# Glossary

# Major functionalities – Overview of ClearCase functional automation

This section provides an overview of the key automation functions in ClearCase and related Rational™ Suite integrations. First, we will outline each of the functions and then provide an explanation and rationale using configuration management methodology.

Then we will justify each of them in context of our project and provide you with a selection rationale to aid management decisions.

## Version Control

### Accessing, maintaining and managing project assets

We needed a tool to facilitate two basic features for software developers. We need to be able to share code/work amongst the team and we need to guard and rollback against unstable progressions.

We define project assets as any articles, source code, documentations, designs and also directory structure of the project file system.

The ability to manage abstract assets is what sets ClearCase apart from traditional version control softwares.

The outcome of managing project assets is achieving working efficiency.

ClearCase uses the “sandbox” approach on accessing project assets with a very obvious and advantageous difference. ClearCase implements a development environment called *View* which act as a configurable data source for the team or developer in question.

To put it simply, the major advantage of using ClearCase *View* feature is that it guarantees stability for the global data repository. Each *View* maintains and keeps its own changes; developers working with other *views* are not affected by changes. However, they may access other views provided they have the appropriate access permissions.

### Software builds process management

### Audit Trails

### Automated workspace management

### Integration with Rational™ Suite and popular IDEs

## Collaboration

Support for parallel development

Multi-platform support

Implementing and enforcing project policies