Game Design for Project Management Game – A Tool to Educate on Software Project Management

Nitish M. Devadiga
Carnegie Mellon University
Pittsburgh, PA, USA

ABSTRACT

Software project management is a fundamental skill for software engineering practitioners. Due to the practical nature of the subject, it is essential to have hands-on experience with software project management. Risk management board game [1] was defined as an effective way for students to learn project management in a classroom setting. This paper outlines a detailed design reference for an online project management game to serve as an implementation guide and efficiently teach and learn software project management.

General Terms

Project management, game design, risk management.

Keywords

Software engineering, software project management

1. INTRODUCTION

"Project management can be defined as a way of developing structure in a complex project, where the independent variables of time, cost, resources and human behavior come together." – Rory Burke [2]

Product management is a function that assists in building the right product at the right time by carrying out activities in all phases of the software development lifecycle. The project manager manages assignments to make sure the product is built as per specification and within budget. A project manager requires precise decision-making skills and should be able to forecast the potential risks that can occur in a project.

The project management game enables the player (project manager) to work on diverse projects with different difficulty levels and scenarios and sharpen their project management skills. It makes the player think what software process to use, forecast activities, identify risks, manage resources, and have a visual blueprint of the desired result.

This paper serves as a potential reference for the online implementation of the 'Risk Management' board game. It explains the reasons for transforming the game from the board game to an interactive online game and how it helps in teaching and understanding project management.

2. PURPOSE

From an education perspective, the game has the following learning goals:

- Teach risk management concepts, along with the importance of managing risk on a software project and its associated costs.
- Taking risk-based decisions and understanding its impact.
- Understand the complexity involved in software development projects, the issues of resource tradeoffs and their effect on taking project decisions.

From a player's perspective, there are three primary goals to the game:

- Manage the development of a software product.
- Compete with other players to get the product delivered.
- Manage project risks while acquiring project capabilities.

3. PROJECT MANAGEMENT GAME

3.1 Board Game

The risk management board game is designed by CMU software engineering faculty [1]. The purpose of this board game is to make the students who take the risk management course get the concrete experience of risk management through the board game simulation.

3.1.1 General Description

The board game is designed like the popular board game Monopoly [3]. The process of this game is to simulate the software development lifecycle. The game represents the four main stages of the software development. Each stage is divided into specific software activities.

At the beginning of the game, each player is assigned a set amount of resources (money, employee), and they must complete the software activities on the game board to generate the software product. The goal of the game is to deliver the software products with the maximum financial gain.

To add risk into the game, after making, a decision (i.e., mitigate risk or move ahead to next software activity or talk to management) in the game, the player must roll dice. If the dice score is equal or lower, that means the player experiences failure or risk in software development, the player must pick a "surprise" or "oops" card, these cards might introduce a kind of risk or threat to the project. As the game progress, the players can collect capability points so that they can organize further advanced software activities.

Once the player finishes one step of the software implementation, the player can decide whether to end development or to continue. The monetary reward will be calculated based on the rank of finish order, finish time, and the number of risks mitigated.

3.1.2 Drawbacks

The player can move from one stage to another i.e., jump from one phase to another, this defeats the purpose of rolling a die as it's not related to the movement between the tasks.

The board game is designed for single a player, and there are no effective ways to get multiple players cooperate or compete in this game. The board is not very engaging when compared to browser games. There is no way to upgrade or increase the variability of the game.

The software processes defined in the board is fixed, the player cannot select processes like the waterfall development or agile development methods. Thus, the player cannot conduct a case study or project as every time the result will be the same.

The rules of surprise, mitigation and risk are written on associating cards, if the player wants to look for a certain rule, it is tedious. A better solution is to keep a copy of manual or booklet, which is sorted in alphabetical order, to facilitate the player to search. This issue is resolved in an online game as the user does not have to manually go through the cards [4].

3.2 Referred Games

The approach at first was to look at online board games such as Monopoly. However, in typical board games, the board serves a purpose of providing positional information and the number of moves of the piece on the board is correlated with the number that is rolled on the dice. In certain games, such as 'Settlers of Catan' [5] the board is shaped to show natural resources, and players occupy sections of the board with their pieces. Games such as 'Dominion', although categorized as a 'Board Game', do not typically have boards and contain a vast array of cards which are grouped into appropriate categories such as Currency, Action cards, and Victory Point cards.

This led to the exploration of options outside of having a board for the risk management game since the Risk management game was more about resource allocation with a complicated probability calculation based on dice rolls. We decided to look at games that focused on resource management.

Although unconventional when designing board games, concepts from Real Time Strategy (RTS) games such as Starcraft [6] are insightful. Players in Starcraft and similar RTS games make tradeoffs in advancing their technology tree against building more units. The concept of acquiring capabilities in the Risk Management game was parallel to the idea of upgrades to a technology tree in RTS games. Like the technology tree, the capability tree has prerequisites and dependencies.

Another key influence was Football Manager, a simulation game that puts a player in charge of a soccer team. The player does not play the actual soccer game, instead plays the role of the manager of a team and makes decisions regarding players such as buying and trading, while the football game between two teams is simulated. The user interface of football manager, with an email and communication tab, was an apparent influence in the UI design for the Risk Management Online game.

3.3 Browser based Game

The online version is a single player game that features the player as the manager heading a single project. The player is given resources in the form of people and money and must utilize them to complete the project.

The primary entities of the online version are:

3.3.1 Activities

It is the core entity of the game. Activities are grouped into phases and have dependencies among them and must be completed in a general order. To complete the game, the player must perform the deploy activity in the final phase. Every activity has a base success probability associated with it which is modified by resource ratings, capabilities and risks.

3.3.2 Capabilities

Capabilities are entities that a player can purchase from an in-game market, it alters the chance of success for performing activities.

3.3.3 Resources

People who can be hired and fired, each one having their own salary and stats (skill, work ethic, teamwork) which affect the chance of an activity succeeding. Salary is deducted with each turn.

3.3.4 Risks

Unlike the board game, where risk cards are distributed to the players, the online version has implicit risks which are tied to activities and modify their chance of success.

3.3.5 Single player

During analysis of board games, it was found that most multiplayer games had a competitive element where the moves of one player directly affected the other. The extreme example of this is Chess, where the opponent reacts to almost every single movement by the player, or Settlers of Catan, where one player can block another player’s actions.

3.3.6 Multiplayer

Competitive multiplayer needs a form of interaction with players, and the game in its current model does not support a strong form of multiplayer gameplay. One clear multiplayer option is shared resources and capabilities in a pool, where the cost increases with an increase in demand and resource usage are mutually exclusive, i.e., different players compete to acquire resources. Other elements of competitiveness can be added in the form of bids against other players to acquire projects.

3.3.7 Co-op play

Co-op play involves multiple players working together to accomplish a single goal. Usually, the challenge in co-op play arises from having to synchronize in timing with other players. The simplest form of co-op play for the Risk Management online game could be a form of a game where each person must work in parallel on different components.

Although the other two forms of gameplay are technically viable, at this the online version of the board game best supports single player core gameplay. With the competitive element arising from ratings and ladders based on best times/turns for the game and a social aspect where having more connections with other players leads to in-game benefits such as a better resource pool and more obscure capabilities in the capability market.

4. GAME DESIGN

One of the main reasons for the transformation to an online game was that students irrespective of their location could play the game without the necessary materials for the game. The game is used for teaching risk management in the Risk Management course. The two main issues observed was that students need to have the board with them to play the game also the game must be played against each other. To overcome these issues, it was decided to that an online game will be better as students can play the game as per their convenience and schedule [7]. A valuable aspect of an online game is that it can be updated and modified as and when required. In such a case the game can be adjusted as per the university's requirements, for e.g. They can add case studies as projects and rate the students as per their performance on those case studies. As an online game, the users can also simulate real work environment as a project manager.

4.1 Gameplay

Primary Objective: To complete the set of activities leading to the 'Deployment' Activity in the Release Phase of the Lifecycle. This is analogous to delivering a product to the customer

The main entities that the player interacts with are [8]:

4.1.1 Activities

Activities: These are grouped into phases and form the primary objective of the game. Activities are grouped into software development phases. The phases are

- Requirements
- Design
- Implementation
- Testing
- Release management

Depending on the project, each phase consists of many activities. These activities have pre-set dependencies that restrict the activities that can be performed at any time. The dependency is stored in the database in the form of a dependency graph. To complete the game, the player needs to perform the minimum set of activities successfully to reach the Release Phase in which the final activity is 'Deployment'. The number of turns the player takes and the amount of money the player spends in completed the tasks determines the rating of his victory. Every Activity has an associated cost (in money and resources) and turns required to finish it.

4.1.1.1 'Perform' an activity

Activities and their phases are available in the lifecycle tab of the game UI. To perform an activity, the player drags the activity to the action area, assigns resources to it, and clicks 'Complete Turn'. Each click on 'Complete Turn' counts as one turn. Any resources allocated to an activity are locked in for the turn duration of that activity. For example, if an activity specifies a resource cost of three, a money cost of 500$ and four turns, this means that three resources will be locked into that activity once the player clicks 'Complete Turn'. They will be available only when the activity is marked as COMPLETE or FAILED.

To perform an activity:

- The activity needs to be available. This means that the dependencies have first to be completed. For example, A player cannot directly start off by trying to perform some activity in the Development phase without completing the Requirements phase first.

- The player must have sufficient money as specified by the money cost of the activity.

- The player must allocate the resource cost mention on the activity.

Once the numbers of turns specified have been completed, the activity will either succeed or fail or internally fail.

A successful activity is marked as SUCCESS and this result in the linked activities that have this activity as a dependency enabled. Successful activities can be performed again (the reason for this will be described later)

A failed activity is marked as FAILED and does not result in the linked activities being enabled. A FAILED activity has a description that gives information as to why the activity failed.

An Internal failed activity is marked as SUCCESS and enables future activities, but can fail at a later stage. This happens when a future activity fails that is dependent on the internally failed activity. Thus, the player sees the task as succeeding but at a later stage, when a future task fails, it is linked to the failure of this task, and it is then marked as FAILED.

Below example illustrates the above scenario:

Consider the following activities in a dependency chain

ACT1 > ACT2 >......ACT10

ACT1 has succeeded and enabled ACT2

ACT2 has internally failed, so as far as the player is concerned, it shows 'SUCCESS' and the player moves along the chain and gets to ACT10. ACT10 fails because it knows that internally ACT2 has failed, and now both ACT2 and ACT10 are marked as FAILED.

Redoing a successful task can reduce the chance of internal failure. A task marked SUCCESS can be re-done as mentioned above, and although it still shows SUCCESS, it has the possibility of changing an internally failed task to an actual success

4.1.1.2 Probability of success calculation

Every activity has a base chance of success as shown in the data model. Every activity may also have a risk associated with it, which applies a reduction multiplier that reduces its chance of success. For example, an activity can have a base chance of success of 0.8, which means the probability of success is 0.8 if all the criteria are met. Several multipliers can be applied to this activity to modify its actual chance of success. (Note: These probabilities are not shown to the user).

4.1.1.3 Risk Multiplier

If a risk affects a certain activity, the game internally keeps track of it. A risk reduces the chance of success. For example, a risk can be 'reduces the chance of gathering requirements by 10%'. Therefore, after the risk multiplier is applied, the actual chance of success becomes 90%(0.8) = 0.72

4.1.1.4 Capability Multiplier

Capabilities can improve the chance of success of an activity. For example, the UI shows a capability that states 'Business Analysis: Increase the chance of gathering requirements by 33%'. The chance of success now becomes, 0.8 * 133% * 90% (taking into consideration that a risk is also present) = 0.95

4.1.1.5 Resource Multiplier

Resources will have three basic attributes which also apply a standardized multiplier to any task. Some activities require a higher level of skill to be completed and their chance of success will be penalized if the allocated resource does not meet the required skill rating.

4.1.1.6 Success multiplier

An activity that has already succeeded or internally failed will have reduced costs in terms of resources as well money, and have a multiplier that increases its chance of success. Thus, repeating an activity has a better chance of avoiding internal failure.

An activity can originally have a resource cost of 3, a money cost of 500$ and a base chance of success (not visible to the user) of 0.7. Once it has either succeeded or internally failed, the costs become (1,200$), and the chance of success has a multiplier of 1.1 which increases the chance of success by 10%. Doing it a second time would increase the chance of success by an additional 4% i.e. 14% in total from the base chance. This multiplier has diminishing returns and will cap out at say 16% beyond which it cannot be improved further without purchasing capabilities or changing the resource stats through training (also a part of capabilities)

4.1.1.7 Failure Multiplier

An activity that is marked as failed will have a penalty associated with it that depends on how far away from the activity the user currently is. If a task has internally failed in the requirements stage, and this comes to light in the testing phase, the user is now 3 phases away from the failed activity. Therefore, reflecting a real-life scenario, the resource and money cost have an x2 multiplier associated with them (or it can only be the money). The user gets some form of notification in-game, such as a mail from an employee stating that 'This activity might not have gone well', which might or might not be true, but the user can decide that it might have internally failed based on many such factors. The values of the risks and multipliers are implicit and are not shown to the user, but the game must convey these values to the user through more realistic forms such as emails from employees or an RSS Feed that contains market news. These will help the player better determine which capabilities to buy and which resources to train.

For example, in the game, if the player gets a mail from his employees at the beginning of the project stating they are not comfortable working on a technology, and the player sees that their skill level is low. It might be in the player’s best interests to spend a few turns and some money purchasing capabilities that train the employees.

4.2 Capabilities

The second most important entity in the game that the user interacts with directly is Capabilities. As mentioned in the Activities section, Capabilities modify the chance of success of an activity by applying an increasing multiplier to them. Like activities, Capabilities also have dependencies and the player cannot directly purchase ‘Competitive Intelligence’ without first purchasing ‘Business Analysis’. Capabilities also have a money cost and a (variable) resource cost associated with them. They cost money, resources and time. The primary objective of the game is to get to the ‘Deployment’ activity in the ‘Release’ phase. In a specific scenario, since most activities have some base chance of success, and risks are randomly allocated, it is possible that the user can just perform the activities, without buying any capabilities and reach the Release activity. But this is highly unlikely and measures should be taken in setting the values so that certain activities will always fail unless certain capabilities are purchased.

4.2.1 Additional pointers:

- Since capabilities cost time, money and lock in resources, it is not in the player’s best interests to just keep purchasing capabilities without making any progress in activities.

- Advanced capabilities cost more money, but are not always necessary. sometimes, just redoing an activity might be more beneficial than going along a dependency chain and purchasing a very expensive capability

- Some capabilities affect a wide range of activities and could be cheap, which make them obvious and good investments for the player early in the game

- Some capabilities will intentionally be expensive, and provide very little benefit to the player, which makes them risky investments

As the user purchases capabilities, some risks might get eliminated, such as when the multiplier becomes large enough to push the base chance of success to 1 (or even beyond 1, but this is not beneficial) which means the 'Activity' will succeed.

As of now, capabilities will impact activities or a group of activities or complete phases. Some capabilities can reduce the resource cost or the money cost of performing an activity or even the cost of performing a failed activity. For example, purchasing the iterative development capability could reduce the penalty of re-doing a failed activity in the earlier phases, which is as expected.

Capabilities are meant to modify the gameplay and the idea is to present a challenge to the user in selecting the right capabilities and making right decisions to complete the primary objective of the game.

4.3 Resources (Employees)

Employees have four main attributes

- Skill

- Work Ethic

- Team Work

- Base Salary

Each of these attributes 1-3 has a rating of 1-10. These ratings are used to calculate a multiplier to apply to every activity. It is reasonable to assume that employees having a better teamwork rating will be more effective in performing activities than that have a considerable resource cost. Technical tasks that have a single employee cost are not affected by the teamwork rating and are affected only by the skill rating.

Base salary is the amount that is deducted on every turn (regardless of whether the resource is locked in completing an activity or not). The base salary is meant to incentivize players to maximize resource utilization and use them in parallel when necessary. (Sometimes, when there are no parallel activities, it is more beneficial to put them to work on SUCCESS activities on the suspicion that they might have internally failed) Regardless, it is up to the player to work out how best to make use of the resources.

The player can access resources through the HRMS and Employees tab in the game. It is up to the player to decide the best buildup of the team to complete a project. Training for resources are also available through the capability market, and these can impact the rating of the resource (with diminishing returns).

Having too many resources will impact the amount deducted on each turn, and could end up finishing all the money much before the Deployment activity can be reached, which is an automatic game over.

4.4 Risks

The final entity that impacts the game is 'Risk'. Risks are implicit and have associations with activities, a group of activities or full phases. As mentioned in the activities section, risks apply negative multipliers to the chance of success of an activity, which reflects the risk probability. In addition to which, they also have a risk impact.

Risks can have an organization-wide impact and can increase the cost of failed activities. If the user gets a notification of a potential employee shortage and does not purchase employee retention capabilities, he could end up having a resource shortage and having to work with the bare minimum number of resources, which affects the player's utilization and parallelization, requiring more turns.

5. CONCLUSION

Product management is a function that assists in building the desired product within the defined time by carrying out activities in all phases of the software development lifecycle. A project manager organizes actions to make sure the product is built on time and within budget. A project manager requires precise decision-making skills and should be able to forecast the potential risks that can transpire in a project.

The project management game enables the player to take on the role of a project manager to work on projects with several difficulty levels and scenarios to sharpen their project management skills. It makes the player evaluate which software process to use, forecast activities, mitigate risks, manage resources, and have a visual blueprint of the desired result.

This paper documents in detail a design reference to build a project risk management game transforming the game from the board game which would aid in teaching project management and managing software development projects.

6. REFERENCES

[1] Taran, Gil. “Using Games in Software Engineering Education to Teach Risk Management.” 20th Conference on Software Engineering Education & Training (CSEET'07), 2007, doi:10.1109/cseet.2007.54.

[2] “Project Management – Sage Strategies.” Sage Strategies, sagestrategies.biz/project-management/.

[3] “Monopoly (Game).” Wikipedia, Wikimedia Foundation, 19 Mar. 2018, en.wikipedia.org/wiki/Monopoly_(game).

[4] Kanode, Christopher M., and Hisham M. Haddad. “Software Engineering Challenges in Game Development.” 2009 Sixth International Conference on Information Technology: New Generations, 2009, doi:10.1109/itng.2009.74.

[5] “Catan.” Wikipedia, Wikimedia Foundation, 19 Mar. 2018, en.wikipedia.org/wiki/Catan.

[6] “StarCraft.” Wikipedia, Wikimedia Foundation, 19 Mar. 2018, en.wikipedia.org/wiki/StarCraft.

[7] Yu, Kuang-Cbao, et al. “The Implementation and Evaluation of Educational Online Gaming System.” ITRE 2005. 3rd International Conference on Information Technology: Research and Education, 2005., doi:10.1109/itre.2005.1503137.

[8] Ariffin Baba, Saiful & Hussain, Hanafizan & Che Embi, Zarina. (2018). An Overview for Parameters of Game Engine.