

Chapter 18

Agent-Based Approach for Monitoring Risks in Software Development Projects

Jirapun Daengdej

Assumption University, Thailand

ABSTRACT

According to various surveys conducted, regardless of how many studies in software development projects have been done, the chance that software development projects may fail remains very high. A relatively new approach to the problem of failure is using the concept of artificial intelligence (AI) to help automate a certain part(s) of the projects in order to minimize the issue. Unfortunately, most of the works proposed to date use AI as a standalone system, which leads to limiting the degree of automation that the overall system can benefit from the technology. This chapter discusses a preliminary work on a novel risk monitoring, which utilizes a number of agent-based systems that cooperate with each other in minimizing risks for the projects. The proposed model not only leads to a high degree of automation in risk management, but this extensible model also allows additional tasks in risk monitoring to be easily added and automated if required.

INTRODUCTION

Issues of software development project failure has long been recognized long time ago. In 1994, one of the most talk about reports by Standish Group called “Chaos Report” rises the issue that only around 16% of software development projects success worldwide (Clancy, 1994). Almost a decade after that, in 2012, a research team of McKinsey Digital in collaboration with University of Oxford found that only around 50% of projects, which cost more than \$15 million, completed projects within their budgets. In fact, they report that large IT projects usually spend 45% over budget, while delivering only around 56% less value than what originally planned (Bloch, et. al., 2012). Regardless of how much resources and researches have been put in area of Software Project Management, chance of software project failure

DOI: 10.4018/978-1-6684-3702-5.ch018

still remains very high. According to the latest survey by KPMG, 70% of organizations has at least one project failure in the past 12 months (Krystal, 2019).

One of the reasons of such a high rate of failure is because there are a large number of risks can occur during project lifecycle. As a result, these risks have to be closely and efficiently monitored in order to minimize issues that may occur. Risk monitoring is considered to be one of the most crucial areas in project management in general. Unfortunately, the necessity and difficult in monitoring these risks can easily be magnified when comes to software development projects. The risks are higher in software development projects than other kind of projects because software is intangible in nature. Successfulness of the projects, in many cases, cannot be easily seen and judge explicitly. Progress of the software projects is also very difficult to measure. This is the reason why agile concept, which focus on final products rather than what happen during the project, has been recently introduced and rapidly accepted and adopted by industry (Burger, 2018)(Wolpers and West, 2019).

The goal of this research is to investigate how agent-based approach proposed by Artificial Intelligence (AI) community, which has already been applied in various areas, can be used in risk monitoring of software project management. This chapter provides a preliminary discussion on how the agent-based approach can be used in performing the task and layout issues that require further investigations for those who are interested.

BACKGROUND

The following discusses three main concepts related to focus of the paper. These concepts include project management, risk management, and agent-based approach.

Software Project Management

Project management has been considered as one of the most mature fields as far as industry is concerned (Spalek, 2005). However, regardless of the success of the field in helping projects to meet their expected results, a large number of software development projects are still considered as challenges or even fails (Coronado and Jaén, 2002). According to Anantatmula and Anantatmula, M. (2008), regardless of the success and failures occurred until today, in general, software project management consist of the following steps:

1. Initiation
2. Project Planning
3. Project Execution
4. Project Monitoring and Controlling
5. Project Termination

As far as the failure or challenges is concerned, according to Arnuphaptrairong (2011), example of issues that can occur during the projects include:

- Misunderstanding of requirements
- Lack of top management commitment and support

9 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/agent-based-approach-for-monitoring-risks-in-software-development-projects/294473

Related Content

Internet of Things (IOT) in Healthcare – Smart Health and Surveillance, Architectures, Security Analysis and Data Transfer: A Review

Parthasarathy Panchatcharamand Vivekanandan S. (2019). *International Journal of Software Innovation* (pp. 21-40).

www.irma-international.org/article/internet-of-things-iot-in-healthcare--smart-health-and-surveillance-architectures-security-analysis-and-data-transfer/223520

ODACE SLA: Ontology Driven Approach for Automatic Establishment of Service Level Agreements

Kaouthar Fakhfakh, Tarak Chaari, Said Tazi, Mohamed Jmaieland Khalil Drira (2010). *International Journal of Systems and Service-Oriented Engineering* (pp. 1-20).

www.irma-international.org/article/odace-sla-ontology-driven-approach/47035

Secure Key Generation for Static Visual Watermarking by Machine Learning in Intelligent Systems and Services

Kensuke Naoe, Hideyasu Sasakiand Yoshiyasu Takefuji (2012). *Theoretical and Analytical Service-Focused Systems Design and Development* (pp. 106-121).

www.irma-international.org/chapter/secure-key-generation-static-visual/66795

A Method for Formulating and Architecting Component- and Service-Oriented Systems

Gerald Kotonya, John Hutchinsonand Benoit Bloin (2005). *Service-Oriented Software System Engineering: Challenges and Practices* (pp. 155-181).

www.irma-international.org/chapter/method-formulating-architecting-component-service/28954

Improving Effectiveness of Process Model Matchers Using Wordnet Glosses

Mostefai Abdelkader (2019). *International Journal of Systems and Software Security and Protection* (pp. 1-19).

www.irma-international.org/article/improving-effectiveness-of-process-model-matchers-using-wordnet-glosses/247489