Taxi Service
SVN Policy

Version 1.00

Revision History

Date	Version	Description	Author
2012-10-31	1.00	Initial version	Karlo Zanki

TABLE OF CONTENTS

1. Introduction

1.1 Purpose of this document

The goal of this document is to provide some conventions regarding SVN usage to all developers working on the project. To avoid conflicts in the code and parts of the code being deleted by other members of the team, all members of the team should stick to the advices given in this document.

1.2 Document organization

The document is organized as follows:

- Section 1, Introduction, general description of the contents and main purpose of this document.
- Section 2, SVN usage policies, useful advices for safe usage of the SVN.

1.3 Intended Audience

The intended audience is:
- Members of the taxi service team.

1.4 Scope

This document will be used during the work on the Taxi Service project but it can also serve as a reference to some future projects.

1.5 References

[1] Subversion conventions: http://solowebstartup.com/top-18-subversion-best-practices
[2] TortosieSVN: http://tortoisesvn.net/downloads
[3] Eclipse SVN plug-in: http://www.eclipse.org/subversive/

2. SVN usage policies

SVN is an open-source version-control system. For the Windows SVN client used in this project we recommend Tortoise SVN, but also it is recommended to use SVN tool inside Visual Studio. AnkhSVN is a good solution for this purpose, once installed it offers functionalities quite similar to those of the Team foundation server. For the Eclipse IDE a tool is also listed under the references section of this document.

2.1 Best practices

2.1.1 Update always before you start to work.

2.1.2 Update as often as you can, at least every hour while you work on the code.

*2.1.3 **ALWAYS** update before you commit*

2.1.4 Commit only buildables.

2.1.5 Commit small sets of changes.

Don't let your commits be too large, it will make merging complicated and increase the chances of getting into conflicts .

2.1.6 Commit related code changes together.

2.1.7 Commit after every bug fix.

2.1.8 Respect other developers and their code.

Don't make aggressive merging, if you have conflicts try to resolve them by discussing with the other developers first.

2.1.9 Take responsibility for your commits

If you commit code that causes other parts of system to fail, you should response quickly and try to fix it as soon as possible and you shouldn't wait for someone else to do it instead of you!

2.1.10 When you check in, spend some time on writing down a summary of changes, it will simplify finding correct versions if something goes wrong later.