1. Contents

[1: Introduction 2](#_Toc528166355)

[2: Dependibility 2](#_Toc528166356)

[3: Security Engineering 3](#_Toc528166357)

[4: Resilience Engineering: 3](#_Toc528166358)

[5: Conclusion 4](#_Toc528166359)

[References 5](#_Toc528166360)

# 1: Introduction

The developing of software is not just developing and adding functionality. A big part of the work in software development is not visible in the end and also it shouldn´t. Other parts of the software like the functionality itself and user interfaces should be visible and to keep them visible it needs support and defense. Dependability and Security are, especially in very important systems, big issues. Big and important systems have to run without an error, without any security issues and this for a long time without a stop in between. Due nothing is a hundred percent safe in software engineering an error may occur but then, when the worst case becomes true, the system has to be able to be recovered quickly without losing important data or producing inconsistent states.

In this paper we will analyse what properties in terms of security and resilience engineering our product should have, if it has them and how they are implemented.

# 2: Dependability

During the implementation of our first version of the game we just developed functionality and interface. We did not spend time and effort in thinking about making things safe or stabile to work. Basically we also do not have to. Since we are developing a mobile app for android we can use all the offered functionalities provided by google. That is on the on hand much cheaper than developing it on our self and also much more reliable since our knowledge about programming secure software is not very big. The extra time to learn it properly will cost to much money and we will need longer to develop updates afterwards.

# 3: Security Engineering

Our System also have nothing like security standards against hacker attacks. If somebody want to crash our app it wont be to difficult for him or her. But why do we not care about it. First of all, our mobile app is based on the device of the user who uses it. If this user decides to crash the game on purpose only he himself can not play it anymore. Other users are not infected by that. If a user really want try destroy the app for all users he would have to be able to write a fake update, pretend to be us and upload it to the play store and wait for others to download it. Since there is no transaction system, log in system or anything else dealing with sensible data of users, this hacker wont get anything from that and would waste his time. Also google security standards should be very high and nearly impossible to hack if you are alone and there wont be a whole organisation trying to kill our little mobile game.

# 4: Resilience Engineering:

But let us imagine there would be somebody who would do that anyway and every user would download the update which basically breaks the app and the user can not play anymore. This would cost us probably some users who would delete the app immediately but other users would report this over the play store rating function or the support function in our app if this is already done. After we recognized the issue we could contact the play store with the last working version and ask them to offer this one instead of the not working one. Problem fixed. The user progress (i.e. in the champagne) can either be saved on the users device or in a cloud provided by the play store so the data would be safe unless the hacker manipulated it so that the app deletes the progress. In this case it wont matter where the data is and the user would have to start from the beginning after our app is recovered. There are not really any other threats like a black out, denial of service attacks, bad user input. Only the elements we provide can be placed and there are no entries like in text boxes.

# 5: Conclusion

In conclusion we showed that Dependability and Security is not really an issue for us. Maybe it could become on in case we will integrate something like micro transactions and a payment system in our app but until then our app is free from sensible data. All security functionalities are provided by the platform we want to offer our app on and also payment transactions and saving user data can be outsourced to it.

# References

Sommerville, I. (2016). *Software engineering*. Harlow, England; New York: Addison-Wesley, 2016.

Davidsen, M.G., and Krogstie J. 2010 “A Longitudinal Study of Development and Maintenance.” *Information and software Technology* 52 (7): 707 - 719