**Comp 3520, Comp 4530**

**Software Engineering / Advanced Software Engineering**

**Assignment #1: Software Process Model**

**Kevin Zwick & Dirk Neumann**

**T00463994, T00xxxxxx**

**Due September 19th**

**Table of Contents**

**Executive Summary**

**Part I: Introduction**

**Part II: Our Business Opportunity**

Like every start-up company before our company has a very small amount of money to start with. Therefore, time and effort that could be invested are very limited until that developed software generates an income. This also means that the product after a short time and without many developers cannot be very big in functionality and so in price. Also, there is a quick feedback from potential customers needed to find out whether further investments are useful or not. That means the product should be fast accessible, easy to handle and easy to evaluate.

The best version to accomplish this goal is by using a mobile app. The development effort is small because of smart development environments like android studio with high level languages like Java so there are no further educations and investigations needed. The basic concepts are already known, and many parts of the app can be found in other contexts in the internet. Android studio also offers the service to show how big the percentage of users of different android operating systems which means we can start developing the app for the biggest one to reach as much users as possible. A mobile application also has the advantage that we can use the services offered by the “Android Play Store” or the “Apple App Store”. We do not have to care about servers to offer the download of our application and it is even easier to get evaluations. Also, we do not need to implement a secure possibility to process financial transactions or store user data and make them protected from illegal access. The platform where we offer our application will take a part of the money we earn, and we can use the rest for further development.

The question is what kind of mobile application we can develop in a short time, but with big potential. We chose a logical game. There are several advantages to this way, explained in the following. At first a logical game generates the fun in the mind of the user. We do not need fancy 3D animations, which need a big amount of the device resources. It can be played on older smart phones and need really less storage capacity. The user does not have to decide whether he wants our game on his phone or a large application (i.e. the Facebook application) because ours will need only 5% to 10% storage of a really big application. This increases again the number of potential users. In addition, the game does not have to offer new content every three weeks to the user. The user can be kept busy with the same gaming principle but at a higher difficulty. This can be a more intelligent artificial intelligence, a handicap for the user, time limits, less possibilities to accomplish the games goal or just more goals. In the following versions new elements can be introduced and communication between multiple users can be unlocked. In this way we can let the user do work for us, for example by creating his own maps or levels and submit them to the application platform. Other users can download them, play them and rate them. In this way we do not have to do anything and it will not get boring for the user.

How to start? The first version of our app will contain just a small amount of levels without an artificial intelligence and user created levels. Only a small campaign with maybe 80 – 100 levels in increasing difficulty. The download is for free but therefore comes with advertisements in the app. The user gets the possibility of removing the advertisement for a small amount of money and we will sell it as a possibility to support the development of more levels. This money and the one from showing the advertisement can be used to create more content. In the end we can offer an advanced version of the application for a few dollars but keep the free version available. The given evaluations can be used as inspiration for future developments or maybe a concept for a new game. The same process for the new game and from there on we can display advertisement in every app for our other products. Our income is therefore based on many users which have paid just a very small amount of money once or twice. If we have a stabile income, we can use these resources to develop a bigger application with many micro transactions in it. For example, $0.99 for five extra tries on a certain puzzle or a special in game currency which abstracts from real money so that the customer forgets about how much money they are actually paying.

What will the game be about? The game idea is to represent a factory in which one or multiple items should be produced. Therefore, different resources must be combined in different machines, delivered through the factory hall, recycled or produced. From level to level the recipes of the machines get harder, maybe multiple outcomes, time limits, less space, no crossings of delivery routes, water or power supply and more. The user can later challenge the community with its own level design and generate more context for all users without us having to do the work.

**Part III: Software Process Model**

Being a new start-up company with ourselves as it’s only two programmers, our time, skills and knowledge are our main assets. Our process model does not yet need to be extremely complex, but to make our efforts successful it does need to be well defined. Our first projects are chosen to be ones that we can develop quickly and get to market quickly, as our business’s first need is to get an income stream. All this has been described in greater detail already, and these are also the basis for our choice of software process model.

We have chosen to use the incremental development model and use integration where possible for some of the modules. Our incremental model will be done using an agile method rather than being more plan driven. With two programmers we can have short meetings or scrums to decide our current priorities and to make sure that the modules we are working on will work properly with each other. This way we can get our current increment up and working in as short a time as we can. Also, to further this goal we will be looking to re-use or integrate existing code for parts of our project. There is existing code already for 2D grid maps and algorithms for logical resource gathering and other parts of our project. This type of code has already been created for Android Studio many times. Our challenge will be to find the code that best fits our purposes and then to integrate that code with the rest of our project. Using these methods combined we should to able to develop the first version of our application in a reasonable amount of time and get it to market quickly. We will also be able to create updates and expansions of this application to keep people interested by releasing new and more challenging content for a small fee. To summarize, our software process model is incremental using the agile method. We will also be using integration for some modules.

**Part IV a: Incremental by Agile**

There are differences in software process models for different size entities. There are Very Small Entities (VSEs) which are defined as “an enterprise, organization, department or project having up to 25 people” (Laporte et al, 2008). The is also a micro size enterprise which is up to 9 people and there are Small, Medium, and Large Enterprises which are defined as <50, <250, and 250 + respectively. “Lifecycle profiles for Very Small Entities” are aimed at meeting the specific needs of VSEs (O’Connor and Laporte, 2011a). With our start-up of two people with essentially no start capital and our main assets of our time, skills, and knowledge, our priority is getting an income stream. Our choice to get an income stream started quickly is to develop a mobile app game to get to market quickly. Our game will add levels and difficulty increases in frequent updates to keep users interested. Some of the expansions of our game will have a small fee. This development plan leads naturally to our software process model being incremental.

With our choice of an incremental software process model there is now another choice of either a plan driven model or an agile model. As we are wanting to get a version of our game quickly to market and our available manpower is very limited, a small amount of concise documental makes the most sense. This fits with the Agile process model. “The Agile models were defined to value individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan” (Abrahamson, et al. 2002). This fits with our priority of getting a working game out to market quickly. Also, with our first release we should be getting customer feedback and using agile methods will allow us to respond to these in our next increment quickly. These reasons support our choice to use a Incremental Process with the Agile Model.

**Part IV b: Integration and Configuration**

The project will also be based on integration and configuration of already existing pieces of code. The reason behind this is to reduce the amount of work as much as possible to reach the point of the first income generated by the application as fast as possible. Especially for the development of android applications there are many tools which need just a few mouse clicks to generate the whole layout and can control things like volume of background music, or auto-rotation if we want our application to support this. Also access to multiple sensors like tilt sensor and GPS sensor is easy. The visualization part of the application, so the game itself, can also profit from multiple already finished parts like: a map to place different things on it, graphics for different items or objects and the rendering of their animation or them self when they are moved over the map. This leaves us with only two tasks: putting the pieces together and adding the logic behind the game. The first task needs some additional configuration which should not be to difficult since mobile applications are very similar, and the tools do most of the work with auto generated code. The second task to add the logic is the harder but also, he only big problem which we need to solve.

**Part V: Conclusion**

**Bibliography**