. Then a planning to control the risk should be established. The main target of risk control is to degrade the effect of risk [17]. Through the study on risk management background of project, Yan Qin enumerates the risk sorts in software project management, analyzes their sources and concludes the reasons of various risks. Finally, she makes some suggestions on risk management and control of software developing projects [18]. All these studies seems subjective lack of the guidance of technologies and methods.

2.3 Analysis

From the above overview on research situation we can see the following problems. Risk evaluation is the core part both in theoretical framework establishment and practical research, while, other aspects of risk management like risk planning and risk control are paid little attention. As we know, The purpose of risk management is to monitor risk evolution process to reduce the occurrence possibility of a risk and its harm, so risk control should play an important role in whole process of risk management.

Another disadvantage of the existing studies is that there are more microscopic view than MESO or macroscopic view. That is to say, the current researches are lack of guidance of systematic method. Obviously, both the process of risk management, and IT projects itself, full reflects the systemic features, so we should take the risks of software project as a system and using systematic method to study the risk system from the point of MESO or macroscopic to discover the risk evolution law. Based on the above analysis, this paper tries to do some work on the analysis of risk control and system characteristic. As software project is a kind of complex R&D project, there are a lot of uncertainties in its development and application. It can be seen from above researches that the theoretical methods in this field are numerous yet jumbled, therefore, there is no one accepted and authorized can exactly seize these uncertainties. Here we want to jump out of the former research thought, to study the problem from the point of system. Chaos theory is used in the paper to grasp the uncertainty of risk and its evolution law. The establishment of chaos theory brings a reliable theoretical guarantee to nonlinear science. This theory exposes the unification of order and disorder as well as certainty and randomness, becomes a milestone of the correct cosmology and natural philosophy and encourages plenty of scholars in natural science, engineering and mathematics to make deeper researches and exploration. the risk system itself is a nonlinear system, so by using it some novel and interesting conclusion could be found. The application of chaos theory in other fields like financial risk will provide references for this study.

3. Chaos System Identification of Software Project Risk System

3.1. Chaos Systems

Chaos is a likely random movement reflected by deterministic system. Neither caused by outside disturbance nor infinite degrees of freedom, this movement is caused by inner certain nonlinear mechanism in the system. Chaos is a kind of seemingly disordered yet regular complex movement displayed by certain nonlinear system [19]. It is difficult to predict this movement due to the sensitive dependence of chaos to initial value. Chaos theory is a kind of science to study the inner rule of these complex nonlinear phenomena.

3.1.1. Chaotic Characteristics: The chaos movement has several important characteristics as follows [19].

(1) The sensibility to initial condition. Chaos is sensitively dependent on initial condition. Any tiny difference of initial condition will result in huge difference of the future chaos tracks. At the beginning, two tracks are adjacent. But as the time passes, they become totally unrelated.

(2) The ergodic property of chaos movement. The chaos movement can reach any point in chaos district within limited time.

(3) The geometrical morphology of chaos movement track in phase space has the fractal characteristic. The fractal is a kind of system which is extremely fragmentary and complex. But it also has self-similarity and self-affinity. The fractal is used to describe some strange structures existing in the nature which could not be produced by Euclidean Geometry.

(4) Chaotic attractor. In phase space, chaotic movement forms various strange attractors which have exquisite inner structures.

3.1.2. Method of Lyapunov Exponent: At present, LyaPunov exponent is one of statistical characteristic values that can show the significance in terms of chaotic movement [20-21]. It can measure the average convergence and divergence of nearby tracks in phase space. Chaos system is described by strange attractors of irregular tracks in phase space. One of the obvious characters of strange attractor is the exponent separation of its proximal point. As the points in phase space express the whole physical system, the exponent separation of proximal point means that the system with totally certain initial state will unavoidably change in the case of long time. This movement reflects the system is sensitive dependent on initial condition. The Lyapunov exponent just can quantitatively express this moving state of strange attractors.

While in three-dimensional system, Lyapunov exponent meet following relationships [22].

(K1, K2, K3) = (-, -, -): stable fixed point;

(K1, K2, K3) = (0, -, -):limit cycle;

(K1, K2, K3) = (0, 0, -): two-dimensional torus;

(K1, K2, K3) = (+, +, 0): instable limit cycle;

(K1, K2, K3) = (+, 0, 0): instable two-dimensional torus;

(K1, K2, K3) = (+, 0, -): chaotic attractors.

When chaotic attractors exist, the system can be judged as chaos system [23].

3.2. Lyapunov Exponents of Software Project Risk System

Various as risk factors are in software project; they can be summed up from a big picture, into three main aspects: management factor, human resource factor and technique factor. There are complex interaction relationships among these three aspects. Therefore, we should take these three factors as systematical factors, seizing evolving rule of risk system through the study of their relationships. Combing some references and experience of participants in the same industrial, we can obtain the relationship among these factors.

1）Aspect of management

From the aspect of management, software project risk is inversely proportional to the management level, and proportional to the number of people. The higher the management level is, the lower the risk of it is. More people involving in the software project will make the risk become higher because with the growth of participants, the uncertain risks will become more.

2）Aspect of human resource

From the aspect of people, software project risk is proportional to the management level, and inversely proportional to the number of people as well as management and skill. The higher management level is, the more restrictions to people are, which will reduce people's enthusiasm on project and increase its risk. More people involving in the software project will make the risk become lower because with the growth of participants, the methods to solve the problems will become more. The level of management and skill decides whether the project will get success or not. The higher the level is, the lower the risk is.

3）Aspect of technique

From the aspect of technology, software project risk is proportional to the management and people, and inversely proportional to the technology. Higher management level and more people will make problems easier to people and then the risk will become lower. On the contrary, higher level of technology will make participants feel difficult to solve the problems and thus, the risk will become higher.

Relationship mentioned above can be expressed by following models referenced from Lorenz model [24-25].

(1) Management risk: $f = -\sigma x + \sigma y + a$

(2) Human resource risk: $f = rx - y - xz + c$

(3) Technology risk: $f = xy - bz + d$

The steady equation of the revised software project risk system model is:

$$\begin{cases} -\sigma x + \sigma y + a = 0 \\ rx - y - xz + c = 0 \\ xy - bz + d = 0 \end{cases}$$

It is a system of linear equations about differential. Its Jacobian matrix is

$$\begin{bmatrix} -\sigma & \sigma & 0 \\ r - z & -1 & -x \\ y & x & -b \end{bmatrix}$$

Therefore, its eigenvalue determinant can be worked out as

$$\begin{vmatrix} -\sigma - \varpi & \sigma & 0 \\ r - z & -1 - \varpi & -x \\ y & x & -b - \varpi \end{vmatrix} = 0$$

Namely: $\varpi^3 + \beta_1 \varpi^2 + \beta_2 \varpi + \beta_3 = 0$

Thereinto, $\beta_1 = \sigma + b + 1$, $\beta_2 = b + \sigma + b\sigma - \sigma r + \sigma z + x^2$, $\beta_3 = b\sigma + xy\sigma - \sigma br + \sigma bz + x^2\sigma$

And σ , r , a and b are parameters, c 、 d is the uncertainty risk factors.

Based on Girolamo Cardano, making $p = -\frac{\beta_1^2}{3} + \beta_2$, $q = \frac{2\beta_1^3}{27} - \frac{\beta_1 \beta_2}{3} + \beta_3$

$A = -\frac{q}{2} - (\frac{q^2}{4} + \frac{p^3}{27})^{\frac{1}{2}}$, $B = -\frac{q}{2} \_ (\frac{q^2}{4} + \frac{p^3}{27})^{\frac{1}{2}}$

Through the experiment, the following conclusion can be drawn. If $0 < \sigma < \frac{3}{2}$, $\frac{4}{7} < r < \frac{11}{3}$, $0 < a < 4$, $b = 1$ then the real parts of three results of eigenvalue ϖ_1 , ϖ_2 , ϖ_3 meet following relationships: $\varpi_1 > 0, \varpi_2 = 0, \varpi_3 < 0, \varpi_1 + \varpi_2 + \varpi_3 < 0$. Three Lyapunov exponents of chaos attractors are (+, 0, -,), and Lyapunov exponent just meet this relationship. Therefore, there are indeed chaos attractors in software project risk system and then, it is a chaos system.

3.3. Verification of Chaos System

First of all to define the lorenz model, and then based on the Jacobi phalanx of we get A, Using the eigenvalue of square matrix A lambda is to make the homogeneous linear equations (A-λE) x = 0 has A non-zero value of lambda and then our results are obtained.

1) Equations for Lorenz system

Input: Coefficient a, b, c, independent variable x

Output: solve y

```
Step1: dy (1) =a*(x (2)-x (1));
Step2: dy (2) =c*x (1)-x (3)*x (1)-x (2);
Step3: dy (3) =x (1)*x (2)-b*x (3);
Step4:  y=[dy(1); dy(2); dy(3)];
```

2) Establishing the equations of Runge- Kutta system

Input: for solution of equations f; initial argument x0; initial function value y0; step length h; step number n.

Output: The final function value T

```
Step1: r=y0
Step2: s=x0
Step3: r=r+s
Step4: $T = \left\{\begin{matrix} 0 & 0 & ... & 0 \\ 0 & 0 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & 0 \end{matrix}\right\}$ (r line n+1 row)
Step5: T (:, 1)=[y0; x0];
Step6: let t=2
Step7: k1=f (T (1: r-1, t-1));
Step8: k2=f (k1*(h/2) +T (1: r-1, t-1); x0+h/2);
Step9:k3=f (k2*(h/2) +T (1: r-1, t-1); x0+h/2);
Step10:k4=f (k3*hot (1: r-1, t-1); x0+h);
Step11:x0=x0+h;
Step12:T(:,t)=[T(1:r-1,t-1)+(k1+k2*2+k3*2+k4)*(h/6);x0];
Step13: t=t+1;
Step14: If t n+1 back  to Step7; or end.
```

3) Solve lorenz equations

The writer brings Lorenz expressions into Runge-Kutta equations. Makes several parameters, and solves lorenz function value. Number of the modified Lorenz model is calculated. If parameters a=2；b=1； =0.5， =2；We put it at the center of the difference equation. The Lorenz model can be fixed In (y, z) plane numerical simulation results. The chaotic attractor can be seen in Figure 1.

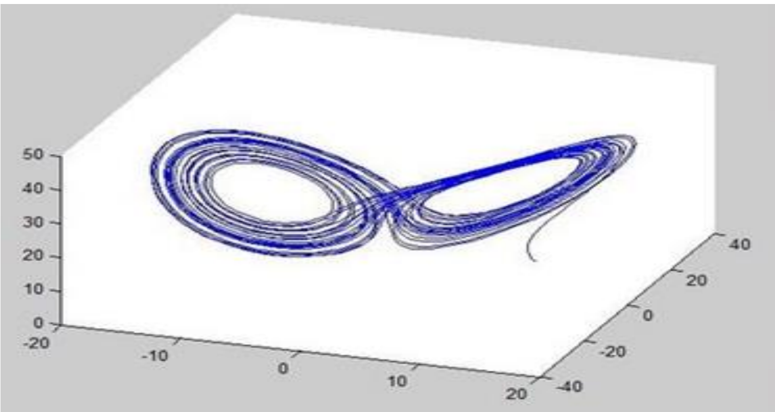

Figure 1. Modified Chaotic Attractors

4. Chaos Characteristics in Software Risk System

The premise to analyze the software project risk management with the chaos theory is that the software project risk management system is a chaos system or it is in the chaos environment. The third part of this paper has quantificationally proved that this system is a chaos one and it has following chaotic characteristics.

4.1. Software Project Risk System is an Open and Dynamic Dissipative System

The software project risk system is not closed at all. The main purpose of risk management is to make a good management on risk factors of the software project so as to reduce the unnecessary loss. As a specific project, the software project ultimately aims to satisfy the need of the enterprise. And from the earnings results, it should also adapt to requirement changing. In this process, the software project will definitely exchange the information with the internal and external environment in order to get necessary substance and energy. Meanwhile, the risk management will also be unscheduled disturbed by some inside or outside factors so that it will deviate from its anticipant direction. As an open and dynamic system with the chaotic property, software project risk system continuously dissipates various data inputted from the outside. At the same time, this system will output the information to users through different transforming mechanisms in the software project risk management.

The software project risk management system is dynamic and irreversible. With the constant development of the software project, some purposeful activities should be adopted to promote external input to turn into your goal quickly. Meanwhile, software projects as one-off operational work over a period of time, which decides that the course of the software project risk management is irreversible in terms of the strategic direction and the practical process. The dissipativeness of the software project risk system not only keeps the system open, dynamic and flexible, but also enable the system to obtain the global stability.

4.2. The Software Project Risk Management System is a Nonlinear System

The software project risk management system is a nonlinear system in which the relationship between the relationship of input and output is not the simple linear one. There are many nonlinear relationships in this system. For instance, the relationship between the input of the manual labor, material resources and financial resources of the software project risk management and its ultimate result is nonlinear. The restrained relationship among five specific periods of the software project risk management reflects the nonlinear relationships among the internal elements.

Many factors will affect the specific activities in the software project. To sum up, there are three aspects. The first factor is the human resource which includes project developers, administrators for risk management, and so on. People with different capability and experience will lead to the instability and uncertainty of the software project risk system. These factors such as demand changes, the different combination of developers, different risk solutions are related with personnel. The relationships among these factors and their influences are nonlinear, rather than linear. The second kind of factor is technology. It mainly includes whether the skill is advanced, whether the application prospect of it is good, whether it is easy to be accepted and whether its lifetime and effect are uncertain. The third one is the management which mainly includes two aspects, namely the risk management and the software project organization management. The latter including the supply of money for the software project, the effective software project organization management, the capability of people in development, the capability of relevant administrators and so on. Different measures adopted by the risk administrator in different stage will have a significant impact on the result which is also uncertain. The factors mentioned above do not tally with the theory of linear superposition. The mutual influence among them is nonlinear and when these factors are organized in a system, a new property of system will appear.

4.3. Software Project Risk System is Sensitively Dependent on Initial Condition

The influence of stakeholders on the feature of the ultimate software project and the expense of the project is most strong in the initial period of the software project development because the need of customers has relative little influence in this period. For instance, a wrong decision of the manager in the initial period of the project will cause the big divergence of the ultimate product from the anticipant one, even the total failure of the software project. However, a proper incentive measure to the staff could increase the staff's enthusiasm, which will largely raise the quality of the software project and largely reduce its risk. If the group of the software project development could make friends with customers in the course of development and cooperate with them in implementation, they will enable the result of the software project to better satisfy the need of customers and then, largely increase the ability of the software project risk management system to adapt to the outside environment.

What's more, uniqueness of the software project let us know there are no totally same software projects because something different or unpredictable will often occur in the period of the software project risk management. Even the same decisive model of the software projects may cause different behaviors of the software project risk management system. Due to different culture, experience and knowledge of the inside and outside environment and the risk management, the administrators will make different decision. Therefore, from different points butterfly effect, namely the property of initial value sensitivity has been proved again to exist in the software project risk management.

4.4. Software Project Risk System has Inner Randomness

In the software project risk system, the relationship among elements of it is nonlinear. Human organization and requirements may be different in each stage of the software project. The uncertainty of the software project will be much higher than that of other conventional projects if these uncertain factors are combined in the period of development. While in software project risk system, as the risk administrators have different experience, capability to sense the risk and plans to deal with the risk, these factors will also increase the uncertainty of the system. Therefore, software project risk system is one with dissipative structure which is composed of complex relationships among relative participants and some specific rules among risks. This system has a characteristic of inner randomness.

There are several reasons for the uncertainty of the software project risk system. Firstly, the cognitive ability of the staff is limited. They cannot make accurate predictions about the whole course and result of the software project. It is not avoidable for them to make errors in stages such as requirement analysis, project plan, schedule arrangement, etc. Secondly, the changing of outside environment will cause a difference of real requirements from its initial ones in virtually, which will also enable the software project to show out the distinct complexity. Thirdly, the software project is something of high intelligence which means it is highly difficult work. People's capability will exert a big influence on its result. However, different people hold different idea to judge whether one is fit for the development of the project, which will also bring some risks to the project. Finally, factors like administrators' experience and ability to deal with the risk make the software project risk management system more complex, which leads to the randomness, volatility, and long-term unpredictability of the software project risk management.

4.5. Software Project Risk System has Strange Attractors

Strange attractors, namely chaotic attractors, are the set of all attractors pointing to the space. The chaos track is around the set. The attractor has fractal dimension and it's a moving point. But it is neither periodic attractors nor limit cycle.

As a characteristic of chaos, strange attractors reflect in geometry that tracks departing from different initial states will finally concentrate on a phase space. That is to say the whole movement reflects purposiveness.

The strange attractor in software project risk management is the minimum of software project risk. The research staff of the software project make all work such as requirement analysis, development plan, schedule arrangement, test and implementation move around the core of quality maximum and risk minimum, which will enable the ultimate project product to meet customs' demand and reduce the risk to maximum extent. In order to realize this essential task, managers of the software project will make management plan and relative risk management plan to timely find out risks in development and then, to deal with them through proper methods. Even so, because of nonlinear relationships among various factors in software project risk management, managers have to properly deal with the plentiful unordered quality and risk problems in reality.

5. Conclusions

According to characters of software project risk management, this thesis, through Lyapunov exponent, proves this system as a chaotic one and makes a study of its chaotic characters. Software project risk system is not only a complex nonlinear system, but also one sensitively dependent on initial condition and a dynamic and open dissipative system. The reason of producing chaos is the uncertainty of project management in enterprise. Based on these chaos characteristics, managers can adopt corresponding management solutions to realize the unification between order and disorder, further improve the software project risk management level and then, realize the object of software project risk management. With various proving methods of chaotic system, this thesis only use Lyapunov exponent to testify. Other methods can also be tried to verify. Making a deep exploration on chaotic characters of software project risk system, this thesis builds a theoretical foundation on the study of software project risk chaotic system. While the further research through theories such as chaotic principles and chaotic dynamics on planning, organization and control of software project risk system under the chaotic environment still waits to be made.

Acknowledgments

This paper is supported by the Project of Natural Science Foundation of China (71301044) and the Project of Natural Science Foundation of Hebei Province (G2011208087).

References

[1] The Standish Group, "Chaos summary, (**2009**), http://www.standishgroup.com. Retrieved May 20, 2010.

[2] J. Liu, Q. Z. Wang and Q. G. Ma, "The effects of project uncertainty and risk management on IS development project performance: A vendor perspective [J]", International Journal of Project Management, vol. 29, no. 7, (**2011**), pp. 923-933.

[3] K. de Bakker, A. Boonstra and H. Wortmann, "Risk managements' communicative effects influencing IT project success", *International Journal of Project Management*, vol. 4, no. 30, (**2012**), pp. 444-457.

[4] B. W. Boehm, "Software risk management: principles and practices", IEEE Software, vol. 8, no. 1, (**1991**), pp. 32-41.

[5] R. P. Higuera and Y. Y. Haimes, "Software risk management [R]", New York: Software Engineering Institute, (**1996**).

[6] E. M. Hall, "Managing risk-Methods for Software System development [M]", Addison-Wesley, (**1998**), p. 225.

[7] K. R. Walsh and H. Schneider, "The role of motivation and risk behaviors in software development success [J]", Information Research, vol. 7, no. 3, (**2002**), pp. 21-25.

[8] S. Alhawari, L. Karadsheh, A. N. Talet and E. Mansour, "Knowledge-Based Risk Management framework for Information Technology project [J]", International Journal of Information Management, vol. 32, no. 1, (**2012**), pp. 50-65.

[9] Z. Chen, "A software project risk management [J]", Economic and Social Development, vol. 12, no. 2, (**2004**), pp. 67-71.

[10] J. D. Edgar, "Controlling murphy: how to budget for project risk [J]", Concepts, vol. 2, (**1982**), pp. 60-73.

[11] S. A. Sherer, "Software failure risk: measurement and management [M]", New York: Plenum Press, (**1992**).

[12] A. Engel and M. Last, "Modeling software testing costs and risks using fuzzy logic paradigm [J]", Journal of Systems and Software, vol. 80, no. 6, (**2007**), pp. 817-835.

[13] E. Lee, Y. Park and J. G. Shin, "Large engineering project risk management using a Bayesian belief network [J]", Expert Systems with Applications, vol. 36, no. 3, (**2009**), pp. 5880-5887.

[14] D. Aloini, R. Dulmin and V. Mininno, "Modelling and assessing ERP project risks: A Petri Net approach [J]", European Journal of Operational Research, vol. 220, no. 2, (**2012**), pp. 484-495.

[15] A. W. Juliano and A. B. Luís, "A framework for risk assessment based on analysis of historical information of workflow execution in IT systems [J]", Computer Networks, vol. 55, no. 13, (**2011**), pp. 2954-2975.

[16] H. J. Hsiau and C. W. R. Lin, "A fuzzy pert approach to evaluate plant construction project scheduling risk under uncertain resources capacity [J]", Journal of Industrial Engineering and Management, vol. 2, no. 1, (**2009**), pp. 31-47.

[17] P. De Li, "For a software development project risk management [J]", Computer and Internet, (**2004**), pp. 57-61.

[18] Y. Qin, "A software project risk management research [J]", The Enterprise Management, vol. 11, no. 2, (**2010**), pp. 89-92.

[19] T. L. Carrol and L. M. Pecora, "Synchronization in chaotic systems [J]", Physics Review Letters, vol. 64, no. 8, (**1990**), pp. 821-824.

[20] P. Pepe, "Direct and converse Lyapunov theorems for functional difference Systems [J]", Engineering Applications of Artificial Intelligence, vol. 50, no. 12, (**2014**), pp. 3054-3066.

[21] M. S. Sadeghi, M. Rezaei and M. Mardaneh, "Affine parallel distributed compensator design for affine fuzzy systems via fuzzy Lyapunov function [J]", Engineering Applications of Artificial Intelligence, vol. 37, (**2014**), pp. 407-416.

[22] J. Jiang, "The enterprise integrated risk early warning model based on chaos theory", Journal of Electronics [J]", (**2005**), pp. 41-47.

[23] G. Hu and J. Xiao, "Chaos control [M]", Shanghai science and technology education press, (**2000**).

[24] D. Roy and Z. E. Musielak, "Generalized Lorenz models and their routes to chaos [J]", Energy-conserving Vertical Mode Truncations, vol. 32, no. 3, (**2007**), pp. 1038-1052.

[25] R. Blender and V. Lucarini, "Nambu representation of an extended Lorenz model with viscous heating [J]", Energy Conversion and Management, vol. 243, no. 1, (**2013**), pp. 86-91.