LINKING SOFTWARE DEVELOPMENT PHASE AND PRODUCT ATTRIBUTES WITH USER EVALUATION: A CASE STUDY ON GAMES

Özge Bengur[1] and Banu Günel[2]

Informatics Institute, Middle East Technical University, Ankara, Turkey
[1]bengur@alumni.bilkent.edu.tr
[2]bgunel@metu.edu.tr

ABSTRACT

This paper presents an evaluation methodology to reveal the relationships between the attributes of software products, practices applied during the development phase and the user evaluation of the products. For the case study, the games sector has been chosen due to easy access to the user evaluation of this type of software products. Product attributes and practices applied during the development phase have been collected from the developers via questionnaires. User evaluation results were collected from a group of independent evaluators. Two bipartite networks were created using the gathered data. The first network maps software products to the practices applied during the development phase and the second network maps the products to the product attributes. According to the links, similarities were determined and subgroups of products were obtained according to selected development phase practices. By this way, the effect of development phase on the user evaluation has been investigated.

KEYWORDS

Software development, software product evaluation, bipartite networks, social network analysis

1. INTRODUCTION

There are several methods for evaluating software products and different analysis techniques for presenting the results of the evaluation. In software projects, the methods applied in development process or product attributes may affect the end user's evaluation. Therefore, the aim of this study is to investigate and visualize any relations by using a "network analysis" technique between three groups of parameters related to the development phase, the product itself and the end users' evaluation. While there are studies that research the reasons of failures of the software projects, the reasons that lead to successful software from the point-of-view of end users are often neglected.

A software development methodology can be defined as an outline intended for structuring, planning, and controlling the process of developing software [1]. As noted by Centers for Medicare & Medicaid Services' Office of Information Service (2008), software development methodologies are there to apply definite frameworks with the intention of developing and/or maintaining software.

Surely, an integrated approach is needed for the organizations providing enterprise-wide software solutions. Therefore, organizational assets shall be managed via an integrated approach for business success. Happily, maturity models, standards, methodologies, and guidelines are there for these organizations to improve the way they do business in such settings.

Sundarapandian et al. (Eds) : ICAITA, SAI, SEAS, CDKP, CMCA-2013
pp. 397–413, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3833

However, there is no any single development, in either technology or management technique which independently assures even one order-of-magnitude improvement in productivity, reliability, success, or simplicity [2]. Organizations try to compose their best practices based on the literature and their own success and failures for software development.

Concerning the evaluation dimension, in the literature there are a lot of studies dealing solely with the user evaluation and other forms of evaluations for software success. However, other than the "Chaos Report" by The Standish Group, 1995, there is no conveniently relevant study in the literature addressing the link between software success and development methodologies, as to the best extent of our researches [3].

Metrics exist for the evaluation of software products from the end-user perspective, especially satisfaction and acceptance [4, 5]. However, these metrics, which include efficiency, effort, accuracy, etc., are not applicable to games. Game metrics, on the other hand are still under development and it is not clear how well the proposed metrics, such as the player character's speed, movement, interaction, points obtained, spaces discovered in the game worlds, such as the number of active users on a social game, sales figures, revenue, etc., correlate with the end user satisfaction. So far, the studies on game user satisfaction focus on online games and investigate the effect of network latency [6, 7].

Various guidelines have also been generated for the development of open source software, such as the Open Source Software Quality Observation [8], Method for Qualification and Selection of Open Source Software (QSOS) [9] and Open Business Readiness Rating [10]. However, for other types of software, especially for those such as games, the effect of software development phase on the user experience of the end product has not been investigated before.

Social networking approach has been utilized for quality assessment of software before. However, the focus was end users, not developers [11]. The majority of the works that utilise social network analysis in software engineering research focus on the communication among the project team members since the collaboration is essential for requirements engineering [12, 13].

In order to determine the effects of the development process and product attributes on the end user evaluation, a new approach based on network analysis has been applied. This approach is powerful in both quantifying and visualising relationships and may reveal more information which is not possible with any other traditional statistical techniques.

Section 2 provides a brief overview of the social network analysis concepts and the bipartite network structures. Section 3 gives the overview of the data collection and bipartite network formation methodology. Section 4 explains different analyses applied on the network data and the results obtained. Discussions of the results have also been provided after the analysis. Section 5 concludes the paper.

2. OVERVIEW OF NETWORK ANALYSIS

Network analysis technique has been chosen for finding the effects of the development process or product attributes on the end user evaluations. The techniques applied are frequently used in social network analysis (SNA), which provides a methodology to detect, analyse and interpret social ties among actors [14]. However, in this study, networks are not social networks, but bipartite networks which were created by mapping. Yet, some key terminology and concepts also apply to these types of networks. Therefore, a brief overview of network analysis is provided below.

The smallest unit in a network is called a vertex (pl. vertices). In SNA it represents an actor. Some example vertices can be people, societies, countries, companies, products, etc. A vertex may be connected to any other vertex with a line. A line is either directed, which is called an arc, or undirected, which is then called an edge. There also exists a special kind of line, called loop, which connects a vertex to itself. A graph is formed with a set of vertices and a set of lines between pairs of vertices. To be able to talk about a network, however, one needs to have additional information on the vertices or the lines in addition to the graph.

In a one-mode network, each vertex can be related to any other vertex. However, in a bipartite network (aka two-mode network or affiliation network), vertices are divided into two sets and vertices can only be related to vertices in the other set. For example, the elements of the first set can be software development companies and the elements of the other set can be the programming languages used. Therefore, lines indicate which programming languages companies use. From a bipartite network, two one-mode networks can be obtained. For example, for our example, a one-mode network of companies for which the value of a line between a pair of vertices indicate how many common programming languages those two companies use, and a one-mode network of programming languages for which the value of a line between a pair of programming languages indicate how many companies use both languages.

Additional information about the vertices, such as the classifications of vertices to different groups and numeric values that represent demographic and similar data may also be provided and used in network analysis. In Pajek, the former is stored as a partition and the latter is stored as a vector.

3. DATA COLLECTION AND NETWORK FORMATION

For the case study, the data has been collected in the games sector in Turkey. Product attributes and practices applied during the development phase have been collected from the software developers via questionnaires. Additionally, user evaluation results were collected from a group of independent evaluators.

The study has been done by the following steps;

- Finding the features of the companies according to their development process and analysing what kind of development processes they apply.
- Determining the unique attributes and/or general attributes of the products.
- Collecting evaluator ratings on the products.
- Evaluating the effects of the development process and product attributes on the end user.
- Analysing the collected data, finding and monitoring the effects of the development process and product attributes on the user by using “Social Network Analysis”.
- In order to create the network, the answers given to the questionnaire were coded and turned into network vertices.

3.1. Data Collection from Questionnaire

It has been decided to use a questionnaire to collect development process and the product attributes data from game companies. In the questionnaire design, questions about the same domain were grouped together to make it easier to answer the questions [15].

After initial design of the questionnaire was completed, review and clarification phase of the questions were done.

Consequently, final questionnaire consisted of two parts;

- First part (Development Phase) aims to collect the data about the development processes of the companies.
- Second part (Product) aims to collect data about the attributes of the products.

There are ten groups of questions categorized in the first part (Development Phase):

1. Questions to learn the lifecycle model applied during the production of the chosen games.

2. Questions to learn the size of the product (estimated and actual size (LOC)).

3. Questions to know the durations of the development phases (defining requirements, design, coding)

4. Questions about the development team (size and the experience of the team.)

5. Questions about the standards used during the development.

6. Questions about the quality factors considered while developing the selected products.

7. Questions about the maintenance approach of the company for the selected product (game).

8. Questions about the testing strategy of the company for the selected product.

9. Questions about the programming language and the tools used to develop the selected product.

10. Questions about user involvement during the development life cycle of the product.

For the Second Part (Product) of the questionnaire, there are domain specific questions included, besides the general product questions. There are three groups of questions categorized in the second part (Product);

1. Questions about the target platform of the product.

2. Questions about the target users of the product (according to age, gender and education).

3. Domain specific questions about:

- Game category,
- Music attributes the product has,
- Game scenario,
- Skills required to play the selected game, and
- Variety of characters, levels, locations in the game.

Questionnaire was prepared in English, and made available both in printed form and online (no online survey mechanism, but by sending via e-mail and collecting the answers via e-mail).

Questionnaires were sent to the candidate games companies to collect data about their development processes and product attributes.

Answers were received for ten products out of eleven involved in the event. That is, nearly 90% of participation ratio was achieved for data collection.

Instead of their original names, the company names were encoded due to the confidentiality reasons.

3.2. Data Collection from Game Awards Event

In November 2012, "Kristal Piksel Best Local Game Awards" was run by Turkish Informatics Associations in the Informatics 2012 event. This event aimed to award local games on several categories.

Instead of using “game selling numbers”, “total revenues” or “use game ratings” and critiques found in various sources on the internet, forums or magazines, we opted for gathering data from the event and build our dataset upon award’s evaluation process, since the mentioned sources would not reveal as detailed information on the games as the evaluators of the game awards event.

In the event, games were evaluated in six main categories:

- Music
- Aesthetics
- Graphics
- AI (Artificial Intelligence)
- Gameplay
- Story

Points to awarding the games are categorized as follows;

- Point for competing for award in a category (Developers had chosen the category (or categories) they wanted their product to compete),
- Points for passing the first elimination,
- Points obtained from the evaluators.
- Points for winning an award in a category.

3.3. Network Formation

Two bipartite networks were created using the gathered data**.** Figure 1 shows the networks of our study. The first network maps software products to the practices applied during the development phase and the second network maps the products to the product attributes. According to the links, similarities or dissimilarities were determined and subgroups of products were obtained according to selected development phase practices. Accordingly, the effect of development phase on the user evaluation has been investigated.

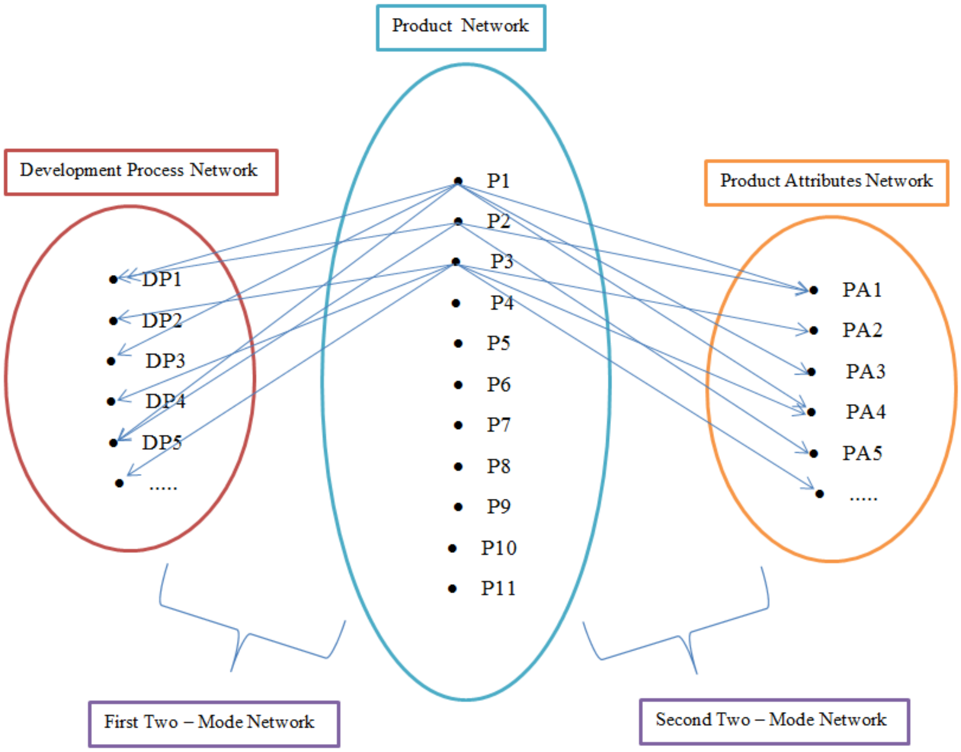

Figure 1. Bipartite network sample of this study

It is possible to group each category by their similarities after determining their connections with each other.

The gathered data has been analysed on Pajek, which is software for analysing and visualizing networks. Finally, evaluations have been made according to the relations between "Development – Product – End User" networks.

4. ANALYSIS AND RESULTS

4.1. Pre-processing

In this study, two "bipartite networks" have been used. The first one is the "Development Processes – Products" network and the other is the "Product Attributes – Products" network.

There are two main partitions used for each bipartite network. The first one is the "Development Partition" for the "Development Processes – Products" bipartite network, which classifies the vertices in the network. The other partition is the "Product Attributes" partition for classifying the vertices in the "Product Attributes – Products" bipartite network.

There are seven main vectors used in the study; six of the vectors are coming from the award categories and includes the total points for each category. The seventh vector is the total points of the products according to the whole categories.

Data and the answers were grouped before analysing according to the groups mentioned in the section 0 of this paper.

Two types of questions were asked about the development process and product attributes in the questionnaire. Either we input a numerical value or we choose an option from the provided values in the question. In this study, every option in the answers of the questions is a vertex but we defined ranges and grouped the answers for the questions that require a numerical value. After grouping, every range became a vertex.

In development phase, subgroups were created according to

- Life cycles
- Estimated and actual software size
- Testing strategies
- Maintenance strategies
- Team size and experience

In product attributes, subgroups were created according to

- Targeted platform
- Category of the product
- Game story
- Game music
- Age, gender, education level of target users

It should be noted that the developers of the fifth product, P5 did not fill in the questionnaire. Therefore, P5 appears as an isolated vertex in the networks. The reason to include this product in the network is due to the fact that evaluator points exist for this product.

4.2. Analysis 1

The aim of this analysis is to find and show the common answers of the companies for the development process questionnaires.

Figure **2** is the bipartite network of the development process questionnaire answers and the game products. Pink vertices represent the products, green vertices represent the answers selected up to four times, red vertices represent the answers selected between four to seven times, and blue vertices represent the answers selected more than seven times in the questionnaire. In other words, green vertices are practices that are applied only by a few companies and blue vertices are practices that are applied by the majority of the companies.

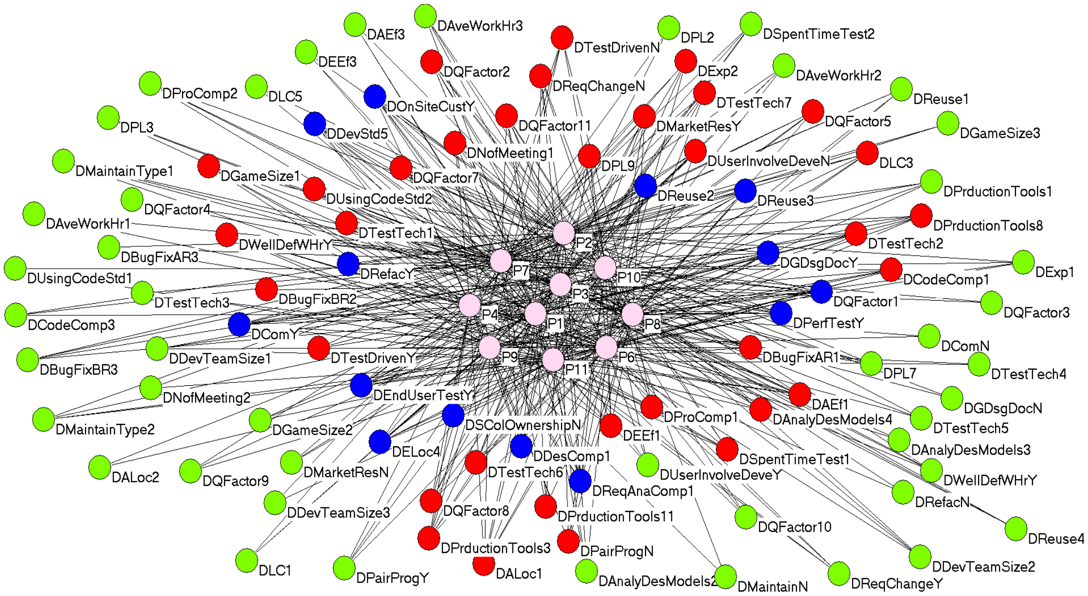

Figure 2 Bipartite network of the "Development Process and Products" according to developers' answers in the questionnaire.

Drawing the network with a different perspective allows seeing the results (information) more easily.

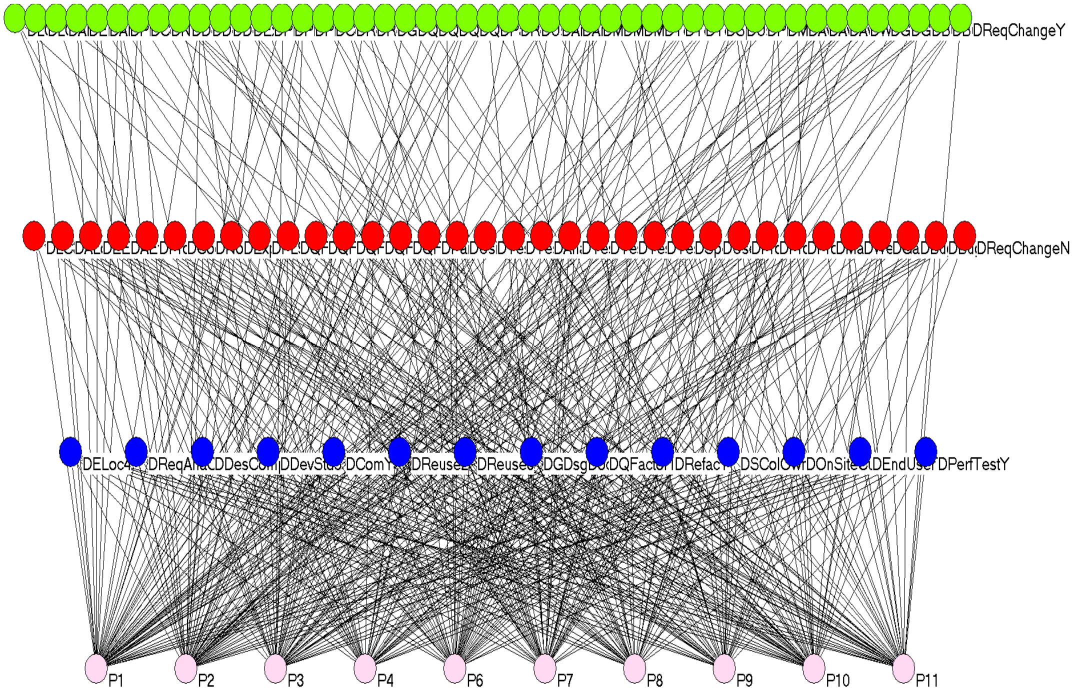

Figure 3 Bipartite network of the "Development Process and Products" in layered view.

Table 1 shows the most common development activities that were applied by at least seven companies out of ten during the development.

Table 1. The most common development activities that applied by companies.

Common Activities
Refactoring
On Site Customer
End User Test
Performance Test
Game Design Document
Code Commenting Mechanism
No collective ownership support
No LOC estimation
Complete the design between 0-50 weeks
Efficiency Quality Factor
Did not apply any Development standards or reference models
Code Reuse
Library Reuse

As expected most of these activities, such as refactoring, end user testing, performance testing, code commenting, library and code reuse are common practices in the software development. Some others may be domain specific.

After transforming this bipartite network to a 1-mode network, the product network is founded, which is shown in Figure **4**. In this figure, the relationship between the games according to shared most marked answers in the questionnaire are shown. Vertex sizes represent the total scores each product received from the evaluators in all award categories. It can be seen that the big vertices are separated from the other vertices. So it can be said that in the development of these high scored products popular practices were not followed. P1, P2, P6 and P11 form a clique, which means in the development of these products similar strategies were followed. However, it can be said that these strategies did not lead to high scored products. Therefore, these companies may benefit from a change in their development strategies.

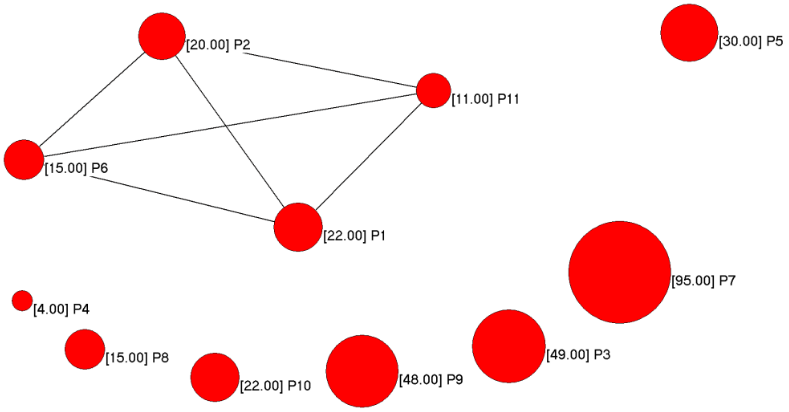

Figure 4. Products network according to common answers. Vertex sizes represent the total points from the user evaluation.

4.3. Analysis 2

The aim of the second analysis is to find and show the common answers of the companies for the product attribute questionnaires.

Figure **5** shows the bipartite network of the “Product attributes and Products” according to the answers in the product questionnaire. Green vertices represent the answers selected up to four times, red vertices represent the answers selected between four to seven times, and blue vertices represent the answers selected more than seven times in the questionnaire.

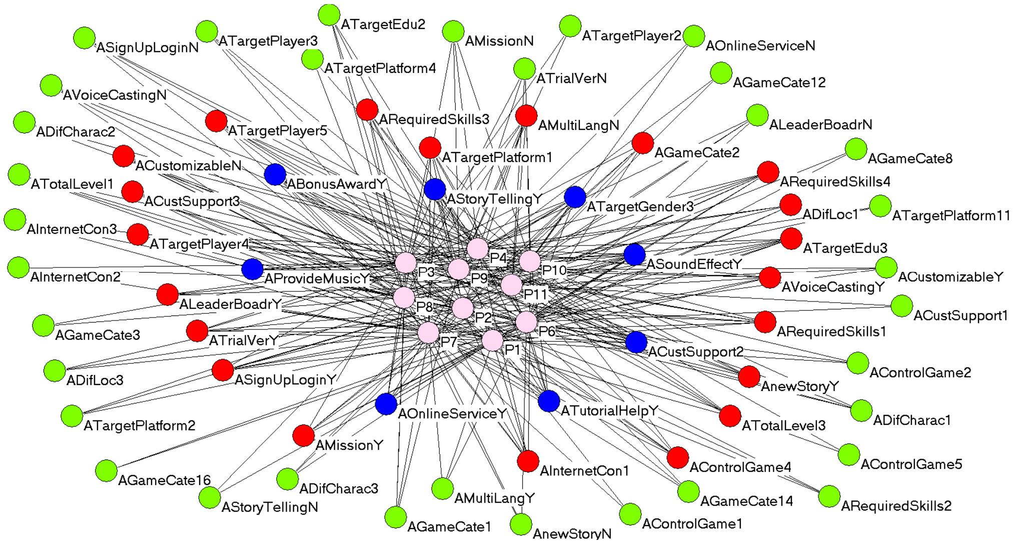

Figure 5. Bipartite network of the “Product attributes and Products” according to answers in the questionnaire.

Drawing the network with a different perspective allows seeing the results (information) more easily.

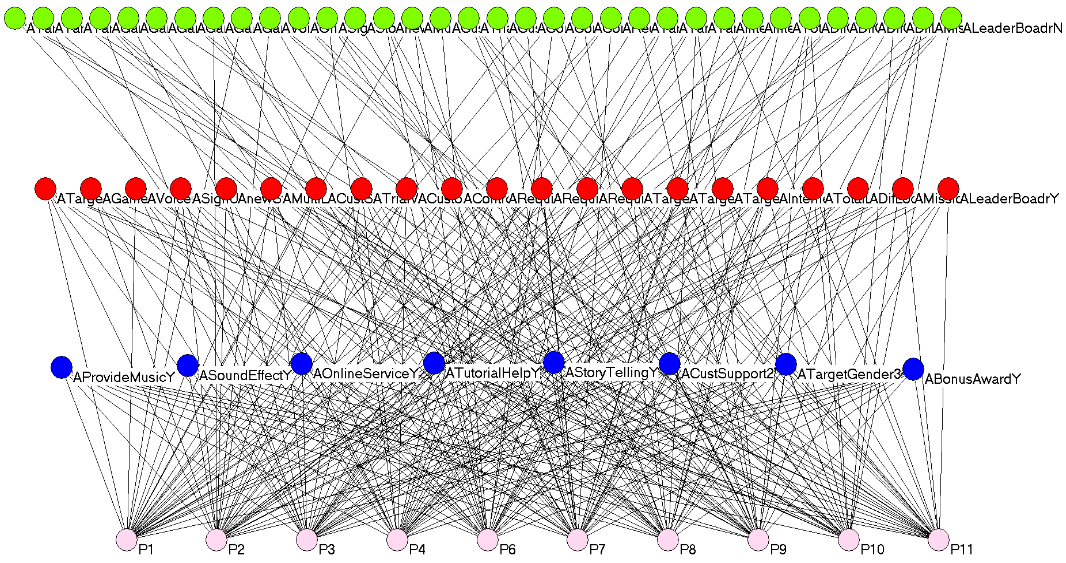

Figure 6. Bipartite network of the “Product Attributes and Products” in layered view.

Table 2 shows the most common product attributes that at least seven products have.

Table 2. The most common product attributes.

Common Product Attributes
Game gives bonus awards
Music
Sound effects
Online service
Story telling
Providing tutorial help
Targets both females and male users
Provides e – mail support

Removing the green and the red vertices from the network, the bipartite network of "Most common product attributes and the product" is formed. After transforming this bipartite network to a 1-mode network a different product network is founded.

In Figure **7** the relations between the products according to common answers in the product questionnaire are shown. Vertex sizes indicate their total scores in all award categories.

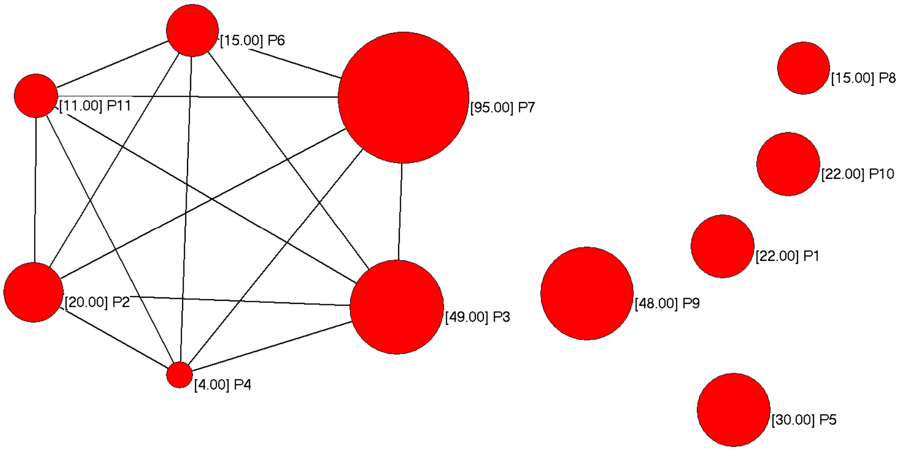

Figure 7. Products network according to common answers in the product questionnaire. Vertex sizes indicate total evaluator scores.

It can be seen that both successful and unsuccessful products share some common product attributes. Considering this observation together with that of the Analysis 1, it can be said that development strategies can be used to divide the products into 2 groups, successful and unsuccessful, whereas product attributes are not discriminating for the computer games. Analysis 3 investigates which development practices contribute to higher scores.

4.4. Analysis 3

The third analysis was done to find the popular answers for development process questionnaire for the highest three and lowest four products when the products were ordered in terms of their total evaluation scores. Therefore, the practices which contribute to the success or failure are determined.

Figure **8** shows the questionnaire answers given for products P7, P9 and P3, which are the most successful products according to the total scores given by the evaluators. Green vertices represent the answers selected up to four times, red vertices represent the answers selected between four to seven times, and blue vertices represent the answers selected more than seven times in the questionnaire. In this network, there are eight blue vertices, one green, and one red vertex with degree 3 (i.e., common answers).

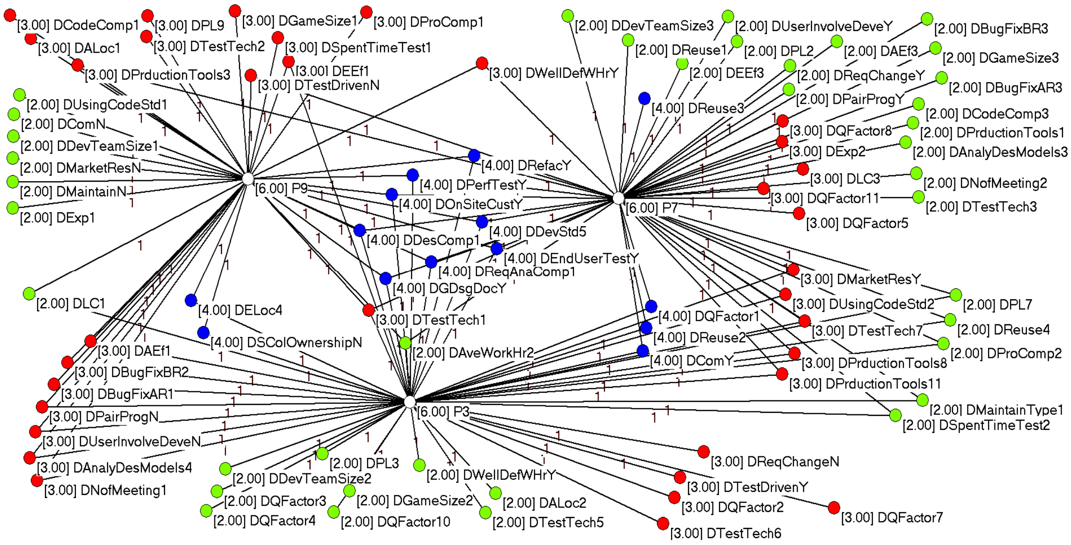

Figure 8. Three highest scored products and the answers given to the development phase questionnaire for these products.

Figure **9** shows the answers for products P4, P11, P6 and P8, which got the lowest points from the evaluation. In this network, there are ten blue vertices and seven red vertices with degree four.

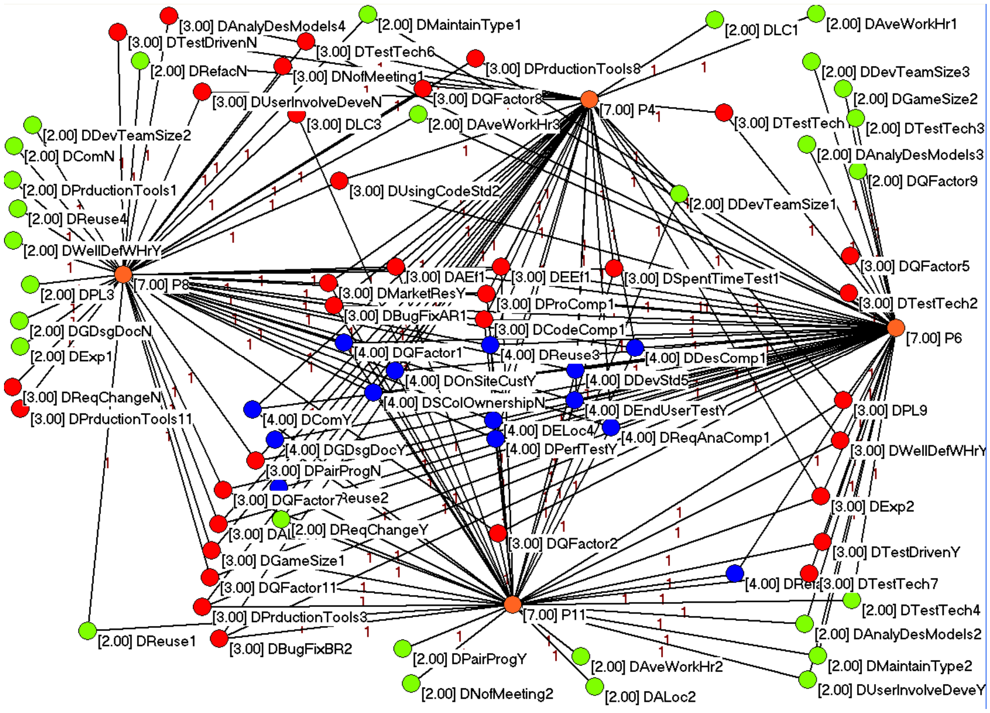

Figure 9. Four lowest scored products and the answers which were given to the development phase questionnaire for these products.

Table 3 shows the most popular answers of the successful and unsuccessful products found by this analysis. According to this table, for both successful and unsuccessful products performance testing and end user testing have been applied. These activities are activities that should be done in development process in software industry. Because of this reason, applying these activities does not make any difference on the success of the products.

Companies who developed the successful products (based on the end user evaluations) have carried out reviews in their development life cycles. As they use reviews to refine and engineer their products, this provides them a notable advantage in end user evaluations. According to this result, there is a positive effect of doing review on the success of the product.

Companies developing high scored products generate game design document to describe their products. Possibly, this is an additional factor resulting in better products. During the documentation of designs, more engineered and refined designs may be obtained, which can then result in better implementations.

Other difference between companies which developed successful and unsuccessful products is refactoring. It may be lead to more optimized products, which in turn may lead to more support from the end users.

Table 3. Common and uncommon activities of the successful products and unsuccessful products on development process

	Successful Products	**Unsuccessful Products**
Common Activities	Performance Test	
	End User Test	
	On Site Customer	
	Complete the design between 0-50 weeks	
	Complete the Requirement analysis between 0-50 weeks	
	Did not apply any Development standards or reference models	
Uncommon Activities	Refactoring	Market Research
	Review	No collective ownership support
	Game Design Document	Actual effort between 0-50 months
	Average working hours in week is between 10-40 hrs.	Estimated effort between 0-50 month
		Fixes bug after release between 0-25
		Efficiency Quality Factor
		No LOC estimation
		Complete the project between 0-12 months
		Complete the coding between 0-26 Weeks
		Spent time for testing between 0-5 weeks
		Library Reuse

4.5. Analysis 4

The fourth analysis was done to find the popular answers for the product attribute questionnaire for the highest three and lowest four products when the products were ordered in terms of their total evaluation scores. Therefore, the product attributes which contribute to success or failure is determined.

Figure **10** shows the questionnaire answers given for products P7, P9 and P3, which are the most successful products according to the total scores given by the evaluators. Green vertices represent the answers selected up to four times, red vertices represent the answers selected between four to seven times, and blue vertices represent the answers selected more than seven times in the questionnaire. In this network there are seven blue vertices, one green, and two red vertices with degree 3.

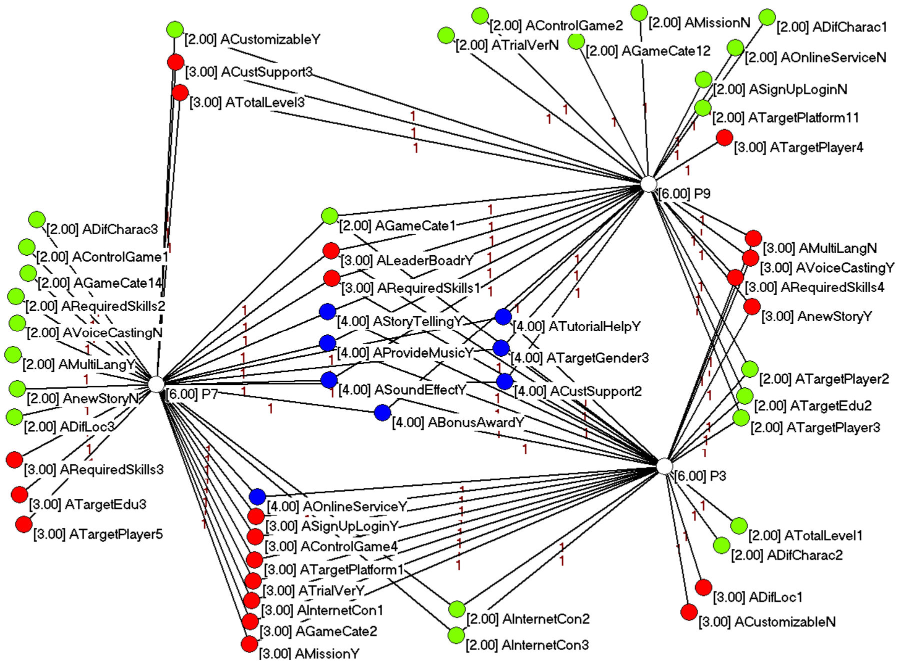

Figure 10. Three highest scored products and the answers which were given to the product attribute questions in the questionnaire for these products.

Figure **11** shows the answers for products P4, P11, P6 and P8, which got the lowest points from the evaluation. In this network, there are six blue vertices with degree 4.

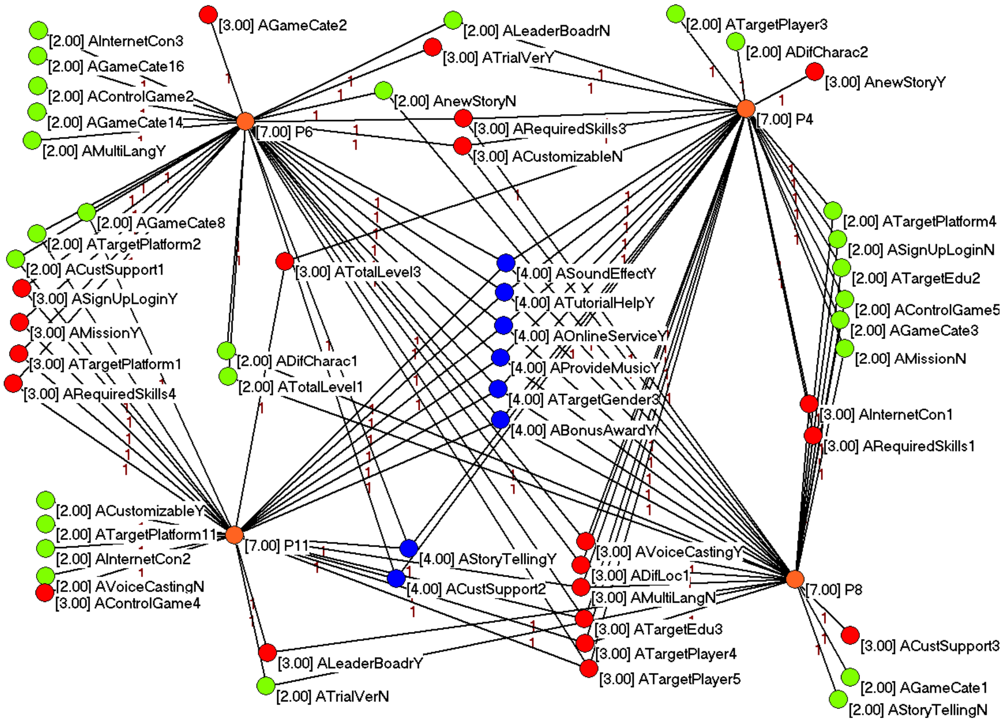

Figure 11. Four lowest scored products and the answers, which were given to the product, attribute questions in the questionnaire for these products.

Table 4 shows the most popular answers of the successful and unsuccessful products found by this analysis. According to this table, both successful and unsuccessful products have got music and sound effects. Therefore it can be said that having these attributes, had no effect on the success of the product. In the game industry musical attributes are seen as necessary attributes that every product should have. Similarly, both successful and unsuccessful products provide bonus award, tutorial and target both male and female gamers. So, these attributes do not have any effect on the success of the products directly.

On the other hand, successful products have story telling in the game. According the analysis above, it can be said that having this attribute has a positive effect on the users and can make the difference on the success of the products.

Table 4. Common and uncommon product attributes of successful and unsuccessful products

	Successful Products	**Unsuccessful Products**
Common Product Attributes	Music	
	Sound effect	
	Bonus award	
	Targeted genders are Females and Males	
	Provide Tutorial	
Uncommon Product Attributes	Action Game Category	Online service
	Leader Board	
	Agility is required skill for playing	
	Story telling	
	E – mail support	

5. Conclusions and Future Work

The aim of this study was to investigate the effect of the development process and the product attributes on the success of the software projects from the point-of-view of the end users. For this aim, a case study has been done. For the case study, the games sector in Turkey has been chosen. Product attributes and practices applied during the development phase have been collected from the developers via questionnaires. After pre-processing of the collected data, the gathered data has been analysed on “Pajek”, which is popular software for analysing and visualizing networks.

This study was a result of a test case made with ten companies active in the game industry. Our purpose was to get common connections and mapping on the groupings in our data instead of providing a general conclusion.

As a result of this study, it was found that the companies that develop successful products according to the user evaluation use different development practices. This means, similarities and differences about the characteristics of organizations can be used to predict the success of their products. Bipartite network analysis can be used for this purpose to determine and visualise how similar and different the products and the developers are. The questionnaires applied in this study made a distinction between domain specific and domain independent practices and attributes. A similar data collection, network formation and analysis methodology may be applied for software products and developers in domains different than games by changing domain-specific questions.

As a future work, more data will be collected from developers in both game and other domains to increase the network size and the reliability of analysis. Then, predicting the end user success of products will be tested based on the practices applied during the development phase and the product attributes.

REFERENCES

[1] Centers for Medicare & Medicaid Services (CMS) Office of Information Service (2008). "Selecting a development approach". United States Department of Health and Human Services (HHS). 2008. Available: http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf

[2] F. P. Brooks, "No Silver Bullet - Essence and Accident in Software Engineering", *IEEE Computer*, vol. 20, no. 4, pp 10-19, 1986.

[3] The Standish Group Report: Chaos (1995). Available: http://www.projectsmart.co.uk/docs/chaos-report.pdf

[4] L. Lin, H. M. Lee, "A New Algorithm of Software Quality Evaluation For User Satisfaction", *International Journal of Innovative Computing, Information and Control*, vol. 4, no. 10, pp.2639-2647, 2008.

[5] A. Holzinger, G. Searle, M. Wernbacher, "The effect of previous exposure to technology on acceptance and its importance in usability and accessibility engineering", *Universal Access In The Information Society*, vol. 10, no. 3, pp. 245-260, 2011.

[6] D. Denieffe, B. Carrig, D. Marshall, D. Picovici, "A Game Assessment Metric for the Online Gamer", *Advances in Electrical and Computer Engineering*, vol. 7, no. 2, pp. 3-6, 2007

[7] Ubicom White Paper, "OPScore or Online Playability Score: A Metric for Playability of Online Games with Network Impairments", 2005. Available: http://www.ubicom.com/news/whitepapers/whitepapers.html

[8] A. Groot, S. Kugler, P. Adams, G. Gousios, "Call for Quality: Open Source Software Quality Observation", Proceedings of The Second International Conference on Open Source Systems, June 8-10, Como, Italy, 2006

[9] Method for Qualification and Selection of Open Source software (QSOS) (2005), Atos Origin Available: http://www.qsos.org/.

[10] Business Readiness Rating (2005). Available: http://www.openbrr.org/.

[11] A. Tawileh, O. Rana, S. McIntosh, "A Social Networking Approach to F/OSS Quality Assessment", ICCNSM, volume 5322 of Lecture Notes in Computer Science, page 157-170. Springer, 2008.

[12] S. Marczak, D. Damian, "How Interaction between Roles Shapes the Communication Structure in Requirements-Driven Collaboration", 19th IEEE International Requirements Engineering Conference, pp. 47-56, Trento, 2011.

[13] B. MacKellar, "A Case Study Of Group Communication Patterns In a Large Project Course", IEEE 25th Conference on Software Engineering Education and Training (CSEE&T), pp. 134-138, 2012.

[14] W. De Nooy, A. Mrvar, and V. Batagelj, "Exploratory Network Analysis with Pajek". Cambridge University Press, 2005.

[15] M. Shuttleworth, "Survey research design", 2008 Available: http://www.experiment-resources.com/survey-research-design.html