Exercise 3 - - Software Engineering Economics

Exercise 3.1 - Good and Bad practices

There are several factors which distinguish good practice software projects with bad practice software projects. The most important factors for good practices are that it is an fixed and experienced team, it is released based and the project has a steady heartbeat. The latter means that the project has a fixed length iteration as short as you can make it. All the three factors mentioned are related to an agile way of working. So for a good practice software project, it is very important to work in an agile manner.
The bad practice software projects are projects which are driven by rules and regulations, which have dependencies with other systems, which are technology driven and which are once-only projects. It is therefore very important for a project to have some freedom within the project, the team and the system, because then it is possible to avoid making the project a bad practice project.

Exercise 3.2 - Visual Basic

Of the Visual Basic projects, the most of those ended up in a good practice project. But it is not so interesting because there were only 6 Visual Basic projects, which is already a small amount of projects within the 352 projects analyzed in the paper. Furthermore, are on average less complex than other projects, therefore end up as a good practice project more often.

Exercise 3.3 - Good/Bad factors

1. Planned testing This belongs to a good practice. It is important to not test at the last moment. It needs planning to make sure that the testing is not only done when the schedule gets tight. It is also important for a project to plan test cases before the coding starts. This all makes sure that members in a project do not waste a lot of time in fixing many tests at the same time. Planning the tests will save a lot of time.
2. Code review This belongs to a good practice. A lot of problems are eliminated earlier when a project team regularly reviews each others code. It is important to review to code on bugs, design, architecture and everything to make the code better. This may be more effective than testing.
3. Requirements check This belongs to a good practice. When a company wants a new program, it is very important as a team to understand exactly what the company wants. When you start the project without checking that it is understood what the company wants, there is a high risk that you don't do it right, which leads to unnecessay work. This costs a lot of time which can be avoided by checking the requirements.

Exercise 3.4 - Bad practice factors

The 3 bad practice factors which will be discussed are: Many team changes,... Many team changes is a factor where, as the name says, some of the team members leave within the project and/or there come new team members within the project. The latter is the biggest problem in a software project, because the member first needs to understand everything of the project. The member needs to, for example, read the code to get an understanding of it. This costs a lot of time to let the new member be a good part of the team. When a person leaves the project, it also costs time to take over the work of that person and to get an understanding what the member was working on. So when a team change occures, it costs time for one or more members to understand a new part. ...